60469290

CONTROL DATA

CDC® CYBER 170 COMPUTER SYSTEMS MODELS 835, 845, AND 855

CDC® CYBER 180 COMPUTER SYSTEMS MODELS 835, 840, 845, 850, 855, 860, AND 990

CDC® CYBER 990E, 995E, AND 994 COMPUTER SYSTEMS

CYBER 170 STATE

HARDWARE REFERENCE MANUAL

Central Processor Instruction Index

AX 16-21
AX 16-22
BX 16-24,44
CC 16-48
CR 16-61
CU 16-49
CW 16-61
CX 16-68
DF 16-12
DM 16-47
DX 16-29
DX 16-32
DX 16-36
EQ 16-13
FX 16-28
FX 16-31
FX 16-34
FX 16-37
GE 16-14
ID 16-13
IM 16-47
IR 16-11
IX 16-8
IX 16-8
JP 16-40
LT 16-15
LX 16-19
LX 16-20
MX 16-68
NE 16-14
NG 16-11
NO 16-65
NX 16-63
NZ 16-10
OR 16-12
PL 16-10
PS 16-66
PX 16-6
RC 16-69
RE 16-17
RJ 16-39
RX 16-67
RX 16-30
RX 16-33
RX 16-35
RX 16-38
SA 16-51
SB 16-55
SX 16-58
UX 16-7
WX 16-67
WE 16-18
XJ 16-42
ZR 16-9
ZX 16-64

60469290

CONTROL DATA

CDC® CYBER 170 COMPUTER SYSTEMS MODELS 835, 845, AND 855

CDC® CYBER 180 COMPUTER SYSTEMS MODELS 835, 840, 845, 850, 855, 860, AND 990

CDC® CYBER 990E, 995E, AND 994 COMPUTER SYSTEMS

CYBER 170 STATE

HARDWARE REFERENCE MANUAL

Manual History

This manual is revision H, printed August 1988. It reflects Engineering Change Order 49867 which adds information in support of CYBER 994.

Revision	Change Order	Date	Reason for Change
01	-	05-15-81	Preliminary issue.
A	-	04-23-82	Manual revised to add support of model 855.
B	44612/PDO3024	12-30-83	Manual revised to include Comment Sheets 2924, 2745, and 3068. Model 845 information added. This edition obsoletes all previous editions.
C	-	04-28-84	Manual revised to add support of CYBER 180 Models 835, 845, 855, and 990.
D	46271	11-02-84	Manual revised to add support of CYBER Models 840 and 850.
E	46744	05-14-85	Manual revised to add Memory Upgrade Option.
F	48300	04-30-87	Manual revised to add support of CYBER 990E and 995E.
G	49229	11-30-87	Manual revised to add IPI information.
H	49867	08-30-87	Manual revised to support CYBER 994.

Contents

About This Manual 7

Audience 7
Organization 7
FCC Compliance 7
Conventions 8
Related Manuals 8
Additional Related Manuals 8
Ordering Manuals 8
Submitting Comments 10
Disclaimer 10

Model 835 System Description 1-1

Physical Characteristics 1-1
Functional Characteristics 1-2
Major System Component Descriptions..... 1-6

Model 835 Functional Descriptions..... 2-1

Central Processor 2-1
Central Memory 2-15
Input/Output Unit 2-23

Model 835 Operating Instructions..... 3-1

Controls and Indicators 3-1
Power-On and Power-Off Procedures..... 3-9
Operating Procedures 3-9
IOU Reconfiguration 3-11

Models 845 and 855 System Description..... 4-1

Physical Characteristics 4-1
Functional Characteristics 4-4
Major System Component Descriptions..... 4-8

Models 845 and 855 Functional Descriptions..... 5-1

Central Processor 5-1
Central Memory 5-15
Input/Output Unit 5-25

Models 845 and 855 Operating Instructions..... 6-1

Controls and Indicators 6-1
Power-On and Power-Off Procedures..... 6-14
Operating Procedures 6-14
IOU Reconfiguration 6-16

Models 840, 850, and 860 System Description..... 7-1

Physical Characteristics 7-1
Functional Characteristics 7-3
Major System Component Descriptions..... 7-7

Models 840, 850, and 860 Functional Descriptions..... 8-1

Central Processor 8-1
Central Memory 8-15
Input/Output Unit 8-23

Models 840, 850, and 860 Operating Instructions..... 9-1

Controls and Indicators 9-1
Power-On and Power-Off Procedures..... 9-10
Operating Procedures 9-10
IOU Reconfiguration 9-12

Model 990 System Description ... 10-1

Physical Characteristics 10-1
Functional Characteristics 10-3
Major System Component Descriptions..... 10-7

Model 990 Functional Descriptions..... 11-1

Central Processor 11-1
Central Memory 11-15
Input/Output Unit 11-23

Model 990 Operating Instructions..... 12-1

Controls and Indicators 12-1

Power-On and Power-Off Procedures 12-6
Operating Procedures 12-6
IOU Reconfiguration 12-7

CYBER 990E, 995E, and 994 System Description 13-1

Physical Characteristics 13-1
Functional Characteristics 13-4
Major System Component Descriptions 13-8

CYBER 990E, 995E, and 994 Functional Descriptions 14-1

Central Processor 14-1
Central Memory 14-15
Input/Output Unit 14-23

CYBER 990E, 995E, and 994 Operating Instructions 15-1

Controls and Indicators 15-1
Power-On and Power-Off Procedures 15-6
Operating Procedures 15-6
IOU Reconfiguration 15-7

Instruction Descriptions 16-1

CP Instruction Formats 16-1
Instructions Description Nomenclature 16-3
CP Operating Modes 16-4
CP Instruction Descriptions 16-4
PP Instruction Descriptions 16-70

Programming Information 17-1

CP Programming 17-1
Memory Programming 17-26
PP Programming 17-29
System Console Programming ... 17-42
Real-Time Clock Programming .. 17-49
Two-Port Multiplexer Programming 17-49
Maintenance Channel Programming 17-56

Figures

1-1. System Configuration 1-1
1-2. Computer System Block Diagram 1-11
2-1. CYBER 170 Exchange Package . 2-9
2-2. Address Format 2-15
2-3. CM Layout 2-22
2-4. Barrel and Slot 2-25
2-5. Formation of Absolute CM Address 2-26
3-1. Deadstart Panel 3-2
3-2. CM Configuration Switches 3-6
4-1. System Configuration Without Memory Upgrade Option 4-2
4-2. System Configuration Without Memory Upgrade Option and With IOU-NIO/CIO Upgrade Option 4-2
4-3. System Configuration With Memory and Optional CP Upgrade Options 4-3
4-4. System Configuration With Memory, Optional CP, and IOU-NIO/CIO Upgrade Options 4-3
4-5. Computer System Block Diagram 4-13
5-1. CYBER 170 Exchange Package . 5-9
5-2. Address Format Without Memory Upgrade Option 5-15
5-3. Address Format With Memory Upgrade Option 5-16
5-4. CM Layout 5-24
5-5. Barrel and Slot 5-27
5-6. Formation of Absolute CM Address 5-28
6-1. Deadstart Panel 6-2
6-2. CM Configuration Switches 6-7
6-3. Reconfiguration Examples (With Memory Upgrade Option) 6-13
7-1. System Configuration 7-2
7-2. System Configuration With Optional IOU-NIO/CIO 7-2
7-3. Computer System Block Diagram 7-12
8-1. CYBER 170 Exchange Package . 8-9
8-2. Address Format 8-15
8-3. CM Layout 8-22

8-4. Barrel and Slot 8-25
8-5. Formation of Absolute CM Address 8-26
9-1. Deadstart Panel 9-2
9-2. CM Configuration Switches 9-6
9-3. Reconfiguration Examples 9-9
10-1. System Configuration 10-2
10-2. Computer System Block Diagram 10-12
11-1. CYBER 170 Exchange Package 11-9
11-2. Address Format 11-15
11-3. CM Layout 11-22
11-4. Barrel and Slot 11-27
11-5. Formation of Absolute CM Address 11-26
12-1. Deadstart Options Display 12-1
12-2. Initial Deadstart Display 12-2
12-3. CM Configuration Switches ... 12-4
13-1. System Configuration 13-2
13-2. Computer System Block Diagram 13-13
14-1. CYBER 170 Exchange Package 14-9
14-2. Address Format (CYBER 990E and 995E) 14-15
14-3. Address Format (CYBER 994) 14-22
14-4. CM Layout 14-22
14-5. Barrel and Slot 14-27
15-1. Deadstart Options Display 15-1
15-2. Initial Deadstart Display 15-2
15-3. CM Configuration Switch 15-4
16-1. CP Instruction Parcel Arrangement 16-2
16-2. PP Instruction Formats 16-72
16-3. PP Data Format 16-72
16-4. PP Relocation (R) Register Format 16-73
17-1. CYBER 170 Exchange Package 17-2
17-2. Floating-Point Format 17-5
17-3. Floating-Add Result Format .. 17-11
17-4. Multiply Result Format 17-11
17-5. Format of Exit Condition Register at (RAC) 17-16
17-6. Memory Map 17-27
17-7. Channel Transfer Timing 17-37
17-8. Data Input Sequence Timing . 17-39
17-9. Data Output Sequence Timing 17-41
17-10. Display Station Output Function Code 17-45
17-11. Cooordinate Data Word 17-46
17-12. Character Data Word 17-46
17-13. Receive and Display Program Flowchart 17-48

Tables

2-1. Port Priority 2-16
2-2. SECDED Syndrome Codes/Corrected Bits 2-18
3-1. Deadstart Panel Controls/Indicators 3-3
3-2. Central Memory Reconfiguration 3-7
3-3. Barrel Numbering Table 3-11
5-1. Port Priority 5-17
5-2. SECDED Syndrome Codes/Corrected Bits 5-20
6-1. Deadstart Panel Controls/Indicators 6-3
6-2. Central Memory Reconfiguration (Without Memory Upgrade Option) 6-8
6-3. Central Memory Reconfiguration (With Memory Upgrade Option) 6-10
6-4. Barrel Numbering Table 6-16
8-1. Port Priority 8-16
8-2. SECDED Syndrome Codes/Corrected Bits 8-18
9-1. Deadstart Panel Controls/Indicators 9-3
9-2. Central Memory Reconfiguration 9-7
9-3. Barrel Numbering Table 9-12
11-1. Port Priority 11-16
11-2. SECDED Syndrome Codes/Corrected Bits 11-18
11-3. IOU Maintenance Registers .. 11-33
12-1. Deadstart Options Display 12-3
12-2. Deadstart Display Operator Entries and Functions 12-3
12-3. Central Memory Reconfiguration 12-5
12-4. PP and Barrel Reconfiguration Example, RP=0 12-8

12-5. PP and Barrel Reconfiguration Example, RP=2 12-12
14-1. Port Priority 14-16
14-2. SECDED Syndrome Codes/Corrected Bits 14-18
14-3. IOU Maintenance Registers .. 14-33
15-1. Deadstart Options Display 15-3
15-2. Deadstart Display Operator Entries and Functions 15-3
15-3. Central Memory Reconfiguration 15-5
15-4. PP and Barrel Reconfiguration Example, RP=0 15-8
15-5. PP and Barrel Reconfiguration Example, RP=2 15-12
16-1. CP Instruction Designators ... 16-3
16-2. CP Integer Arithmetic Instructions 16-5
16-3. CP Branch Instructions 16-9
16-4. CP Block Copy Instructions .. 16-16
16-5. CP Shift Instructions 16-19
16-6. CP Logical Instructions 16-23
16-7. CP Floating-Point Instructions 16-27
16-8. CP Jump Instructions 16-39
16-9. CP Exchange Jump Instructions 16-41
16-10. CP Compare/Move Instructions 16-44
16-11. Collate Table 16-49
16-12. CP Set Instructions 16-50
16-13. CP Normalize Instructions .. 16-62
16-14. CP Pass Instructions 16-65
16-15. CP Illegal Instructions 16-66
16-16. PP Nomenclature 16-71
16-17. PP Load/Store Instructions .. 16-74
16-18. PP Arithmetic Instructions .. 16-78
16-119. PP Logical Instructions 16-82
16-20. PP Replace Instructions 16-86
16-21. PP Branch Instructions 16-90
16-22. PP Central Memory Access Instructions 16-96
16-23. PP I/O Instructions 16-100
16-24. Other IOU Instructions 16-106
16-25. PP Instruction Timing 16-109
17-1. Bits 58 and 59 Configurations 17-5
17-2. Xj Plus Xk (30, 32, and 34 Instructions) 17-8
17-3. Xj Minus Xk (31, 33, and 35 Instructions) 17-9
17-4. Xj Multiplied by Xk (40, 41, and 42 Instructions) 17-9
17-5. Xj Divided by Xk (44 and 45 Instructions) 17-10
17-6. Contents of Exit Condition Register at (RAC) 17-16
17-7. Error Exits in CYBER 170 Monitor Mode (MF=1) 17-17
17-8. Error Exits in CYBER 170 Job Mode (MF=0) 17-20
17-9. Keyboard Character Codes ... 17-43
17-10. Display Character Codes 17-44
17-11. MCH Function Word Bit Assignments (CP and CM) 17-58
17-12. MCH Function Word Bit Assignments, All Models Except 990 and CYBER 990E, 995E and 994 (IOU) 17-60
17-13. MCH Function Word Bit Assignments, Model 990 and CYBER 990E, 995E, and 994 (IOU) 17-60
17-14. CP Internal Address Assignments 17-64
17-15. CM Internal Address Assignments 17-64
17-16. IOU Internal Address Assignments 17-65

About This Manual

This manual contains hardware reference information for the CDC® CYBER 170 Models 835, 845, and 855 Computer Systems; CYBER 180 Models 835, 840, 845, 850, 855, 860, and 990 Computer Systems; and CYBER 990E, 995E, and 994 Computer Systems.

Audience

This manual is for use by customer, marketing, training, programming, and Engineering Services personnel who operate, program, and maintain the computer systems.

Organization

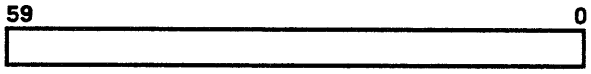
The manual describes the functional, operational, and programming characteristics of the computer system hardware. Additional hardware reference information is available in the publications listed in the system publication index.

Chapters 1 through 15 contain the following information for each of the CYBER computer systems:

- System Description
- Functional Descriptions
- Operating Instructions

Chapter 16 contains instruction descriptions, and chapter 17 contains programming information applicable to all the CYBER computer systems. Appendix A contains the glossary.

There are two methods used within this manual to designate bit numbers. In the majority of the manual, bits are numbered 59 through 0 reading from left to right.



However, in the context of the two-port multiplexer and maintenance registers, bits are numbered 0 through 63 from left to right.

FCC Compliance

This equipment generates, uses and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. As temporarily permitted by regulation, it has not been tested for compliance with the limits for Class A computing devices pursuant to Subpart J of Part 15 of the FCC Rules which are designed to provide reasonable protection against such interference. Operation of this equipment in a residential area is likely to cause interference in which case the user at his/her own expense will be required to take whatever measures may be required to correct the interference.

Conventions

New features, as well as technical changes, deletions, and additions to this manual are indicated by vertical bars in the margins.

Related Manuals

Additional system hardware information is available in manuals listed in the system publication index.

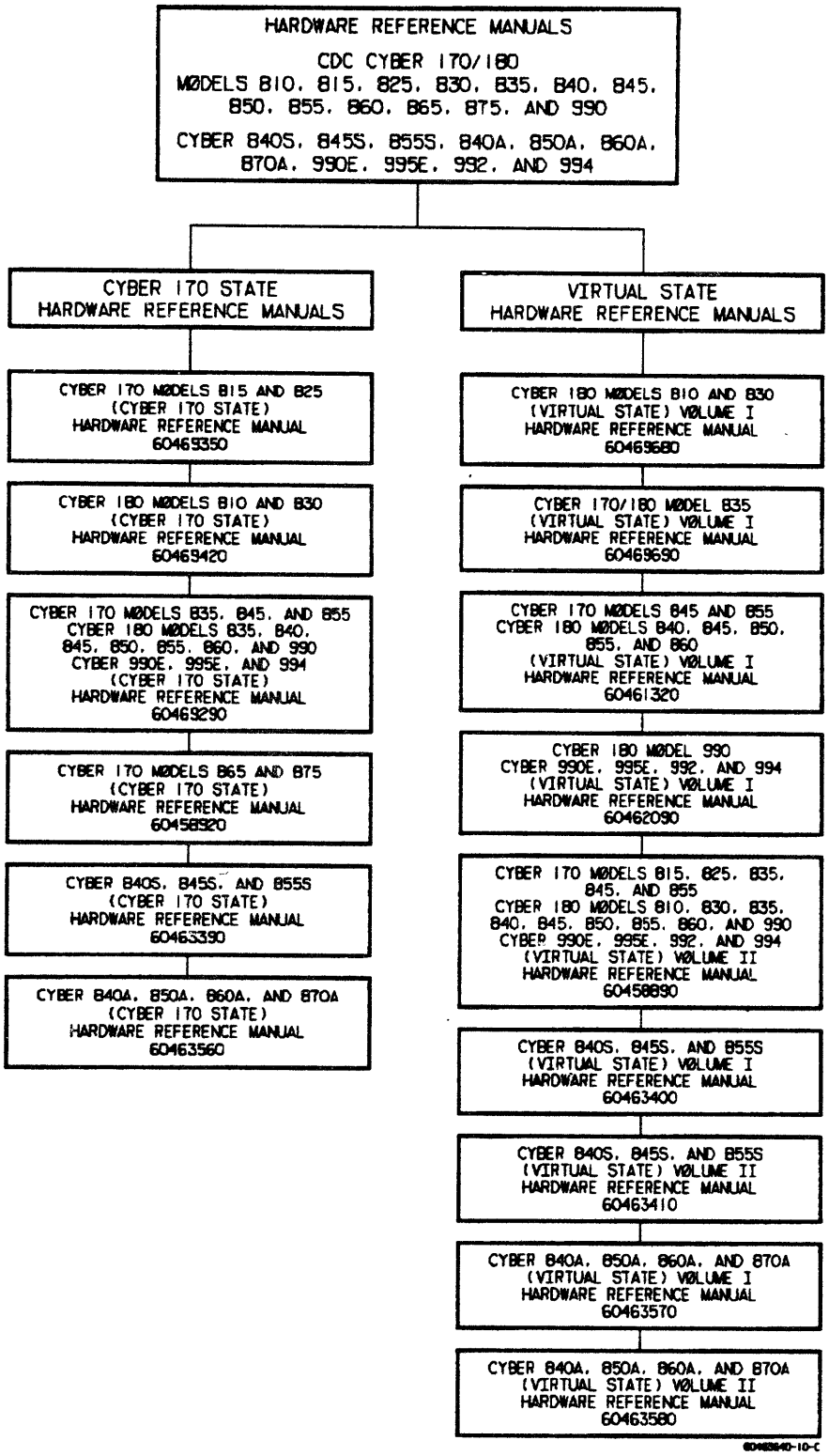
Additional Related Manuals

Other manuals that are applicable to the CYBER 170, CYBER 180, and CYBER 990E, 995E, and 994 Computer Systems but not listed in the system publication index are:

Title	Publication Number
NOS Version 2 Operator/Analyst Handbook	60459310
NOS Version 2 Systems Programmer's Instant	60459370
NOS Version 1 Operator's Guide	60457700
NOS Version 1 Systems Programmer's Instant	60457790
NOS/BE Version 1 Operator's Guide	60457380
NOS/BE Version 1 System Programmer's Reference Manual, Volume 1	60458480
NOS/BE Version 1 System Programmer's Reference Manual, Volume 2	60458490
Maintenance Register Codes Booklet	60458110
Codes Booklet	60458100
CDC 721 Enhanced Display Terminal (CC634-B) Hardware Reference Manual	62950102
CYBER Initialization Package (CIP) Reference Manual	60457180
CDC 19003 System Console (CC598-A/B) Operations and Maintenance Guide	60463610

Publication ordering information and latest revision levels are available from the Literature Distribution and Services catalog, publication number 90310500.

SYSTEM PUBLICATION INDEX



Ordering Manuals

Control Data manuals are available through Control Data sales offices or through:

Control Data Corporation
Literature and Distribution Services
308 North Dale Street
St. Paul, Minnesota 55103

Submitting Comments

Control Data welcomes your comments about this manual. Your comments may include your opinion of the usefulness of this manual, your suggestions for specific improvements, and the reporting of any errors you have found.

You can submit your comments on the comment sheet on the last page of this manual. If the manual has no comment sheet, mail your comments to:

Control Data Corporation
Technology and Publications Division ARH219
4201 Lexington Avenue North
St. Paul, Minnesota 55126-6198

Please indicate whether you would like a written response.

Disclaimer

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features and parameters.

Model 835 System Description 1

Physical Characteristics 1-1

Functional Characteristics 1-2
Central Processor 1-3
Central Memory 1-4
Input/Output Unit 1-5

Major System Component Descriptions 1-6
Central Processor 1-6
Instruction Section 1-6
Registers 1-6
Execution Section 1-7
Cache Memory 1-7
Addressing Section 1-7
Central Memory 1-8
Input/Output Unit 1-9
System Console 1-10

Model 835 System Description 1

This chapter describes the physical and functional characteristics and major system components.

This high-speed computer system is for both business and scientific applications. The system includes the following components.

- Central processor (CP)
- Central memory (CM)
- Input/output unit (IOU)

Physical Characteristics

The system configuration (figure 1-1) includes a three-section cabinet for the CP, CM, and IOU. (The system console is also required for system operation.)

Each cabinet section contains a logic chassis with plug-in circuit boards. The logic chassis in the IOU also contains a deadstart panel with initialization and maintenance controls and displays. Each cabinet section also contains a self-contained cooling unit to cool the logic chassis, an ac/dc control section with voltage margin testing facilities, and dc power supplies. For additional cooling or power information, refer to the cooling system and power system manuals listed in the system publication index in About This Manual.

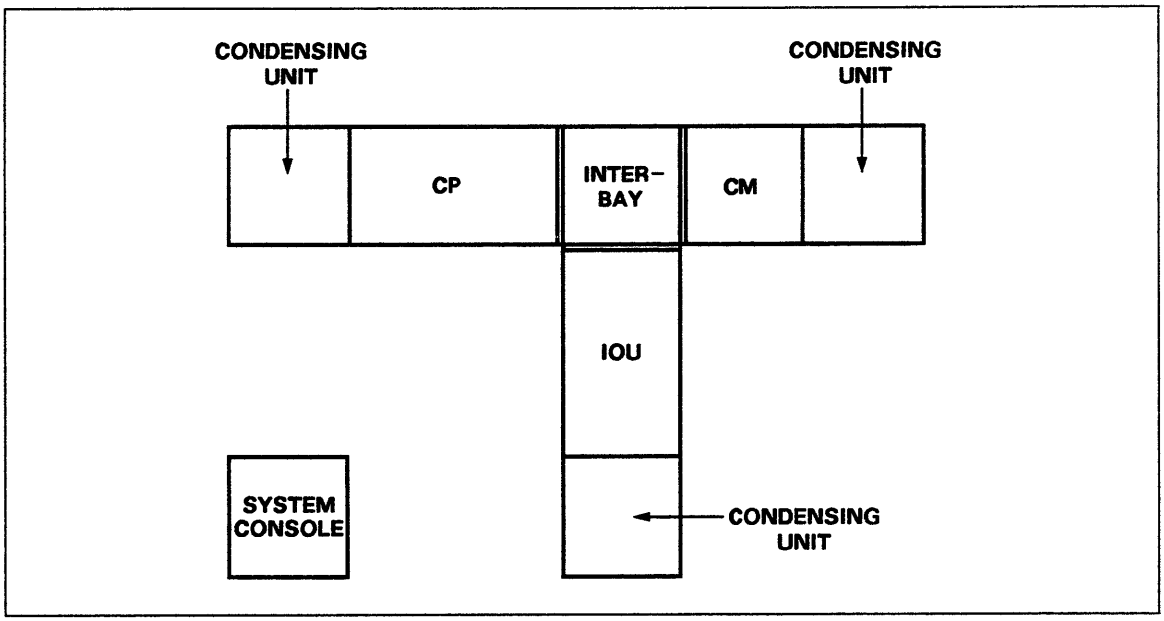


Figure 1-1. System Configuration

Functional Characteristics

To achieve high computation speeds, emitter-coupled logic (ECL) is used.

High speed is also the objective of the CP design, which is based on the assumption that both data and instructions are, in most cases, accessed from successive memory locations. Accordingly, the CP prefetches both instructions and data expected to be used next while the current instruction is being processed.

The CP supports two states of operation.

Virtual State	Operates with virtual-memory byte addressing, using the Virtual State instruction set and data formats. Virtual State is the native operating state of the CP.
CYBER 170 State	Operates with real-memory word addressing, using the CYBER 170 State CP instruction set and data formats.

The Virtual State and CYBER 170 State environments may be present at the same time with the CP executing in either environment.

- NOS/VE is the operating system of Virtual State.
- NOS is the operating system of CYBER 170 State.

The semiconductor central memory is divided into eight independent banks. These banks may all be simultaneously in the process of completing read/write requests which are queued and distributed at ECL speeds. System input/output speeds are determined by the capabilities of existing external devices.

Central Processor

The CP has the following characteristics:

- 60-bit internal word.
- Eight 60-bit operand (X) registers.
- Eight 18-bit address (A) registers.
- Eight 18-bit index (B) registers.
- Two registers that isolate each user's central memory space (RAC, FLC).
- Two registers that isolate each user's extended memory space (RAE, FLE).
- Register exchange instructions (exchange jumps) for interrupting programs.
- Floating-point (FP) arithmetic (10-bit exponent plus sign bit, 48-bit coefficient plus sign bit). Some FP instructions use 96-bit (double-precision) coefficients.
- Integer arithmetic (60/18-bit operands).
- Character string compare/move facilities (6-bit characters).
- Packed instructions (15/30/60-bit instructions in 60-bit words).
- Synchronous internal logic.
- 56-nanosecond clock period.
- 2048-word cache buffer memory, option available for 4096-word cache.
- Instruction and branch instruction lookahead.
- Microcode control.
- Parity checking of all major data and address paths.
- Maintenance channel to IOU.

Central Memory

The CM has the following characteristics:

- 72-bit data word (60 data bits, 8 single-error correction/double-error detection bits, and 4 unused bits).
- 524K words of refresh-type semiconductor memory, options available to 2097K words.
- Organization of eight independent banks.
- Two memory ports.
- Bounds register to limit write access.
- 56-nanosecond clock period.
- Maximum data transfer rate of one word every 56 nanoseconds.
- 672-nanosecond read access time.
- 448-nanosecond read/write cycle time.
- 896-nanosecond partial write cycle time.
- Read and write data queuing capability.
- Single-error correction/double-error detection (SECDED) on stored data.
- Parity checking of all major data, address, and control paths.
- Unified-extended memory (UEM) which serves as extended memory within CM.

Input/Output Unit

The IOU has the following characteristics:

- Ten peripheral processors (PPs), 15-PP/20-PP options available. Each PP has 4K independent memory (PPM) comprised of 16-bit words with the upper 4 bits zero.
- Port to central memory.
- Bounds register to limit writes to central memory.
- Twelve 12-bit CYBER 170 channels to external devices, 24 channel option available.
- Real-time clock (channel 14_8).
- Display controller (CYBER 170 channel 10_8).
- Two-port multiplexer (channel 15_8).
- Maintenance channel (channel 17_8).
- Parity checking on all major data and address paths.
- Operating speed of 250 nanoseconds and a minor cycle of 50 nanoseconds.

Major System Component Descriptions

Central Processor

The CP hardware (figure 1-2) consists of the following:

- **Instruction section**
- **Registers**
- **Execution section**
- **Cache memory**
- **Addressing section**

The CP is isolated from the IOU and is thus able to carry on computation or character manipulation unencumbered by I/O requirements.

Instruction Section

The instruction section directs the arithmetic and manipulative functions for instruction execution. The instruction section prefetches instruction words from memory and disassembles them into instructions.

Registers

Operating registers reduce storage accesses for operands used during the execution of an instruction. These registers are:

- Eight 60-bit X registers (X0 through X7) which hold operands used for computation.
- Eight 18-bit A registers (A0 through A7) which use A0 primarily for indexing and A1 through A7 for CM operand addressing.
- Eight 18-bit B registers (B0 through B7) which are primarily indexing registers to control program execution. The B0 register always contains all zeros.

Eight support registers support the operating registers during program execution. These registers are:

- 18-bit program address (P) register.
- 21-bit reference address for CM (RAC) register. This is a program's lower bound.
- 21-bit field length for CM (FLC) register. This is a program's upper bound.
- 6-bit exit mode (EM) register.

- 6-bit flag register.
- 21-bit reference address for UEM (RAE) register.
- 24-bit field length for UEM (FLE) register.
- 18-bit monitor address (MA) register.

The registers store data and control information, present operands to the execution section, and store results.

The operating and support registers reside in the registers section.

Execution Section

The execution section combines the operands to achieve the result.

Cache Memory

The cache memory consists of two sets of fast bipolar memory, capable of storing 2048 60-bit words. It can be expanded to four sets with a capacity of 4096 words. The memory addressing sections determine whether a requested word is in the cache memory. If it is not, they read four consecutive words from central memory into the cache memory.

Addressing Section

The addressing section checks memory addresses against the CP registers RAC, FLC, RAE, and FLE to ensure isolation of user memory space.

Central Memory

The CM (figure 1-2) consists of the following:

- Eight memory banks
- Memory ports

The CM without the Memory Upgrade Option is a refresh-type metal oxide semiconductor (MOS) memory organized into eight independent banks.

A portion of CM can be reserved for use as extended memory. It is called unified extended memory (UEM), and is referenced by the RAE and FLE registers. The UEM operates in 24-bit standard addressing mode. All memory ports have queuing buffers.

Input/Output Unit

The IOU (figure 1-2) consists of the following:

- Ten logically independent peripheral processors (PPs). Options are available to increase total to 15 or 20 PPs.
- Internal interface to 12 I/O channels. 24-channel option is available.
- External interfaces to I/O channels:
  - 11 or 23 CYBER 170 channel interfaces.
  - Display controller interface (CYBER 170 channel 10_8).
  - Real-time clock interface (channel 14_8).
  - Two-port multiplexer interface (channel 15_8).
  - Maintenance channel interface (channel 17_8).
- Interface to central memory.
- Bounds register to limit writes to CM.
- Cache invalidation bus interface to CP.

The PPs are organized in groups of five, called barrels. The PPs in a barrel time-share common hardware. Each PP has its own independent memory and communicates with all I/O channels and central memory.

System Console

The system console, required for system operation, provides a visual, alphanumeric readout for the computer. The receipt of symbol and position information from the computer enables displaying program information on a cathode-ray tube (CRT). The station also contains an alphanumeric keyboard which enables an operator to send data to the computer. The keyboard and CRT combination permits the computer operator to monitor and control system operation. Except for programming information in chapter 17, refer to the system console manual listed in the system publication index in About This Manual for further information.

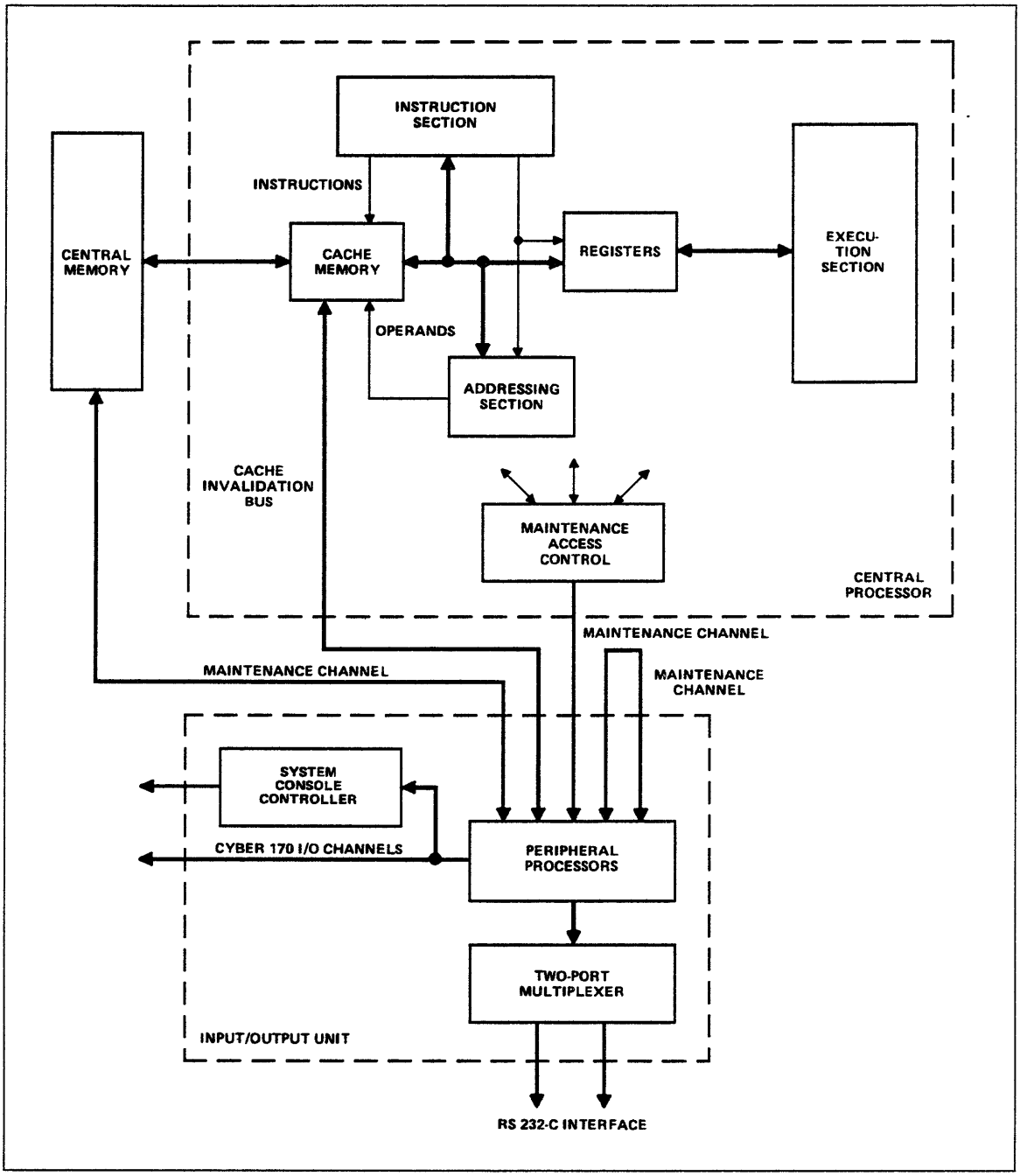


Figure 1-2. Computer System Block Diagram

Model 835 Functional Descriptions 2

Central Processor . . . 2-1
Instruction Section . . . 2-1
Instruction Lookahead . . . 2-1
Maintenance Access Control . . . 2-1
Instruction Control Sequences . . . 2-2
Boolean Sequence . . . 2-2
Shift Sequence . . . 2-3
Floating-Add Sequence . . . 2-4
Floating-Multiply and Floating-Divide Sequence . . . 2-4
Increment Sequence . . . 2-4
Compare/Move Sequence . . . 2-6
CYBER 170 Exchange Sequence . . . 2-7
Block Copy Sequence . . . 2-7
Direct Read/Write Sequence . . . 2-7
Normal Jump Sequence . . . 2-8
Return Jump Sequence . . . 2-8
Registers . . . 2-9
Operating Registers . . . 2-10
X Registers . . . 2-10
A Registers . . . 2-10
B Registers . . . 2-10
Support Registers . . . 2-11
P Register . . . 2-11
RAC Register . . . 2-11
FLC Register . . . 2-11
EM Register . . . 2-12
Flag Register . . . 2-12
RAE Register . . . 2-13
FLE Register . . . 2-13
MA Register . . . 2-13
Execution Section . . . 2-14
Cache Memory . . . 2-14
Addressing Section . . . 2-14

Central Memory . . . 2-15
Address Format . . . 2-15
CM Access and Cycle Times . . . 2-16
CM Ports and Priorities . . . 2-16
SECDED . . . 2-17
CM Layout . . . 2-22
CM Bounds Register . . . 2-22
Central Memory Reconfiguration . . . 2-22

Input/Output Unit . . . 2-23
Peripheral Processor . . . 2-23
Deadstart . . . 2-24
Barrel and Slot . . . 2-24
PP Registers . . . 2-26
R Register . . . 2-26
A Register . . . 2-26
P Register . . . 2-27
Q Register . . . 2-27
K Register . . . 2-27

PP Numbering 2-28
PP Memory 2-28
I/O Channels 2-28
Real-Time Clock 2-29
Two-Port Multiplexer 2-29
Maintenance Channel 2-29
Central Memory Access 2-29

Model 835 Functional Descriptions 2

This chapter provides functional descriptions of the central processor (CP), central memory (CM), and input/output unit (IOU) as shown in the system block diagram in chapter 1. Functional descriptions for the system display station and the cooling system are in their respective manuals listed in the system publication index in About This Manual.

Central Processor

The CP consists of the instruction section, registers, the execution section, cache memory, and the addressing section.

Instruction Section

The instruction section consists of logic for instruction control.

Instruction Lookahead

The instruction lookahead hardware (ILH) prefetches instruction words to make the next instruction immediately available when the execution of the previous instruction is complete; for example, during conditional branch instructions. To accomplish this, ILH reads instructions from cache/CM into a three-word, first-in, first-out buffer.

When ILH detects a conditional branch, it reads two instruction words from cache/CM, starting at the target address, into a branch buffer, and holds them until the branch is resolved. If the branch takes place, the branch buffer contains the next two executable instruction words; if not, ILH purges the branch buffer and processing continues with the next instruction in the three-word buffer.

Maintenance Access Control

The maintenance access control performs initialization and maintenance operations in the CP.

Instruction Control Sequences

The instruction control section performs instruction translation and control sequences. Each control sequence obtains the necessary instruction operands from the operating registers and provides the control signals for execution. Instructions read from CM are 60-bit instruction words that are in four 15-bit and 30-bit groups. The 15-bit groups are termed parcels with the first parcel (parcel 0) being the highest-order 15 bits of a 60-bit CM word. Second, third, and fourth parcels (parcels 1, 2, and 3) follow in order. The 30-bit groups contain two 15-bit parcels.

The instruction control sequences control the execution of one or more instructions of a common type. These sequences and associated instructions are briefly described in this section. For further information, refer to CP Instruction Descriptions in chapter 16.

Boolean Sequence

The Boolean sequence controls instructions that require bit-by-bit data manipulation. This includes both the logical and transmissive operations. The instructions requiring logical operations are:

11	Logical product (Xj) and (Xk) to Xi	BXi Xj * Xk
12	Logical sum of (Xj) and (Xk) to Xi	BXi Xj + Xk
13	Logical difference of (Xj) and (Xk) to Xi	BXi Xj - Xk
15	Logical product of (Xj) with complement of (Xk) to Xi	BXi -Xk * Xj
	Logical sum of (Xj) with complement of (Xk) to Xi	BXi -Xk + Xj
17	Logical difference of (Xj) with complement of (Xk) to Xi	BXi -Xk - Xj

The instructions requiring transmissive operations are:

10	Transmit (Xj) to Xi	BXi Xj
11	Transmit complement of (Xk) to Xi	BXi -Xk

Shift Sequence

The shift sequence controls instructions that require shifting the 60-bit field of data within the operand word. The shift instructions are:

20	Left shift (Xi) by jk	LXi jk
21	Right shift (Xi) by jk	AXi jk
22	Left shift (Xk) nominally (Bj) places to Xi	LXi Bj, Xk
23	Right shift (Xk) nominally (Bj) places to Xi	AXi Bj, Xk
43	Form mask of jk bits to Xi	MXi jk

The shift sequence also controls the pack and unpack instructions. In the packed floating format, the coefficient is contained in the lower 48 bits. The sign and biased exponents are contained in the upper 12 bits. The unpack instruction obtains the packed word from the Xk register, delivers the coefficient to the Xi register, and delivers the exponent to the Bj register. The unpack and pack instructions are:

26	Unpack (Xk) to Xi and Bj	UXi Bj, Xk
27	Pack (Xk) and (Bj) to Xi	PXi Bj, Xk

The shift sequence also controls the normalize operations. The coefficient portion of the operand is repositioned, and the exponent is adjusted so that the most significant bit of the coefficient is in the highest-order bit position of the coefficient, and the exponent is decreased by the number of bit positions shifted. The normalize instructions are:

24	Normalize (Xk) to Xi and Bj	NXi Bj, Xk
25	Round normalize (Xk) to Xi and Bj	ZXi Bj, Xk

Floating-Add Sequence

The floating-add sequence controls the operations necessary to form the 48-bit floating sum with a 12-bit exponent of the floating-point sum or difference of two floating-point operands. The floating-add instructions are:

30	Floating sum of (Xj) and (Xk) to Xi	FXi Xj + Xk
31	Floating difference of (Xj) and (Xk) to Xi	FXi Xj - Xk
32	Floating double-precision sum of (Xj) and (Xk) to Xi	DXi Xj + Xk
33	Floating double-precision difference of (Xj) and (Xk) to Xi	DXi Xj - Xk
34	Round floating sum of (Xj) and (Xk) to Xi	RXi Xj + Xk
35	Round floating difference of (Xj) and (Xk) to Xi	RXi Xj - Xk

Floating-Multiply and Floating-Divide Sequence

The floating-multiply and floating-divide sequence controls the operation of floating-multiply, floating-divide, and population-count instructions.

The multiply instructions are:

40	Floating product of (Xj) and (Xk) to Xi	FXi Xj * Xk
41	Round floating product of (Xj) and (Xk) to Xi	RXi Xj * Xk
42	Floating double-precision product of (Xj) and (Xk) to Xi	DXi Xj * Xk

The divide instructions are:

44	Floating divide (Xj) by (Xk) to Xi	FXi Xj/Xk
45	Round floating divide (Xj) by (Xk) to Xi	RXi Xj/Xk

The population-count instruction counts the number of one bits in a 60-bit operand. The instruction is:

47	Population count of (Xk) to Xi	CXi Xk

Increment Sequence

The increment sequence controls the ones complement addition and subtraction of 18-bit fixed-point operands for increment instructions 50 through 77. The sequence also controls the 60-bit ones complement sum and difference values for long-add instructions 36 and 37.

The increment instructions are:

50	Set Ai to (Aj) + K	SAi Aj + K
51	Set Ai to (Bj) + K	SAi Bj + K
52	Set Ai to (Xj) + K	SAi Xj + K
53	Set Ai to (Xj) + (Bk)	SAi Xj + Bk
54	Set Ai to (Aj) + (Bk)	SAi Aj + Bk
55	Set Ai to (Aj) - (Bk)	SAi Aj - Bk
56	Set Ai to (Bj) + (Bk)	SAi Bj + Bk
57	Set Ai to (Bj) - (Bk)	SAi Bj - Bk
60	Set Bi to (Aj) + K	SBi Aj + K
61	Set Bi to (Bj) + K	SBi Bj + K
62	Set Bi to (Xj) + K	SBi Xj + K
63	Set Bi to (Xj) + (Bk)	SBi Xj + Bk
64	Set Bi to (Aj) + (Bk)	SBi Aj + Bk
65	Set Bi to (Aj) - (Bk)	SBi Aj - Bk
66	Set Bi to (Bj) + (Bk)	SBi Bj + Bk
67	Set Bi to (Bj) - (Bk)	SBi Bj - Bk
70	Set Xi to (Aj) + K	SXi Aj + K
71	Set Xi to (Bj) + K	SXi Bj + K
72	Set Xi to (Xj) + K	SXi Xj + K
73	Set Xi to (Xj) + (Bk)	SXi Xj + Bk
74	Set Xi to (Aj) + (Bk)	SXi Aj + Bk
75	Set Xi to (Aj) - (Bk)	SXi Aj - Bk
76	Set Xi to (Bj) + (Bk)	SXi Bj + Bk
77	Set Xi to (Bj) - (Bk)	SXi Bj - Bk

The long-add instructions are:

36	Integer sum of (Xj) and (Xk) to Xi	IXi Xj + Xk
37	Integer difference of (Xj) and (Xk) to Xi	IXi Xj - Xk

Compare/Move Sequence

The compare/move sequence controls data manipulation on a character basis. The compare/move instructions (also referred to as CMU instructions) are 60-bit instructions that use six support registers for source and result field CM addresses and character position offsets. The support registers load from the 60-bit instruction word. The compare/move instructions are:

464	Move indirect (Bj) + K	IM Bj + K
465	Move direct	DM
466	Compare collated	CC
467	Compare uncollated	CU

The support registers are:

- An 18-bit K1 register that specifies which relative CM address word contains the first character of the source data field.
- An 18-bit K2 register that specifies which relative CM address word contains the first character of the result field.
- A 4-bit C1 register that specifies the character position or offset of the first CM word of the source field.
- A 4-bit C2 register that specifies the character position or offset of the first CM word of the result field.
- Two 16-bit L registers (LA and LC) that specify the number of characters in the data field. The LA register is associated with K1, and the LC register is associated with K2. Instruction 464 uses 14 register bits. Instructions 465, 466, and 467 use only the lower eight register bits.

NOTE

CMU instructions are provided for compatibility with previous systems. For better performance, recompile jobs to avoid use of CMU instructions.

CYBER 170 Exchange Sequence

A CYBER 170 exchange sequence is the method used to swap jobs in and out of execution. When a CYBER 170 exchange jump instruction occurs, the CYBER 170 exchange sequence writes the contents of the current job's CP registers (described later in this chapter) into an area of central memory called a CYBER 170 exchange package. A CYBER 170 exchange package is associated with each job. It contains sufficient information to restart a job if the job is interrupted during execution and swapped out by a CYBER 170 exchange jump. To complete the sequence, CP registers for another job are read from its CYBER 170 exchange package and that job begins or resumes execution. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

Block Copy Sequence

The block copy sequence controls the transfer of data between CM and UEM. The number of words to be transferred is determined by the addition of K to the contents of Bj. The starting address for CM is formed by adding either the A0 register or certain bits of the X0 register to the RAC reference address. The starting address for UEM is formed by adding certain bits of the X0 register to the RAE reference address. The block copy instructions are:

011	Block copy Bj + K words from UEM to CM	RE Bj + K
012	Block copy Bj + K words from CM to UEM	WE Bj + K

Direct Read/Write Sequence

Instructions 014 and 015 perform single-word, direct read and write operations for UEM; and instructions 660 and 670 perform single-word, direct read and write operations for central memory.

014	Read one word from UEM at (Xk + RAE) into Xj	RXj Xk
015	Write one word from Xj to UEM at (Xk + RAE)	WXj Xk
660	Read central memory at (Xk) to Xj	CRXj Xk
670	Write Xj into central memory at (Xk)	CWXj Xk

Normal Jump Sequence

The normal jump sequence controls the execution of branch instructions 02 through 07. The 02 instruction performs an unconditional jump to the Bi register address plus K. The branch address is K with i equals zero. The 02 instruction is:

02	Jump to (Bi) + K	JP Bi + K

The conditional jump instructions 03 through 07 branch to address K if the jump condition is met.

These instructions are:

030	Branch to K if (Xj) = 0	ZR Xj, K
031	Branch to K if (Xj) = 0	NZ Xj, K
032	Branch to K if (Xj) is positive	PL Xj, K
033	Branch to K if (Xj) is negative	NG Xj, K
034	Branch to K if (Xj) is in range	IR Xj, K
035	Branch to K if (Xj) is out of range	OR Xj, K
036	Branch to K if (Xj) is definite	DF Xj, K
037	Branch to K if (Xj) is indefinite	ID Xj, K
04	Branch to K if (Bi) = (Bj)	EQ Bi, Bj, K
05	Branch to K if (Bi) ≠ (Bj)	NE Bi, Bj, K
06	Branch to K if (Bi) ≥ (Bj)	GE Bi, Bj, K
07	Branch to K if (Bi) < (Bj)	LT Bi, Bj, K

Return Jump Sequence

The return jump sequence controls the execution of three instructions.

00	Error exit to MA or program stop	PS
010	Return jump to K	RJ K
013	Central exchange jump to (Bj) + K or monitor exchange jump to MA	XJ Bj + K

Registers

The CP contains the operating and support registers described in the following paragraphs. These registers are located in the registers section (refer to figure 1-2).

The contents of these registers can be written into memory and reloaded from memory as a CYBER 170 exchange package by a single CP instruction (CYBER 170 exchange jump). Figure 2-1 shows the CYBER 170 exchange package.

The time a CYBER 170 exchange package resides in CP hardware is called an execution interval. During this interval, the contents of X, A, B, and P registers can be changed by CP instructions. The contents of other support registers change only as a result of a CYBER 170 exchange jump. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

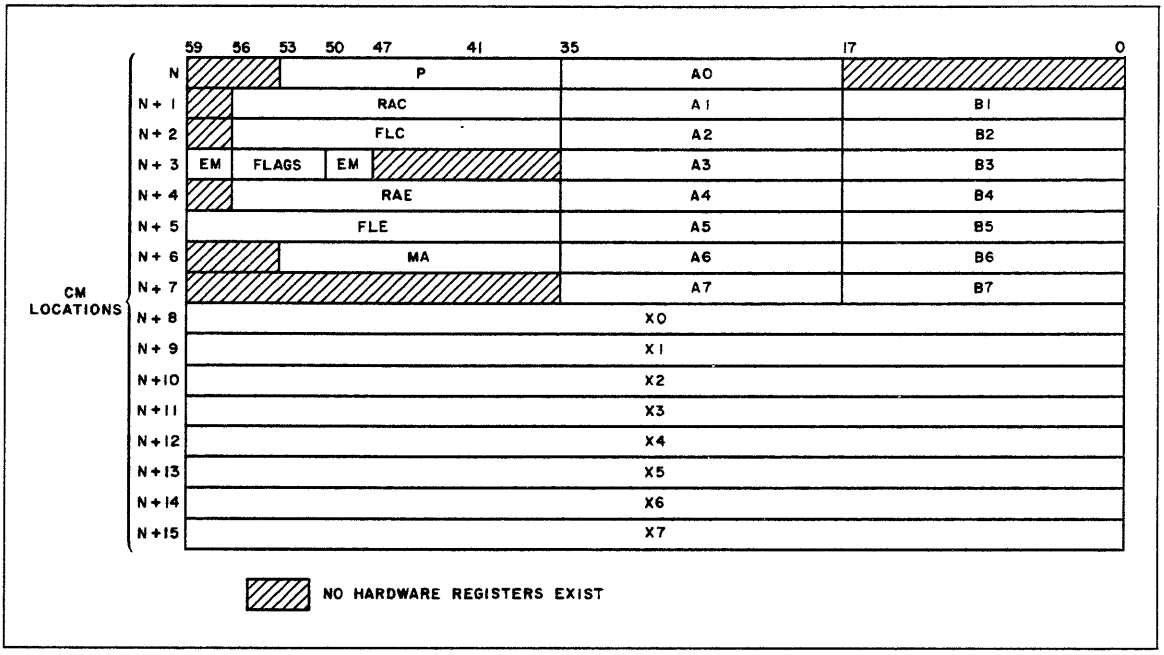


Figure 2-1. CYBER 170 Exchange Package

Operating Registers

The operating registers consist of operand (X), address (A), and index (B) registers. These registers minimize memory references for arithmetic operands and results.

X Registers

The CP contains eight 60-bit X registers, X0 through X7. The X0 register is used in the compare instructions to indicate if two fields of characters are equal. Also, the X0 register provides the relative UEM starting address in a block copy operation.

The X1 through X7 registers are primarily data handling registers for computation. X1 through X5 are used to input data from CM, and X6 and X7 are used to transmit data to CM.

Operands and results transfer between CM and the X registers as a result of placing CM addresses into corresponding A registers.

A Registers

The CP contains eight 18-bit A registers, A0 through A7. The A0 register serves as an intermediate register for the user's discretion. The A0 register is used in the compare collate instruction for the collate table address. Also, the A0 register provides the relative CM starting address in a block copy operation.

The A1 through A7 registers are essentially CM operand address registers associated one-for-one with the X registers. Placing a quantity into an address register (A1 through A5) causes a CM read reference to that address and transmits the CM word to the corresponding X register (X1 through X5). Similarly, placing a quantity into the A6 or A7 register causes the word in the corresponding X6 or X7 register to be written into that relative address of CM.

B Registers

The CP contains eight 18-bit B registers, B0 through B7. These registers are primarily indexing registers to control program execution. Program loop counts may also be incremented or decremented in these registers.

Program addresses may be modified on the way to an A register by adding or subtracting B register quantities. The B registers also hold shift counts for the nominal Bj shifts, the resultant exponent for the unpack, the operand exponent for the pack, and the resultant shift count from a normalize. The B0 register always contains positive zero which can be used as an operand. This register cannot hold results from instructions.

Support Registers

Eight support registers assist the operating registers during the execution of programs. The contents of the support registers are stored in CM, and their new contents are loaded from CM during a CYBER 170 exchange sequence. With the exception of the P register, the contents of the support registers cannot be altered during the execution interval of a CYBER 170 exchange package. When the execution interval completes, the data in the support registers is sent back to CM through a CYBER 170 exchange jump.

P Register

The 18-bit program address (P) register loads from CM during the first word of a CYBER 170 exchange sequence and contains the current program execution address. The register serves as a program address counter and holds the relative CM address for each program step.

RAC Register

The 21-bit CM reference address (RAC) register loads from CM during the second word of a CYBER 170 exchange sequence. An absolute CM address forms by adding RAC to a relative address determined by the instruction. The content of the P register is added to RAC to form the program address in CM. A P-equal-to-zero condition specifies relative address 0 and, therefore, (RAC). This CM location is reserved for recording error exit conditions and should not be used to store data or instructions.

FLC Register

The 21-bit CM field length (FLC) register loads from CM during the third word of a CYBER 170 exchange sequence. The FLC register defines the size of the field of the program in execution. Relative CM addresses are compared with FLC to check that the program is not going out of its allocated memory range.

EM Register

The 6-bit exit mode (EM) register loads from CM during the fourth word of a CYBER 170 exchange sequence. The EM register holds six exit-mode selection bits that control individual error conditions for a program. Selected EM register bits cause the CP to error exit when the corresponding conditions occur. Any or all of the six bits can be set at one time. Clear EM register bits allow the CP to continue, without error processing, when most of the corresponding conditions occur. Refer to the error exit tables under Error Response in chapter 17 for specific cases. The exit mode selection bits appear in the exchange package as bits 48 through 50 and 57 through 59. The bits and their corresponding conditions are:

Mode Selection Bit	Significance
48	Address out of range
49	Infinite operand
50	Indefinite operand
57	Hardware error
58	Hardware error
59	Hardware error

Flag Register

The 6-bit flag register loads from CM during the fourth word of a CYBER 170 exchange sequence. The flag register holds six bits that function as control flags.

Bit	Significance
51	Hardware error bit.
52	Instruction stack (lookahead) purge flag. If set, extended purging of instruction lookahead registers is enabled. For further information, refer to Instruction Lookahead Purge Control under CP Programming in chapter 17.
53	CMU interrupted flag. If set, one of instructions 464 through 467 has been interrupted. The information necessary to resume operation has been saved.
54	Block copy flag. If set, block copy instructions (011 and 012) use bits 30 through 50 of X0 rather than A0 to determine the CM address. For further information, refer to the descriptions of the block copy instructions in chapter 16.
55	Expanded addressing select flag. If set, UEM is operating in expanded addressing mode; if clear, UEM is operating in 24-bit standard addressing mode. This bit must be clear. For further information, refer to Addressing Modes under Memory Programming in chapter 17.
56	UEM enable flag. If set, UEM is available. This flag must be set to allow 011, 012, 014, and 015 instructions to access UEM.

RAE Register

The 21-bit UEM reference address (RAE) register loads from CM during the fifth word of a CYBER 170 exchange sequence. The lower six bits of this register are always zero. An absolute UEM address forms by adding RAE to the relative address which is determined by the instruction.

FLE Register

The 24-bit UEM field length (FLE) register loads from CM during the sixth word of a CYBER 170 exchange sequence. The lower six bits of this register are always zero. The FLE register defines the size of the field in UEM for the program in execution. Relative UEM addresses are compared with FLE.

MA Register

The 18-bit monitor address (MA) register loads from CM during the seventh word of a CYBER 170 exchange sequence. The MA register contains the absolute starting address of an exchange package which is used when executing a central exchange jump (013) instruction with the CYBER 170 monitor flag clear, or when honoring a monitor exchange jump to MA (262x) instruction with the CYBER 170 monitor flag clear. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

Execution Section

The execution section combines the operands into results, providing additional sequencing control where necessary.

Cache Memory

Cache memory is a high-speed buffer memory which is transparent to the user. It reduces effective CM access time by eliminating unnecessary CM references. When the CP first reads CM, a block of four words from CM (containing the requested word) is read rapidly into cache memory. These words may be instructions or data. On subsequent reading of any of these words, CM need not be accessed when these words are in cache memory. Often this is the case because the same data is read more than once, or because a loop of instructions is repeatedly executed. Cache memory is 2048 words or, optionally, 4096 words.

Addressing Section

An address adder calculates memory addresses for data and unconditional jump instructions.

Memory management hardware verifies that memory addresses are to access permitted memory areas. If this is the case, this hardware accesses cache memory and, if necessary, central memory.

Central Memory

The CM performs the following functions.

- Eight memory banks store from 524K to 2097K of 64-bit words (the leftmost 4 bits are undefined) and an 8-bit SECDED code.
- The two ports make CM accessible to the CP and every PP.
- A bounds register limits access to CM from either or both ports.
- The SECDED generators generate the SECDED code bits stored with each word. SECDED checks circuits, correct single-bit errors, and detect double-bit errors.
- The maintenance channel interface gives a PP in the IOU access to the CM maintenance registers for system initialization, corrective action, error reporting and diagnostics, and for setting the port bounds register.

Address Format

Figure 2-2 illustrates the address format.

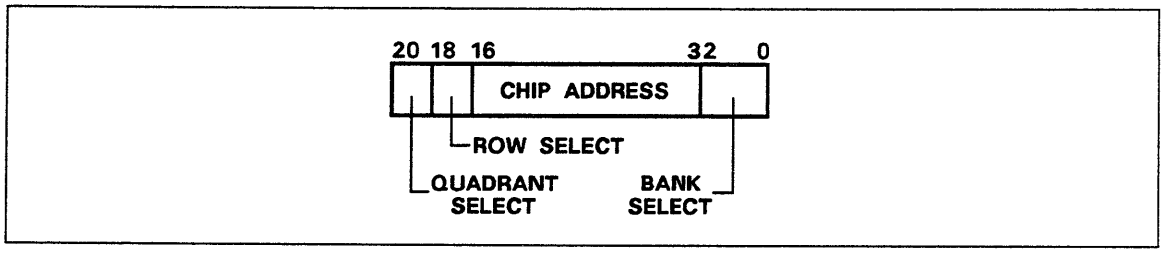


Figure 2-2. Address Format

The following list defines the address fields for figure 2-2.

- Bank Select specifies one of eight banks. Since the bank address is the lowest order 3 bits of the storage address, sequential addressing results in a phased-bank operation which allows a maximum data transfer rate of one word each clock period.
- Chip Address specifies the address of one word in 16K MOS memory chips for the selected bank.
- Row Select selects one of four word rows in a quadrant.
- Quadrant Select selects one of four quadrants. It is used only for storage units larger than 524K.

CM Access and Cycle Times

The following paragraphs list CM access and cycle times. The CM operates on an internal clock period of 64 nanoseconds (major cycle).

The CM access time for a read operation is 672 nanoseconds.

One bank cycle is 8 clock periods (448 nanoseconds). Cycle time for a read or write operation is 448 nanoseconds (8 clock periods). Cycle time for a partial write (read/modify/write) is 896 nanoseconds (16 clock periods).

CM Ports and Priorities

A priority network resolves access conflicts on a rotating basis, preventing long-term lockout of any port. In case of simultaneous requests, the CP has priority. The refresh mechanism may consume a maximum of 6 percent of memory time. Refresh requests have priority over port requests. Refer to table 2-1 for maximum request lockout time in bank cycles.

Table 2-1. Port Priority

Port	Read or Write Requests
Refresh	1
Port 0	4
Port 1	5

Note:

One bank cycle equals 8 clock periods (448 nanoseconds).

SECDED

The SECDED logic corrects single-bit errors during a CM read, permitting unimpeded computer operation. The SECDED logic prepares for the error correction by generating error correction code (ECC) bits for each data word, and by storing these ECC bits in CM with the data word during the CM write. Table 2-2 lists the hexadecimal codes for all the combinations of syndrome bits with the number of the data bit assigned each code or a note categorizing the code. Then, during a CM read, CM performs the following SECDED sequence.

1. Read one CM word and generate new ECC bits for data portion of CM word.
2. Compare new ECC bits with CM word ECC bits.
3. If old and new ECC bits match, no error exists. Send data to requesting unit.
4. If bits do not match, generate syndrome bits from result of ECC compare.
5. Decode syndrome bits to determine if single or multiple bit failure.
6. If single bit failure, correct by inverting failing bit in data word. Send corrected word to requesting unit.
7. If multiple bit or other uncorrectable error, send uncorrectable error response code to CP or IOU. A PP in the IOU may then analyze the syndrome bits using the maintenance channel.

Table 2-2. SECDED Syndrome Codes/Corrected Bits

Code	Bit	Code	Bit	Code	Bit	Code	Bit
00	7	10	67[2]	20	66[2]	30	2/3[6]
01	71[2]	11	3	21	3	31	4
02	70[2]	12	3	22	3	32	4
03	6/7[6]	13	4	23	4	33	3
04	69[2]	14	3	24	3	34	4
05	3	15	4	25	4	35	3
06	3	16	4	26	4	36	3
07	24[1]	17	24[5]	27	28[5]	37	28[1]
08	68[2]	18	3	28	3	38	4
09	3	19	4	29	4	39	3
0A	3	1A	4	2A	4	3A	3
0B	16[1]	1B	16[5]	2B	20[5]	3B	20[1]
0C	4/5[6]	1C	4	2C	4	3C	3
0D	8[1]	1D	8[5]	2D	12[5]	3D	12[1]
0E	0[1]	1E	0[5]	2E	4[5]	3E	4[1]
0F	3	1F	4	2F	4	3F	3

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error with indicated bit(s) inverted.

6. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes.

7. No error detected.

(Continued)

Table 2-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
40	65[2]	50	3	60	3	70	56[1]
41	3	51	4	61	4	71	56[5]
42	3	52	4	62	4	72	60[5]
43	4	53	3	63	3	73	60[1]
44	3	54	4	64	4	74	58[5]
45	4	55	3	65	3	75	58[1]
46	4	56	3	66	3	76	62[1]
47	26[5]	57	26[1]	67	30[1]	77	30/62[5]
48	3	58	4	68	4	78	57[5]
49	4	59	3	69	3	79	57[1]
4A	4	5A	3	6A	3	7A	61[1]
4B	18[5]	5B	18[1]	6B	22[1]	7B	22/61[5]
4C	4	5C	3	6C	3	7C	59[1]
4D	10[5]	5D	10[1]	6D	14[1]	7D	14/59[5]
4E	2[5]	5E	2[1]	6E	6[1]	7E	6/63[5]
4F	4	5F	3	6F	3	7F	63[1]

1. Corrected single-bit error.
2. Syndrome code bit failed (single code bit set).
3. Double error or multiple error (even number of code bits set).
4. Multiple error reported as a single error.
5. Double error or multiple error with indicated bit(s) inverted.

(Continued)

Table 2-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
80	64[2]	90	3	A0	3	B0	48[1]
81	3	91	4	A1	4	B1	48[5]
82	3	92	4	A2	4	B2	52[5]
83	4	93	3	A3	3	B3	52[1]
84	3	94	4	A4	4	B4	50[5]
85	4	95	3	A5	3	B5	50[1]
86	4	96	3	A6	3	B6	54[1]
87	25[5]	97	25[1]	A7	29[1]	B7	29/54[5]
88	3	98	4	A8	4	B8	49[5]
89	4	99	3	A9	3	B9	49[1]
8A	4	9A	3	AA	3	BA	53[1]
8B	17[5]	9B	17[1]	AB	21[1]	BB	21/53[4]
8C	4	9C	3	AC	3	BC	51[1]
8D	9[5]	9D	9[1]	AD	13[1]	BD	13/51[5]
8E	1[5]	9E	1[1]	AE	5[1]	BE	5/55[5]
8F	4	9F	3	AF	3	BF	55[1]

1. Corrected single-bit error.
2. Syndrome code bit failed (single code bit set).
3. Double error or multiple error (even number of code bits set).
4. Multiple error reported as a single error.
5. Double error or multiple error with indicated bit(s) inverted.

(Continued)

Table 2-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
C0	0/1[5]	D0	40[1]	E0	32[1]	F0	2
C1	3	D1	40[4]	E1	32[4]	F1	3
C2	3	D2	44[4]	E2	36[4]	F2	3
C3	2	D3	44[1]	E3	36[1]	F3	2
C4	3	D4	42[4]	E4	34[4]	F4	3
C5	2	D5	42[1]	E5	34[1]	F5	2
C6	2	D6	46[1]	E6	38[1]	F6	2
C7	27[1]	D7	27/46[4]	E7	31/38[4]	F7	2
C8	3	D8	41[4]	E8	33[4]	F8	3
C9	2	D9	41[1]	E9	33[1]	F9	2
CA	2	DA	45[1]	EA	37[1]	FA	2
CB	19[1]	DB	19/45[4]	EB	23/37[4]	FB	23[1]
CC	2	DC	43[1]	EC	35[1]	FC	2
CD	11[1]	DD	11/43[4]	ED	15/35[4]	FD	15[1]
CE	3[1]	DE	3/47[4]	EE	7/39[4]	FE	7[1]
CF	2	DF	47[1]	EF	39[1]	FF	2

1. Corrected single-bit error.

2. Double error or multiple error (even number of code bits set).

3. Multiple error reported as a single error.

4. Double error or multiple error with indicated bit(s) inverted.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes.

CM Layout

Central memory contains an area that is reserved for special software called Virtual State software. Along with the hardware and microcode, this software handles the operations of Virtual State as described in chapter 17. Virtual State software is located at the higher end of memory. The remaining memory is available to the CYBER 170 State and may be allocated as central memory (accessible via RAC and FLC) or as unified extended memory (accessible via RAE and FLE and the 011, 012, 014, and 015 instructions). Refer to figure 2-3.

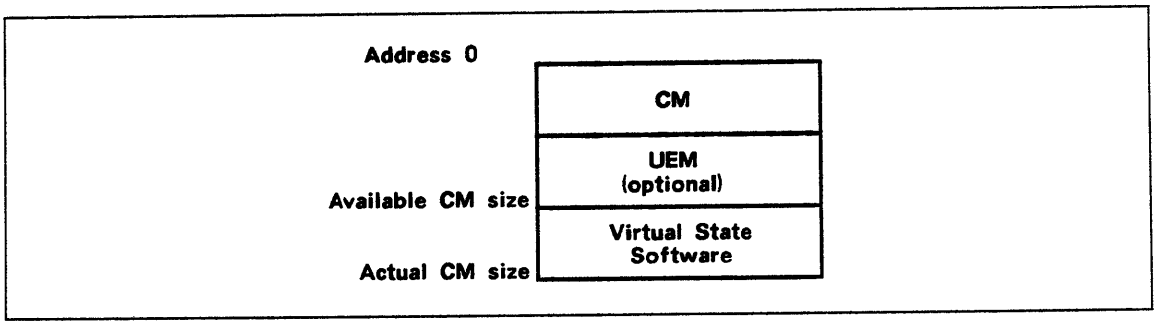


Figure 2-3. CM Layout

CM Bounds Register

The CM bounds register limits the write access to CM from specified ports. The ports are limited to the area between an upper and lower bound as specified in the CM bounds register. Bits in byte 0 specify the port(s) from which the write access is limited. The CM bounds register is set through the maintenance channel. For further information, refer to Maintenance Channel Programming in chapter 17.

Central Memory Reconfiguration

Central memory reconfiguration is a manually performed function that permits the computer operator to restructure the CM addresses so that a failing part of CM can be quickly locked out to provide a continuous block of usable CM. CM reconfiguration is accomplished by setting the switches on the memory unit to manipulate the upper address bits.

When a configuration switch is set forcing a CM address bit to a zero/one, the address range corresponding to the original installed memory accesses some parts of the reconfigured memory more than once. Addresses up to the rightmost forced bit, and half the addresses using the rightmost forced bit, cover a contiguous address space from location 0, which is the reconfigured memory. For further information, refer to chapter 3.

Input/Output Unit

The input/output unit (IOU) performs the functions required to locate, select, and initialize the external devices connected to the system, and controls the transfer of data between a selected device and CM. The IOU also performs system maintenance functions.

The IOU contains the following functional areas.

- Peripheral processor (PP)
- I/O channels
- Real-time clock
- Two-port multiplexer
- Maintenance channel
- CM access

Peripheral Processor

The basic IOU contains 10 PPs and can be expanded to 20 PPs in 5-PP increments. Each PP is a logically independent computer with its own memory. Each 5-PP group is organized into a multiplexing system which allows the PPs to share common hardware for arithmetic, logical, and I/O operations without losing independence. This multiplexing system comprises five ranks of registers termed a barrel. Each rank contains information related to the instruction being executed by one PP.

Each PP can communicate with the CP by issuing a CYBER 170 exchange request to a specific CYBER 170 exchange package associated with the issuing PP. In addition, a PP can also communicate with the CP via CM read and write operations. PPs can communicate with each other over the I/O channels and through CM.

Each PP executes programs alone or with other PPs to control data transfers between external devices and CM. These programs are comprised of instructions from the IOU instruction set and respond to requests issued through CM by the operating system. The programs translate generalized operating system requests into control functions for accessing the external devices and may also perform device scheduling and optimization. The programs use PP memory as a buffer for the data transfer between external devices and CM to isolate IOU data transfer from variations in CM transfer rate.

Deadstart

A deadstart sequence allows the IOU to initialize itself. This deadstart sequence is initiated by the DEAD START switch on the deadstart panel or the DEAD START switch on the system console. The panel includes controls for assigning any PPM to PP0. For further information refer to chapter 3.

Barrel and Slot

The barrel consists of the R, A, P, Q, and K registers, each one of which has five ranks 0 through 4. (Refer to figure 2-4.) Information in these registers moves from one rank to the next at a uniform 20-megahertz rate, providing a multiplexed system of five PPs, each operating at a 4-megahertz rate. The registers are stationary while the PPs rotate. For example, rank 4 registers contain PP0, PP1, PP2, PP3, and PP4 in succession, each consuming 50 nanoseconds of the total cycle time of 250 nanoseconds. Since PP memories operate at a slower rate, independent memory with its own address and data registers is provided for each PP.

Each time data enters the slot, a portion of the instruction for that data is executed. The slot performs tasks such as arithmetic and logic operations and program address manipulation. Complete execution of an instruction may require the R, A, P, Q, and K register quantities to go more than one trip around the barrel and through the slot.

The PPM may be referenced once each time the PP passes around the barrel and through the slot. During its slot time, the PP may also communicate with CM or with any of the I/O channels.

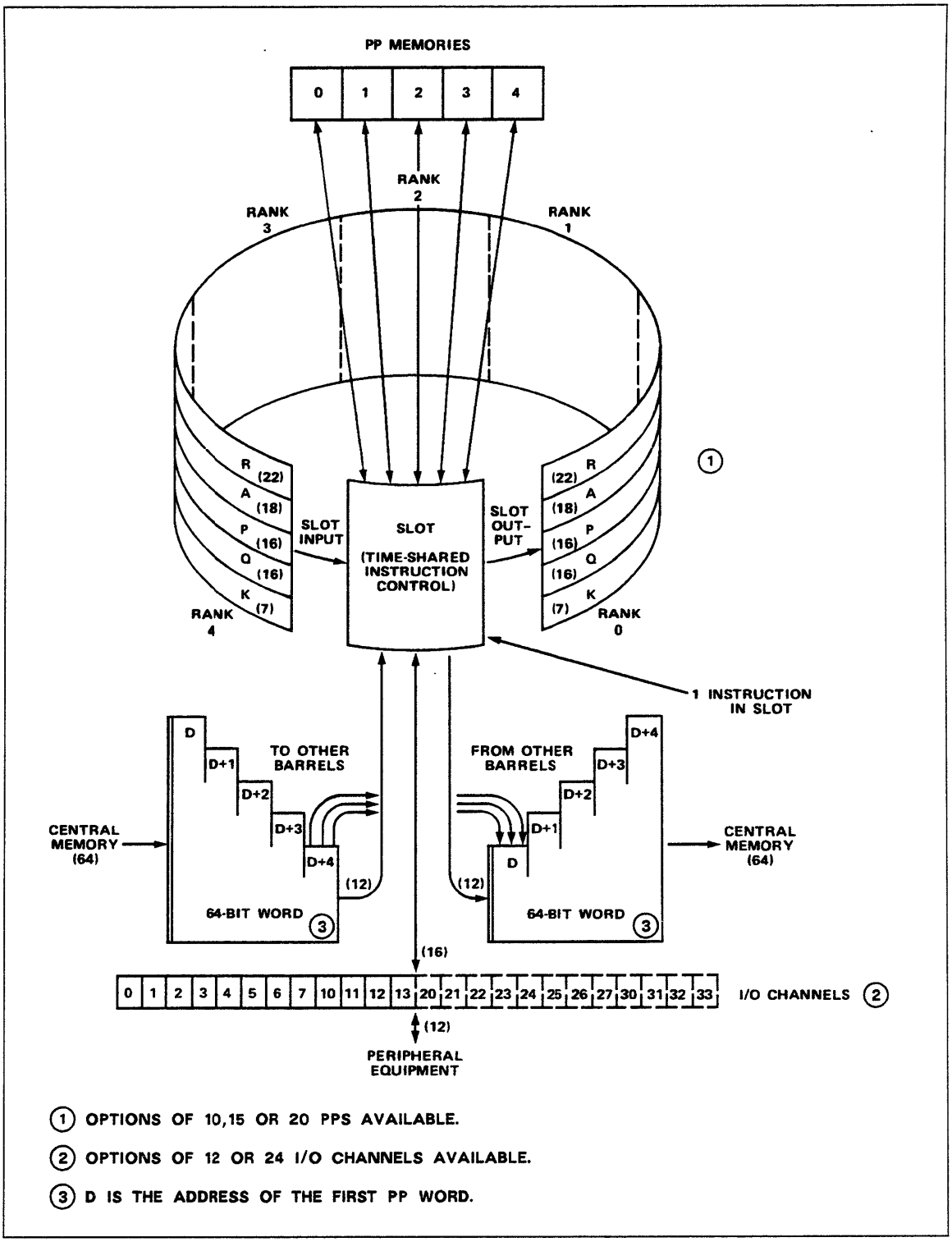


Figure 2-4. Barrel and Slot

PP Registers

The PP registers, which are discussed in the following paragraphs, are:

- R register
- A register
- P register
- Q register
- K register

R Register

The 28-bit R register, in conjunction with the A register, forms an absolute CM address for CM read/write instructions. When bit 17 of the A register is set, the absolute CM address is formed by appending six zeros to the lower end of the contents of the R register and adding to the result bits 0 through 16 of the contents of the A register (refer to figure 2-5).

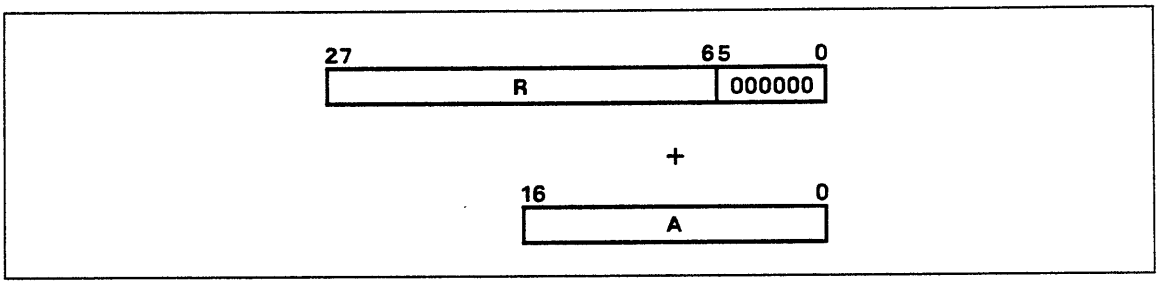


Figure 2-5. Formation of Absolute CM Address

A Register

The 18-bit A register holds one operand for arithmetic, logic, or selected I/O operations. The content of the A register may be an arithmetic operand, CM address, I/O function, or I/O data word. Various instructions operate on 6, 12, 16, or 18 bits of the A register.

When the A register is used as the CM address, parity is generated for transmission with the address to memory control. At deadstart, the A register is set to 10000_8). When bit 17 of the A register is clear, the A register is used as the CM address; however, when bit 17 is set, the R register is added to the A register (as described in the R register description) to obtain the absolute CM address for CM read/write instructions.

P Register

The 12-bit P register is the program address register, except during the execution of instructions 61, 63, 71, and 73. For these instructions, the P register contains the PPM address of the data transfer. At deadstart, the P register is set to zero.

Q Register

The 12-bit Q register holds data for several functions such as the address of the operand during direct addressing and indirect addressing, peripheral address of data used during one-word central read or write instructions, upper 6 bits during constant mode instructions, channel number on all I/O and channel instructions, shift count, and relative jump designator. At deadstart, each rank of the Q register is set to a corresponding PP number. Rank 0 is set to PP0, rank 2 is set to PP2, and so on.

K Register

The 12-bit K register is visible to the programmer through the maintenance channel only. This register holds the operation code field of an instruction for display on the IOU deadstart panel and for deadstart panel interrogation. When a PP is halted (idled), this register contains all ones.

PP Numbering

PPs are numbered in octal as follows:

Barrel	PPs
0	00 to 04
1	05 to 11
2	20 to 24
3	25 to 31

The deadstart sequence is used to determine PP numbering within a barrel. The sequence assigns barrel numbers according to the switch settings and, during the first minor cycle after deadstart, loads a zero into the Q register in barrel 0. This defines all the data in that rank of the barrel as belonging to PP0 and since Q is the channel selector, assigns PP0 to channel 0. During the next minor cycle, Q loads with a one. This defines PP1 and assigns it to channel 1. This process occurs in parallel in all barrels until the IOU assigns each rank of each barrel with a PP number and a channel number. Reassignment can be done only during a deadstart.

PP Memory

Each PP has an independent 4K word memory; each word contains 16 data bits with the upper four bits set to zero and one parity bit. PP0 executes the deadstart program from the deadstart panel during the deadstart operation. Therefore, PP memory 0 must be operational. A PP memory reconfiguration feature allows the user to restore IOU operation if the IOU detects a fault in the PP memory normally assigned to PP0.

To reconfigure, the operator assigns a good PP memory to PP0 and the operating system removes the failing PP memory. Computer operation can continue without the failing PP memory, and repairs can be made during scheduled maintenance. The system must be deadstarted to reconfigure PPMs.

I/O Channels

The I/O channels are comprised of an internal interface that allows common hardware and software to control the external devices, and an external interface that allows the IOU to communicate with the external devices using 12-bit data channels. The internal interface can transfer 16-bit data words between two PPs, or between a PP and an external device at a maximum rate of one word every 250 nanoseconds. This rate can be sustained for the maximum practical channel transfer (4096 words). During transfers between PPs, if the PPs are in the slot at the same time, the transfer rate is 500 nanoseconds.

Any PP can access any of the CYBER 170 bidirectional I/O channels. All PPs communicate with external devices through the independent I/O channels. Each channel may be connected to one or more pieces of external equipment, but only one piece of equipment can use a channel at one time. All channels can be active simultaneously.

The display station controller (DSC) is attached to CYBER 170 channel 10_8. The DSC is the IOU interface between the PPs and the system console, servicing both the keyboard and the cathode-ray tube (CRT). It transmits function words and digital symbol size/position data to the system console and receives digital character codes from the keyboard. It also receives digital symbol codes from the PPs and converts these to analog signals to the CRT.

Real-Time Clock

The real-time clock is a 12-bit free-running counter, incrementing at a 1-megahertz rate. It is permanently attached to channel 14_8. This channel may be read at any rate as its active and full flags are always set.

Two-Port Multiplexer

The two-port multiplexer provides communication capability between a PP and two attached terminals. One port is reserved for maintenance purposes and the other port is reserved for future use. The two-port multiplexer is permanently attached to channel 15_8.

Maintenance Channel

The maintenance channel is used for initialization of the CP and CM maintenance registers and monitoring of error status.

The maintenance channel consists of the maintenance channel interface on channel 17_8, a maintenance access control in each system element, and a set of interconnecting cables.

Central Memory Access

Any PP can access CM. During a write from the IOU to CM, the IOU assembles five successive 12-bit PP words into a 64-bit CM word with the leftmost four bits undefined. During a CM read, the IOU disassembles the rightmost 60 bits of the 64-bit CM word into five PP words. To find the CM address, a PP reads the A register. If bit 17 of the A register is clear, the PP uses the contents of the A register for the CM address. If bit 17 of the A register is set, the PP adds the relocation address from the R register of the A register to form the CM address.

A maximum of 20 PPs in various stages of assembly/disassembly can simultaneously read CM words, and five PPs can write CM words.

Model 835 Operating Instructions 3

Controls and Indicators 3-1
Deadstart Panel Controls/Indicators 3-1
Central Memory Controls 3-6

Power-On and Power-Off Procedures 3-9

Operating Procedures 3-9
Control Checks 3-9
Deadstart Sequences 3-10

IOU Reconfiguration 3-11

Model 835 Operating Instructions 3

This chapter describes mainframe controls and indicators and the operating procedures which are hardware-dependent. Software-dependent procedures are in system software reference manuals listed under Additional Related Manuals in About This Manual.

Controls and Indicators

This section describes IOU deadstart controls and indicators and CM configuration switches used by the system operator. Other controls used by maintenance personnel are described in the hardware operator's guide and the hardware maintenance manuals of the power distribution and warning system, the cooling system, and the system console listed in the system publication index in About This Manual.

Deadstart Panel Controls/Indicators

The deadstart panel (figure 3-1) is in the IOU. It contains PP register selection and display facilities, deadstart controls, error indicators, and a switch matrix, which is the source for a short PP program for initialization or troubleshooting. The switches, indicators, and their functions are listed in table 3-1.

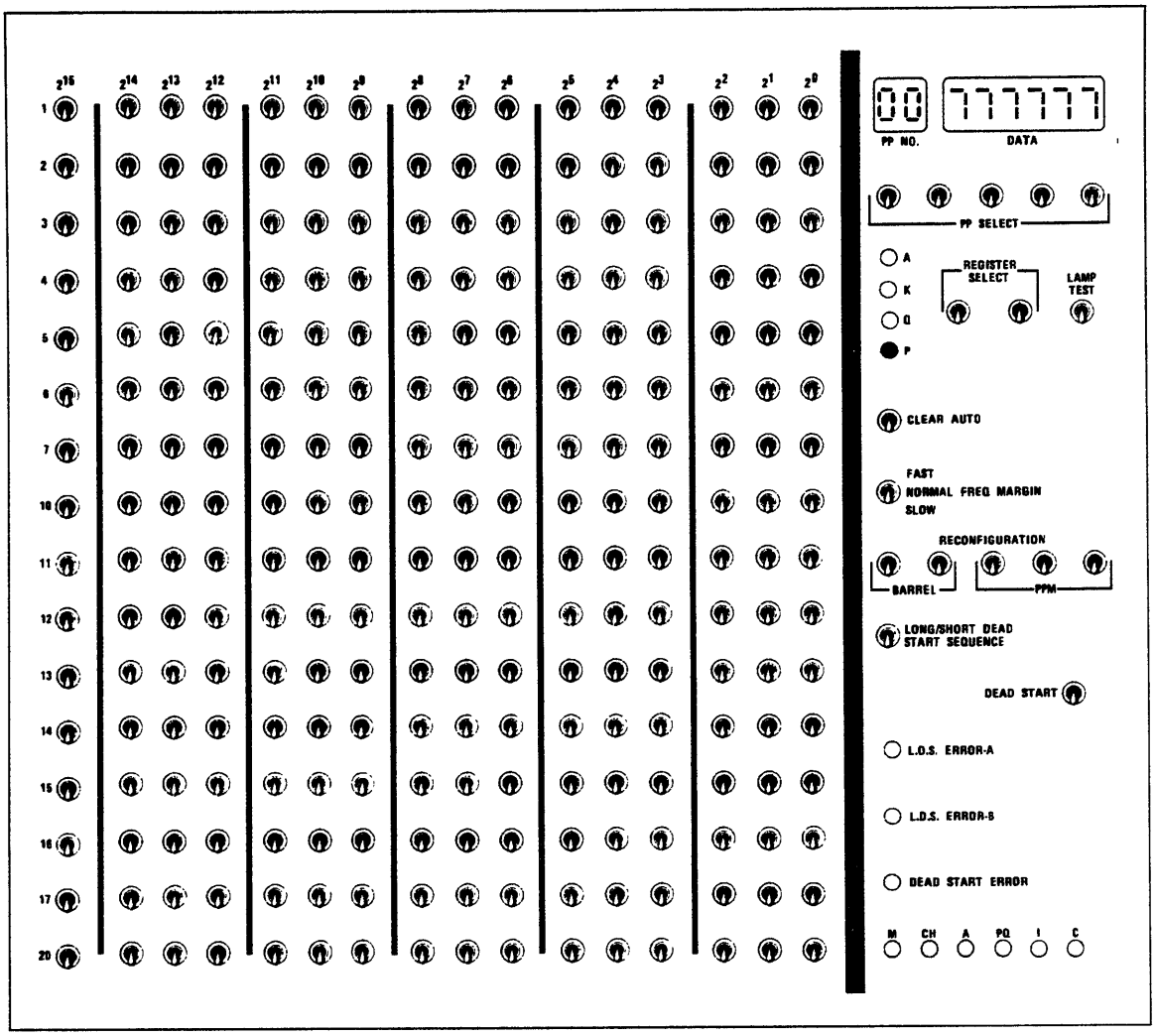


Figure 3-1. Deadstart Panel

Table 3-1. Deadstart Panel Controls/Indicators

Panel Nomenclature	Description	Function
20 through 215 by 1 through 208_8	Toggle switch matrix (two-position switches)	Provides a 16-word deadstart program for PP0. Switches 2^0 through 2^{11} set 12 bits for each of the program words, labeled 1 through 20 (octal). Switches 2^{12} through 2^{15} are set to zero. Up position sets bit. Down position clears bit.
PP NO	Octal display	Shows the PP selected by PP SELECT switches.
DATA	Octal display	Shows the content of the register selected by REGISTER SELECT switches.
PP SELECT	Toggle switches (two-position)	Selects the PP whose register is to be displayed.
REGISTER SELECT	Toggle switches (two-position)	Selects the register to be displayed (00 = P, 01 = Q, 10 = K, 11 = A).
A, K, Q, P	Indicators	One of these lights to indicate which register is selected by REGISTER SELECT switches.
LAMP TEST	Toggle switch (two-position)	Lights all indicators and display segments.
CLEAR AUTO	Toggle switch (two-position)	Allows manual clearing of auto-mode bit (bit 34 of the environment control register) to override possible auto-mode selection. This allows the selection of the PP and register from the deadstart panel if bit 34 is set.
FREQ MARGIN	Toggle switch (three-position)	Determines the frequency margin selected (FAST/NORMAL/SLOW). The setting of this switch is sensed only at deadstart time.

(Continued)

Table 3-1. Deadstart Panel Controls/Indicators *(Continued)*

Panel Nomenclature	Description	Function
RECONFIGURATION, BARREL	Toggle switches (two-position)	Selects the physical barrel which is logical barrel 0. All the other logical barrels are numbered from the selected physical barrel circularly. (If physical barrel 1 is selected by the switches, physical barrel 2 is logical barrel 1, and so on.)
RECONFIGURATION, PPM	Toggle switches (two-position)	Selects the physical PP memory which is logical PPM0. All the other PPMs in all barrels are numbered from the selected physical PPM circularly. If the switches are set to a value greater than four, no reconfiguration takes place.
LONG/SHORT DEAD START SEQUENCE	Toggle switch (two-position)	Selects the LONG/SHORT deadstart sequence. The setting of this switch is sensed only at deadstart.
DEAD START	Toggle switch (three-position, center is off)	Selects the fast or slow repetitive deadstart, which generates a master clear pulse every 250 or 4000 microseconds respectively. Up position selects fast deadstart; down position selects slow deadstart. (The single deadstart control pushbutton is on the system console.)
L.D.S. ERROR-A	Indicator	Remains lit when long deadstart branch tests are not completed within 10.25 microseconds.
L.D.S. ERROR-B	Indicator	Remains lit when a long deadstart sequence does not go to completion.
DEAD START ERROR	Indicator	Lights in case of long deadstart ROM address/data parity error.

(Continued)

Table 3-1. Deadstart Panel Controls/Indicators *(Continued)*

Panel Nomenclature	Description	Function	
M, CH, A, PQ, I, C	Indicators	Lights in case of hardware failures as follows:	
		M:	PP memory failure
		CH:	I/O channel failure
		A:	A barrel failure
		PQ:	P or Q barrel failure
		I:	Firmware or control failure
		C:	12/16 conversion failure

Central Memory Controls

The CM, without the Memory Upgrade Option, contains four three-position configuration switches (figure 3-2). These switches are located along the edge of a printed circuit board located just to the right of the center post in the middle section of the memory cabinet (location D01).

The switches are used to eliminate CM sections with malfunctions. For this model without the Memory Upgrade Option, each switch, SW3 through SW6, forces one corresponding CM address bit, 23 through 20, either to a zero (switch down) or to a one (switch up). Refer to table 3-2.

In case of CM malfunctions, the remaining good memory can be reconfigured so it is accessible by contiguous addresses from 0 to the maximum remaining address. This is accomplished by setting configuration switches (figure 3-2) as listed in table 3-2. Refer to the hardware operator's guide listed in the system publication index for further information.

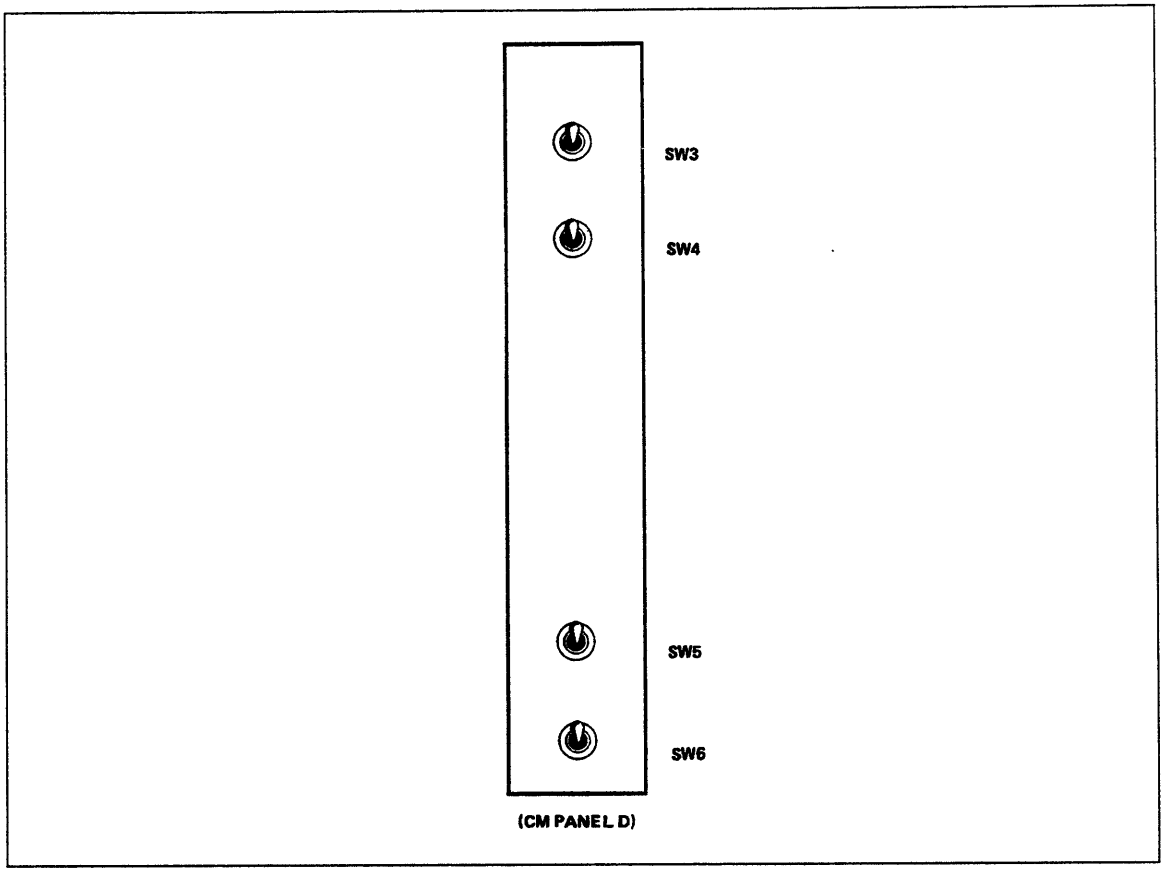


Figure 3-2. CM Configuration Switches

Table 3-2. Central Memory Reconfiguration

Original CM		Reconfigured CM				
Location of Failing CM[1]						
Words (Size)	**Address Range**	**Words (Size)**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**
524K (4 MB)	0-1 777 777	262K (2 MB)			0	X
		262K (2 MB)			1	X
1049K (8 MB)	0-3 777 777	524K (4 MB)		0	X	X
		524K (4 MB)		1	X	X
1573K (12 MB)	0-5 777 777	524K (4 MB)	0	0	X	X
		524K (4 MB)	0	1	X	X
		1049K (8 MB)	1	0	X	X
2097K (16 MB)	0-7 777 777	1049K (8 MB)	0	X	X	X
		1049K (8 MB)	1	X	X	X

1. CM remaining can be further reconfigured by setting additional configuration switches.

(Continued)

Table 3-2. Central Memory Reconfiguration *(Continued)*

Original CM		Reconfigured CM				
Reconfiguration Setting[2]						
Words (Size)	**Address Range**	**Words (Size)**	**SW3**	**SW4**	**SW5**	**SW6**
524K (4 MB)	0-1 777 777	262K (2 MB)	-	-	U	-
		262K (2 MB)	-	-	D	-
1049K (8 MB)	0-3 777 777	524K (4 MB)	-	U	-	-
		524K (4 MB)	-	D	-	-
1573K (12 MB)	0-5 777 777	524K (4 MB)	-	U	-	-
		524K (4 MB)	-	D	-	-
		1049K (8 MB)	D	-	-	-
2097K (16 MB)	0-7 777 777	1049K (8 MB)	U	-	-	-
		1049K (8 MB)	D	-	-	-

2. U equals up, D equals down, and dash (-) equals center position.

Power-On and Power-Off Procedures

In case of an emergency, use the system EMERGENCY OFF switch. The power-on and power-off procedures are described in the hardware operator's guide listed in the system publication index in About This Manual.

CAUTION

Improper application or removal of power may damage system circuits and/or air conditioning system. Power must be turned on/off by designated personnel only, except for the system EMERGENCY OFF switch. Use only for extreme emergency, not for normal shutdown.

Operating Procedures

Refer to the hardware operator's guide listed in the system publication index in About This Manual. The system is initialized by setting its control switches, and then by running either a long or short deadstart sequence (defined later in this section). After initialization, the keyboard is used to instruct the system further, under program control.

Control Checks

Before activating a long or short deadstart sequence, check the positions of deadstart panel switches against their intended use. These checks can be made by using table 3-1. The normal settings of these switches is as follows:

Switch	Position
CLEAR AUTO	Down
FREQ MARGIN	Center
RECONFIGURATION	All down
LONG/SHORT DEAD START SEQUENCE	Down for a short deadstart sequence
DEAD START	Center
All error lights	Not lit

Deadstart Sequences

In response to a deadstart signal from either the deadstart pushbutton on the system console, or from the DEAD START switch on the deadstart panel, circuits in the IOU perform a deadstart sequence. Depending on the setting of the LONG/SHORT DEAD START SEQUENCE switch on the deadstart panel, either the long or the short deadstart sequence is performed. The short deadstart sequence is used when hardware integrity verification is not required. The long deadstart sequence performs all the tasks performed by the short deadstart sequence and some additional tasks. The main additional task is the running of a diagnostic program, from a read-only memory (ROM) in the IOU, on logical PP0. The diagnostic program takes approximately 1 minute to run.

Both deadstart sequences begin with a master clear which sets up all PPs, except logical PP0, for a 4096-word block input starting at PP location 0. The input into each PP is from the channel with the same number as the logical number of the PP concerned. The master clear also resets all external devices and sets maintenance channel connect code bit 52.

The individual channels and registers are set as follows:

Channel	Active/ Inactive Flag	Full/Empty Flag	Channel Flag	Channel Error Flag
0	Inactive	Empty	Clear	Clear
10 (display controller)	Active	Empty	Clear	Clear
14 (real-time clock)	Active	Full	Set	Set
15 (two-port mux)	Active	Empty	Clear	Clear
17 (maintenance)	Active	Empty	Clear	Clear
Other installed channels	Active	Empty	Clear	Clear
Noninstalled channels	Inactive	Empty	Clear	Clear

The flags of channel 14 and of noninstalled channels are fixed by hardware and cannot be changed.

Register	Initialization[1]	Description
K	007100_8	Instruction display on deadstart panel
P	007777_8	Causes block input to start from location 0
A	$10{,}000_8$	Count of 4096 words
Q	0, 1, 2...	I/O channel numbers (PP0: 0, PP1: 1, and so on)

All registers in all barrels are set to these values, except the registers of PP0.

1. Leading zeros are not displayed on deadstart panel.

If the long deadstart sequence is being performed, hardware clears location 7777_8 in all PP memories and sets the P register of PP0 to 6000_8. PP0 starts performing a test program from a read-only memory in IOU and lights the deadstart panel L.D.S. ERROR-A and L.D.S. ERROR-B indicators. Indicator A remains lit unless the test program reaches location 6200_8 within 10.25 microseconds. Indicator B remains lit until the test program reaches location 7776_8. When this happens, the unique part of the long deadstart sequence ends with a master clear.

Next, both deadstart sequences clear PP0 location 0, write the settings of the deadstart panel matrix switches into PP0 memory locations 1_8 to 20_8, and clear PP0 location 21_8. PP0 then starts executing the program entered from the matrix switches, which is normally a bootstrap program to input more data from an assigned external device.

The short deadstart sequence does not disturb PP memory other than PP0 locations 0 to 21_8. Both deadstart sequences leave all PPs, except PP0, waiting for a block input or for action through the maintenance channel. After the block input is complete, each PP starts executing the program entered from whatever address was entered into location 0 of that PP.

IOU Reconfiguration

The logical PP numbers and hardware are assigned to physical PPs circularly from the settings of IOU deadstart panel RECONFIGURATION switches, which specify which physical barrel and PPM is PP0. If the PPM section of these switches is set to a value greater than 4, the value 0 is substituted. If the BARREL section of these switches is set to a value greater than the number of installed barrels, the value 0 is substituted. Thus, possible barrel numbering is as described in table 3-3.

NOTE

The minimum system option is 10 PPs.

Table 3-3. Barrel Numbering Table

Barrels Installed	Physical Barrel	Logical PPs in Physical Barrel with BARREL RECONFIGURATION Switch Values			
		0	1	2	3
4 Barrels (20 PPs)	0	0-4	25-31	20-24	5-11
	1	5-11	0-4	25-31	20-24
	2	20-24	5-11	0-4	25-31
	3	25-31	20-24	5-11	0-4
3 Barrels (15 PPs)	0	0-4	20-24	5-11	(0-4)
	1	5-11	0-4	20-24	(5-11)
	2	20-24	5-11	0-4	(20-24)
2 Barrels (10 PPs)	0	0-4	5-11	(0-4)	(0-4)
	1	5-11	0-4	(5-11)	(5-11)
1 Barrel (5 PPs)	0	0-4	(0-4)	(0-4)	(0-4)

Models 845 and 855 System Description 4

Physical Characteristics 4-1

Functional Characteristics 4-4
Central Processor 4-5
Central Memory 4-6
Input/Output Unit 4-7

Major System Component Descriptions 4-8
Central Processor 4-8
Instruction Section 4-8
Registers 4-8
Execution Section 4-9
Cache Memory 4-9
Addressing Section 4-9
Central Memory Control 4-9
Central Memory 4-10
Input/Output Unit 4-11
System Console 4-12

Models 845 and 855 System Description 4

This chapter describes the physical and functional characteristics and major system components.

These high-speed computer systems without the Memory Upgrade Option are used for both business and scientific applications. The system includes the following components.

- Central processor (CP)
- Central memory (CM)
- Input/output unit (IOU)

Physical Characteristics

The system configuration without the Memory Upgrade Option (figure 4-1) include a three-section cabinet for the CP, CM, and IOU. (The system console is also required for system operation.)

Each cabinet section contains a logic chassis with plug-in circuit boards. The logic chassis in the IOU also contains a deadstart panel with initialization and maintenance controls and displays. Each cabinet section also contains a self-contained cooling unit to cool the logic chassis, an ac/dc control section with voltage margin testing facilities, and dc power supplies. For additional cooling or power information, refer to the cooling system and power system manuals listed in the system publication index.

The system configurations shown in figures 4-2, 4-3, and 4-4 each contain an interconnected three-section cabinet for the CP, CM, and IOU. (The system console is also required for system operation.) The model 855 with the Memory Upgrade Option supports an optional second CP, which is contained in an additional one-bay section. (The model 855 second CP does not support CYBER 170 State operation.)

Each cabinet section contains a logic chassis with plug-in circuit boards. The CP cabinet section comprises three attached subsections, each with separate power and cooling facilities. A cooling unit provides cooling for the CP subsections and CM. The IOU cabinet section has a self-contained cooling unit to cool the IOU logic chassis. Each cabinet section also contains an ac/dc control section with voltage margin testing facilities and dc power supplies. For additional cooling or power information, refer to the cooling system and power system manuals listed in the system publication index.

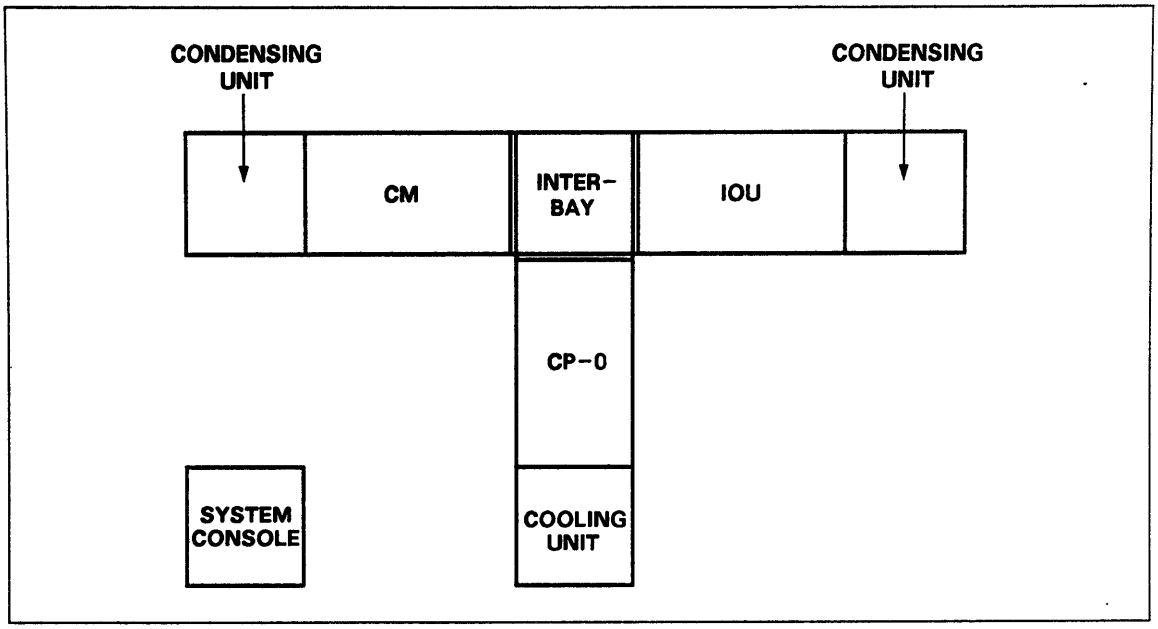


Figure 4-1. System Configuration Without Memory Upgrade Option

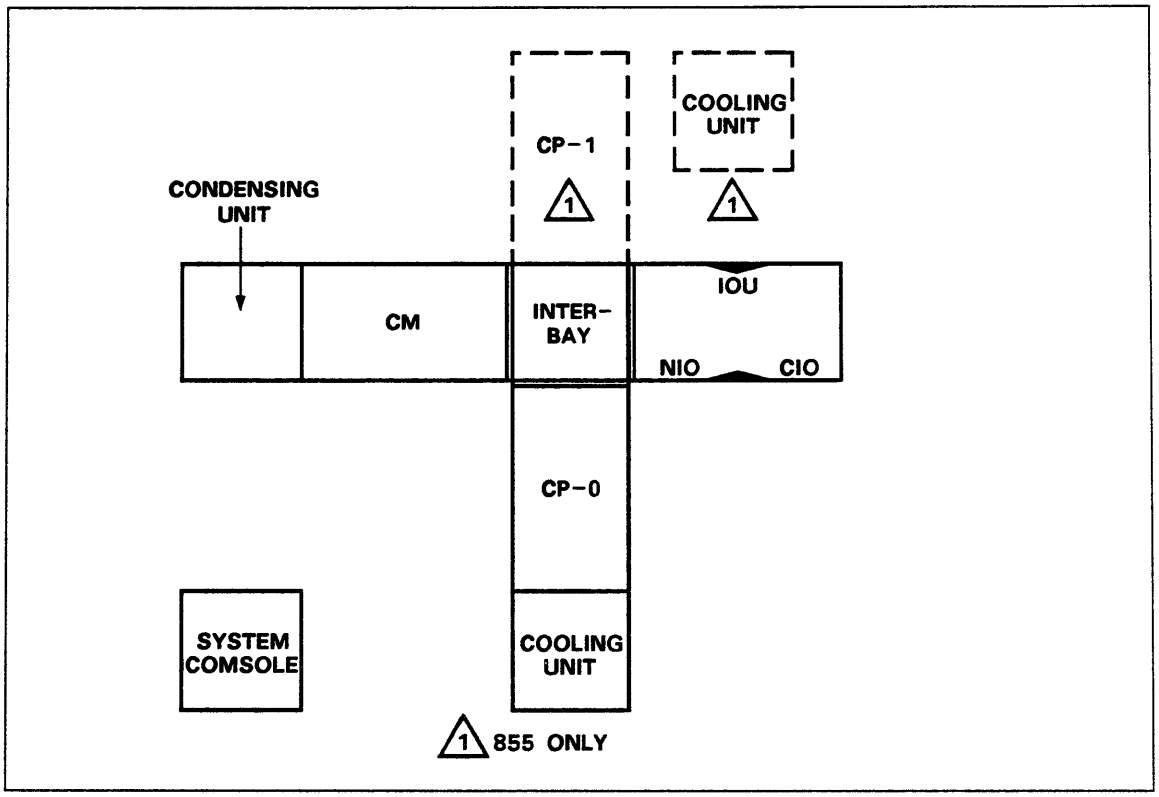


Figure 4-2. System Configuration Without Memory Upgrade Option and With IOU-NIO/CIO Upgrade Option

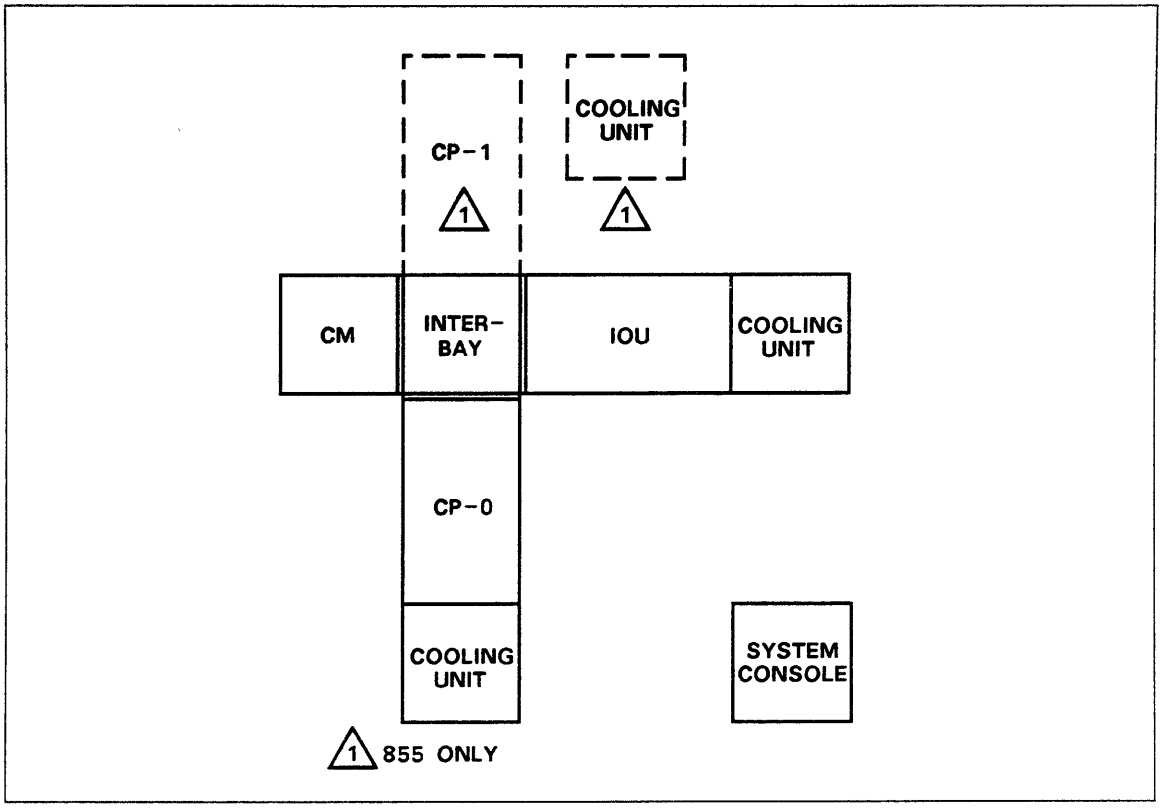


Figure 4-3. System Configuration With Memory/Optional CP Upgrade Options

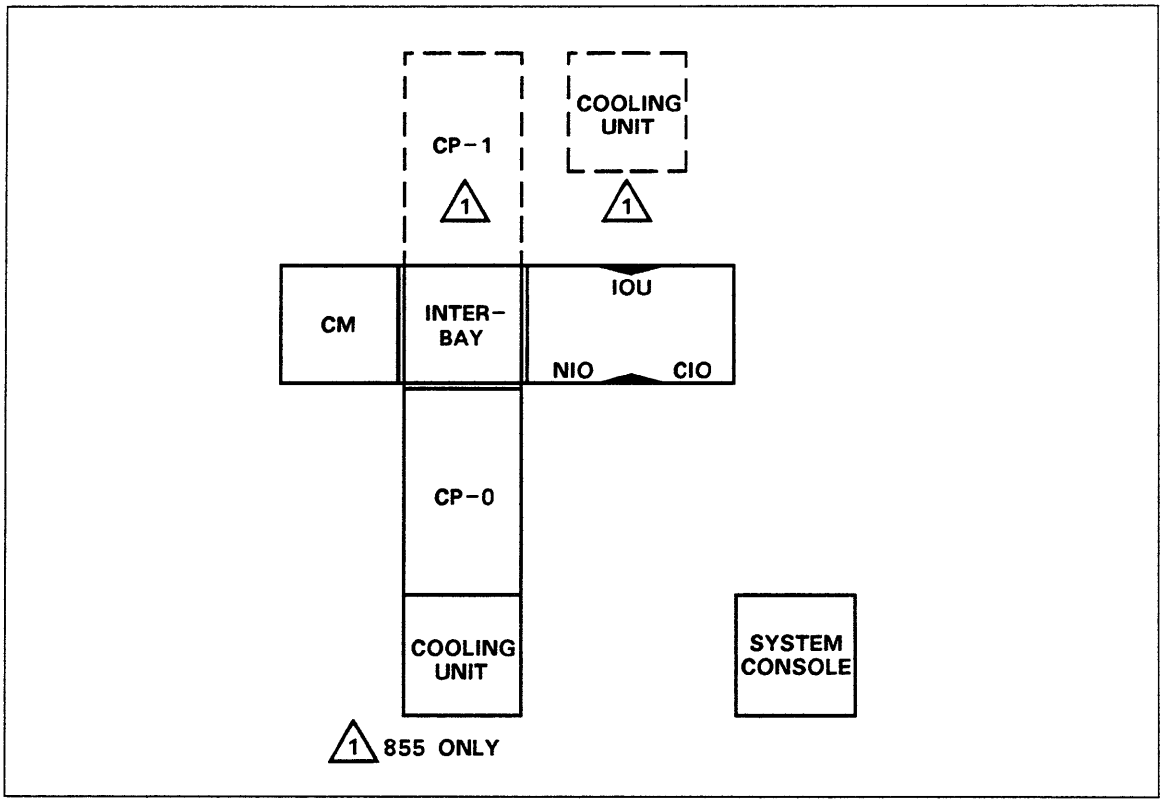


Figure 4-4. System Configuration With Memory, Optional CP, and IOU-NIO/CIO Upgrade Options

Functional Characteristics

To achieve high computation speeds, these models use ECL and large-scale integration (LSI) logic. High speed is also the objective of the CP design, which is based on the assumption that both data and instructions are, in most cases, accessed from successive memory locations. Accordingly, the CP prefetches both instructions and data expected to be used next while the current instruction is being processed.

The CP supports two states of operation.

Virtual State	Operates with virtual-memory byte addressing, using the Virtual State instruction set and data formats. Virtual State is the native operating state of the CP.
CYBER 170 State	Operates with real-memory word addressing, using the CYBER 170 State CP instruction set and data formats.

The Virtual State and CYBER 170 State environments may be present at the same time with the CP executing in either environment.

- NOS/VE is the operating system of Virtual State.
- NOS is the operating system of CYBER 170 State.

The semiconductor central memory is divided into eight independent banks. These banks may all be simultaneously in the process of completing read/write requests which are queued and distributed at ECL speeds. System input/output speeds are determined by the capabilities of existing external devices.

Central Processor

The CP has the following characteristics:

- 60-bit internal word.
- Eight 60-bit operand (X) registers.
- Eight 18-bit address (A) registers.
- Eight 18-bit index (B) registers.
- Two registers that isolate each user's central memory space (RAC, FLC).
- Two registers that isolate each user's extended memory space (RAE, FLE).
- Register exchange instructions (exchange jumps) for interrupting programs.
- Floating-point (FP) arithmetic (10-bit exponent plus sign bit, 48-bit coefficient plus sign bit). Some FP instructions use 96-bit (double-precision) coefficients.
- Integer arithmetic (60/18-bit operands).
- Character string compare/move facilities (6-bit characters).
- Packed instructions (15/30/60-bit instructions in 60-bit words).
- Synchronous internal logic.
- 64-nanosecond clock period.
- 2048-word cache buffer memory, option available for 4096-word cache.
- Instruction and branch instruction lookahead.
- Microcode control.
- Parity checking of all major data and address paths.
- Maintenance channel to IOU.

Central Memory

The CM without the Memory Upgrade Option has the following characteristics:

- 72-bit data word (60 data bits, 8 single-error correction/double-error detection bits, and 4 unused bits).
- 524K words of refresh-type semiconductor memory, options available to 4194K words.
- Organization of eight independent banks.
- Two memory ports (located in the central processor cabinet).
- Bounds register to limit write access.
- 64-nanosecond clock period.
- Maximum data transfer rate of one word every 64 nanoseconds.
- 528-nanosecond read access time.
- 448-nanosecond read/write cycle time.
- 896-nanosecond partial write cycle time.
- Read and write data queuing capability.
- Single-error correction/double-error detection (SECDED) on stored data.
- Parity checking of all major data, address and control paths.
- Unified-extended memory (UEM) which serves as extended memory within CM.

The CM with the Memory Upgrade Option has the following characteristics (timing references are from CMC/CPU interface):

- 72-bit data word (60 data bits, 8 single-error correction/double-error detection bits, and 4 unused bits).
- 2097K words (16 megabytes) of dynamic random-access memory, options available to 16 776K words (128 megabytes).
- Organization of eight independent banks.
- Two memory ports (located in the central processor cabinet).
- Bounds register to limit write access.
- 64-nanosecond clock period.
- Maximum data transfer rate of one word every 32 nanoseconds.
- 464-nanosecond read access time.
- 384-nanosecond read/write cycle time.
- 768-nanosecond partial write cycle time.
- Read and write data queuing capability.

- Single-error correction/double-error detection (SECDED) on stored data.
- Parity checking of all major data, address, and control paths.
- Unified-extended memory (UEM) which serves as extended memory within CM.

Input/Output Unit

The IOU has the following characteristics:

- Ten peripheral processors (PPs), 15-PP/20-PP options available. Each PP has 4K independent memory (PPM) comprised of 16-bit words with the upper 4 bits zero.
- Port to central memory.
- Bounds register to limit writes to central memory.
- Twelve 12-bit CYBER 170 channels to external devices, 24 channel option available.
- Real-time clock (channel 14_8).
- Display controller (CYBER 170 channel 10_8).
- Two-port multiplexer (channel 15_8).
- Maintenance channel (channel 17_8).
- Parity checking on all major data and address paths.
- Operating speed of 250 nanoseconds and a minor cycle of 50 nanoseconds.
- Optional concurrent input/output (CIO) PPs and direct-memory access (DMA) I/O channel adapter. Available only on systems with IOU-NIO/CIO cabinet upgrades.

Major System Component Descriptions

Central Processor

The CP hardware (figures 4-5) consists of the following:

- Instruction section
- Registers
- Execution section
- Cache memory
- Addressing section
- Central memory control

The CP is isolated from the IOU and is thus able to carry on computation or character manipulation unencumbered by I/O requirements.

Instruction Section

The instruction section directs the arithmetic and manipulative functions for instruction execution. The instruction section prefetches instruction words from memory and disassembles them into instructions.

Registers

Operating registers reduce storage accesses for operands used during the execution of an instruction. These registers are:

- Eight 60-bit X registers (X0 through X7) which hold operands used for computation.
- Eight 18-bit A registers (A0 through A7) which use A0 primarily for indexing and A1 through A7 for CM operand addressing.
- Eight 18-bit B registers (B0 through B7) which are primarily indexing registers to control program execution. The B0 register always contains all zeros.

Eight support registers support the operating registers during program execution. These registers are:

- 18-bit program address (P) register.
- 21-bit reference address for CM (RAC) register. This is a program's lower bound.
- 21-bit field length for CM (FLC) register. This is a program's upper bound.
- 6-bit exit mode (EM) register.

- 6-bit flag register.
- 21-bit reference address for UEM (RAE) register.
- 24-bit field length for UEM (FLE) register.
- 18-bit monitor address (MA) register.

The registers store data and control information, present operands to the execution section, and store results.

The operating and support registers reside in the operand issue section.

Execution Section

The execution section combines the operands to achieve the result.

Cache Memory

The cache memory consists of two sets of fast bipolar memory, capable of storing 2048 60-bit words. It can be expanded to four sets with a capacity of 4096 words. The memory addressing sections determine whether a requested word is in the cache memory. If it is not, they read four consecutive words from central memory into the cache memory.

Addressing Section

The addressing section checks memory addresses against the CP registers RAC, FLC, RAE, and FLE to ensure isolation of user memory space.

Central Memory Control

Central memory control (CMC) is integrated within the CP and controls the flow of data between CM and requesting system components.

Central Memory

The CM (figure 4-5) consists of the following:

- Eight memory banks
- Memory ports

The CM without the Memory Upgrade Option is a refresh-type metal-oxide semiconductor (MOS) memory organized into eight independent banks.

The CM with the Memory Upgrade Option is a dynamic random-access organized into eight independent banks.

A portion of CM can be reserved for use as extended memory. It is called unified-extended memory (UEM) and is referenced by the RAE and FLE registers. The memory ports are located in the central processor cabinet, with one port having a queuing buffer.

Input/Output Unit

The IOU (figure 4-5) consists of the following:

- Ten logically independent peripheral processors (PPs). Options are available to increase total to 15 or 20 PPs.
- Internal interface to 12 I/O channels. 24-channel option is available.
- External interfaces to I/O channels
  - 11 or 23 CYBER 170 channel interfaces.
  - Display controller interface (CYBER 170 channel 10_8).
  - Real-time clock interface (channel 14_8).
  - Two-port multiplexer interface (channel 15_8).
  - Maintenance channel interface (channel 17_8).
- Interface to central memory.
- Bounds register to limit writes to CM.
- Optional CIO PPs and DMA I/O channel adapter. Available only on systems with IOU-NIO/CIO cabinet upgrades.

The PPs are organized in groups of five, called barrels. The PPs in a barrel time-share common hardware. Each PP has its own independent memory and communicates with all I/O channels and central memory.

System Console

The system console, required for system operation, provides a visual, alphanumeric readout for the computer. The receipt of symbol and position information from the computer enables displaying program information on a cathode-ray tube (CRT). The station also contains an alphanumeric keyboard which enables an operator to send data to the computer. The keyboard and CRT combination permits the computer operator to monitor and control system operation. Except for programming information in chapter 17, refer to the system console manual listed in the system publication index in About This Manual for further information.

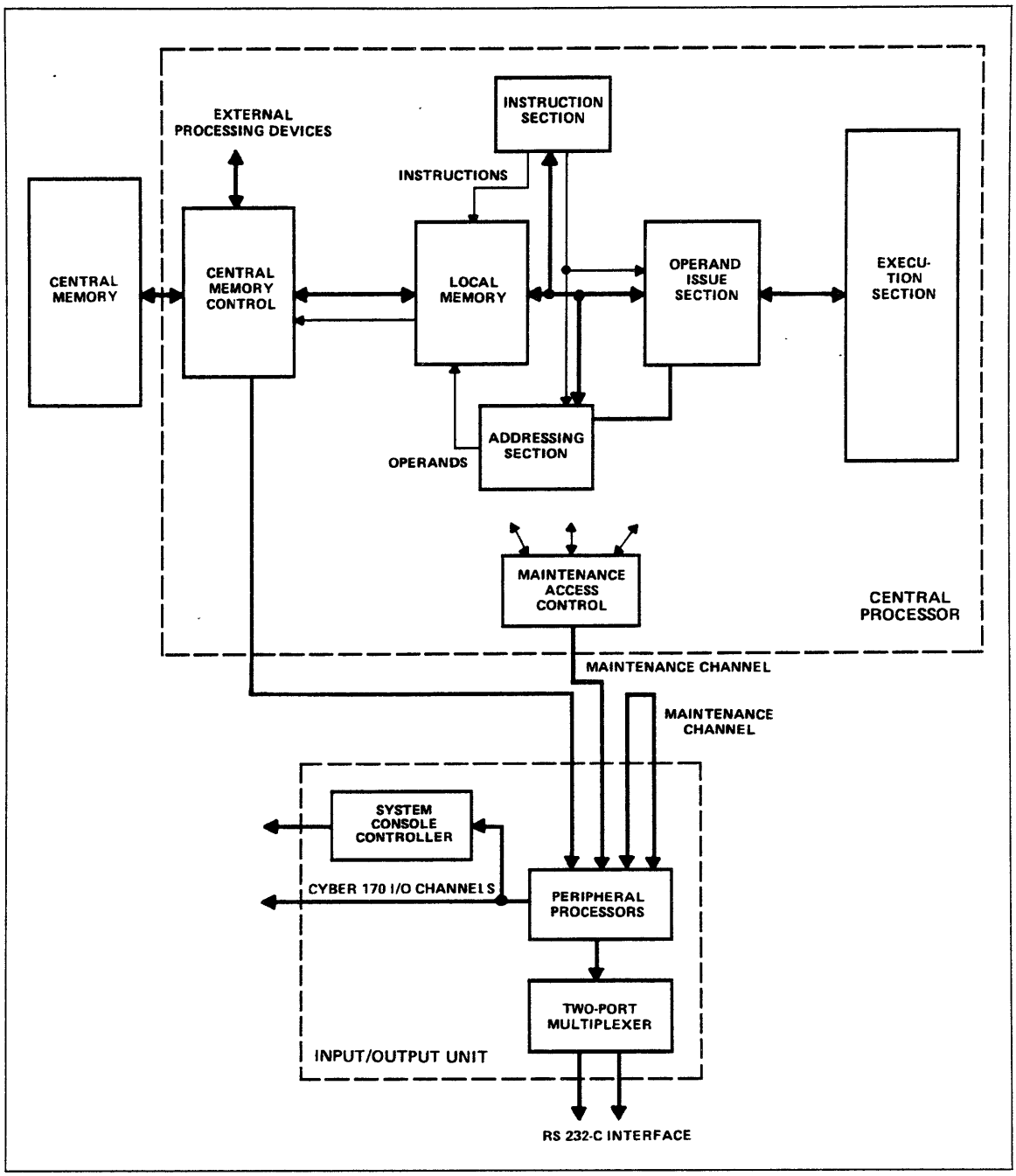


Figure 4-5. Computer System Block Diagram

Models 845 and 855 Functional Descriptions 5

Central Processor 5-1
Instruction Section 5-1
Model 845 Instruction Lookahead 5-1
Model 855 Instruction Lookahead 5-1
Maintenance Access Control 5-1
Instruction Control Sequences 5-2
Boolean Sequence 5-2
Shift Sequence 5-3
Floating-Add Sequence 5-4
Floating-Multiply and Floating-Divide Sequence 5-4
Increment Sequence 5-4
Compare/Move Sequence 5-6
CYBER 170 Exchange Sequence 5-7
Block Copy Sequence 5-7
Direct Read/Write Sequence 5-7
Normal Jump Sequence 5-8
Return Jump Sequence 5-8
Registers 5-9
Operating Registers 5-10
X Registers 5-10
A Registers 5-10
B Registers 5-10
Support Registers 5-11
P Register 5-11
RAC Register 5-11
FLC Register 5-11
EM Register 5-12
Flag Register 5-12
RAE Register 5-13
FLE Register 5-13
MA Register 5-13
Execution Section 5-14
Cache Memory 5-14
Addressing Section 5-14
Central Memory Control 5-14

Central Memory 5-15
Address Format Without Memory Upgrade Option 5-15
Address Format With Memory Upgrade Option 5-16
CM Access and Cycle Times 5-16
CM Access and Cycle Times Without Memory Upgrade Option 5-17
CM Access and Cycle Times With Memory Upgrade Option 5-17
CM Ports and Priorities 5-17
SECDED 5-19
CM Layout 5-24
CM Bounds Register 5-24
Central Memory Reconfiguration 5-24

Input/Output Unit 5-25
Peripheral Processor 5-25
Deadstart 5-26
Barrel and Slot 5-26

PP Registers 5-28
R Register 5-28
A Register 5-28
P Register 5-29
Q Register 5-29
K Register 5-29
PP Numbering 5-30
PP Memory 5-30
I/O Channels 5-31
Real-Time Clock 5-32
Two-Port Multiplexer 5-32
Maintenance Channel 5-32
Central Memory Access 5-32

Models 845 and 855 Functional Descriptions 5

This chapter provides functional descriptions of the central processor (CP), central memory (CM), and input/output unit (IOU) as shown in the system block diagram in chapter 4. Functional descriptions for the system display station and the cooling system are in their respective manuals listed in the system publication index in About This Manual.

Central Processor

The CP consists of the instruction section, registers, the execution section, cache memory, central memory control, and the addressing section.

Instruction Section

The instruction section consists of logic for instruction control.

Model 845 Instruction Lookahead

The model 845 instruction lookahead hardware (ILH) prefetches a maximum of 12 instructions to make the next instruction immediately available when the execution of the previous instruction is complete. This is accomplished by reading instructions from cache/CM into a series of buffer ranks.

The model 845 responds to both negative and positive resolution of a conditional branch by purging the buffer ranks and reinitializing the Instruction Fetch Unit.

Model 855 Instruction Lookahead

The model 855 instruction lookahead hardware (ILH) prefetches a maximum of 12 instructions to make the next instruction immediately available when the execution of the previous instruction is complete. This is accomplished by reading instructions from cache/CM into a series of buffer ranks.

When ILH detects a conditional branch, it assumes that the branch condition will be met. ILH computes the branch target address and reads instructions from cache/CM starting at the target address. If the branch is taken, the buffer ranks contain the next executable instruction words. If the branch is not taken, the hardware purges the buffer ranks and resumes prefetching at the instruction word following the unsatisfied branch instruction.

Maintenance Access Control

The maintenance access control performs initialization and maintenance operations in the CP.

Instruction Control Sequences

The instruction control section performs instruction translation and control sequences. Each control sequence obtains the necessary instruction operands from the operating registers and provides the control signals for execution. Instructions read from CM are 60-bit instruction words that are in four 15-bit groups, two 30-bit groups, or a combination of 15-bit and 30-bit groups. The 15-bit groups are termed parcels with the first parcel (parcel 0) being the highest-order 15 bits of a 60-bit CM word. Second, third, and fourth parcels (parcels 1, 2, and 3) follow in order. The 30-bit groups contain two 15-bit parcels.

The instruction control sequences control the execution of one or more instructions of a common type. These sequences and associated instructions are briefly described in this section. For further information, refer to CP Instruction Descriptions in chapter 16.

Boolean Sequence

The Boolean sequence controls instructions that require bit-by-bit data manipulation. This includes both the logical and transmissive operations. The instructions requiring logical operations are:

11	Logical product (Xj) and (Xk) to Xi	BXi Xj * Xk
12	Logical sum of (Xj) and (Xk) to Xi	BXi Xj + Xk
13	Logical difference of (Xj) and (Xk) to Xi	BXi Xj - Xk
15	Logical product of (Xj) with complement of (Xk) to Xi	BXi -Xk * Xj
	Logical sum of (Xj) with complement of (Xk) to Xi	BXi -Xk + Xj
17	Logical difference of (Xj) with complement of (Xk) to Xi	BXi -Xk - Xj

The instructions requiring transmissive operations are:

10	Transmit (Xj) to Xi	BXi Xj
11	Transmit complement of (Xk) to Xi	BXi -Xk

Shift Sequence

The shift sequence controls instructions that require shifting the 60-bit field of data within the operand word. The shift instructions are:

20	Left shift (Xi) by jk	LXi jk
21	Right shift (Xi) by jk	AXi jk
22	Left shift (Xk) nominally (Bj) places to Xi	LXi Bj, Xk
23	Right shift (Xk) nominally (Bj) places to Xi	AXi Bj, Xk
43	Form mask of jk bits to Xi	MXi jk

The shift sequence also controls the pack and unpack instructions. In the packed floating format, the coefficient is contained in the lower 48 bits. The sign and biased exponents are contained in the upper 12 bits. The unpack instruction obtains the packed word from the Xk register, delivers the coefficient to the Xi register, and delivers the exponent to the Bj register. The unpack and pack instructions are:

26	Unpack (Xk) to Xi and Bj	UXi Bj, Xk
27	Pack (Xk) and (Bj) to Xi	PXi Bj, Xk

The shift sequence also controls the normalize operations. The coefficient portion of the operand is repositioned, and the exponent is adjusted so that the most significant bit of the coefficient is in the highest-order bit position of the coefficient, and the exponent is decreased by the number of bit positions shifted. The normalize instructions are:

24	Normalize (Xk) to Xi and Bj	NXi Bj, Xk
25	Round normalize (Xk) to Xi and Bj	ZXi Bj, Xk

Floating-Add Sequence

The floating-add sequence controls the operations necessary to form the 48-bit floating sum with a 12-bit exponent of the floating-point sum or difference of two floating-point operands. The floating-add instructions are:

30	Floating sum of (Xj) and (Xk) to Xi	FXi Xj + Xk
31	Floating difference of (Xj) and (Xk) to Xi	FXi Xj - Xk
32	Floating double-precision sum of (Xj) and (Xk) to Xi	DXi Xj + Xk
33	Floating double-precision difference of (Xj) and (Xk) to Xi	DXi Xj - Xk
34	Round floating sum of (Xj) and (Xk) to Xi	RXi Xj + Xk
35	Round floating difference of (Xj) and (Xk) to Xi	RXi Xj - Xk

Floating-Multiply and Floating-Divide Sequence

The floating-multiply and floating-divide sequence controls the operation of floating-multiply, floating-divide, and population-count instructions.

The multiply instructions are:

40	Floating product of (Xj) and (Xk) to Xi	FXi Xj * Xk
41	Round floating product of (Xj) and (Xk) to Xi	RXi Xj * Xk
42	Floating double-precision product of (Xj) and (Xk) to Xi	DXi Xj * Xk

The divide instructions are:

44	Floating divide (Xj) by (Xk) to Xi	FXi Xj/Xk
45	Round floating divide (Xj) by (Xk) to Xi	RXi Xj/Xk

The population-count instruction counts the number of one bits in a 60-bit operand. The instruction is:

47	Population count of (Xk) to Xi	CXi Xk

Increment Sequence

The increment sequence controls the ones complement addition and subtraction of 18-bit fixed-point operands for increment instructions 50 through 77. The sequence also controls the 60-bit ones complement sum and difference values for long-add instructions 36 and 37.

The increment instructions are:

50	Set Ai to (Aj) + K	SAi Aj + K
51	Set Ai to (Bj) + K	SAi Bj + K
52	Set Ai to (Xj) + K	SAi Xj + K
53	Set Ai to (Xj) + (Bk)	SAi Xj + Bk
54	Set Ai to (Aj) + (Bk)	SAi Aj + Bk
55	Set Ai to (Aj) - (Bk)	SAi Aj - Bk
56	Set Ai to (Bj) + (Bk)	SAi Bj + Bk
57	Set Ai to (Bj) - (Bk)	SAi Bj - Bk
60	Set Bi to (Aj) + K	SBi Aj + K
61	Set Bi to (Bj) + K	SBi Bj + K
62	Set Bi to (Xj) + K	SBi Xj + K
63	Set Bi to (Xj) + (Bk)	SBi Xj + Bk
64	Set Bi to (Aj) + (Bk)	SBi Aj + Bk
65	Set Bi to (Aj) - (Bk)	SBi Aj - Bk
66	Set Bi to (Bj) + (Bk)	SBi Bj + Bk
67	Set Bi to (Bj) - (Bk)	SBi Bj - Bk
70	Set Xi to (Aj) + K	SXi Aj + K
71	Set Xi to (Bj) + K	SXi Bj + K
72	Set Xi to (Xj) + K	SXi Xj + K
73	Set Xi to (Xj) + (Bk)	SXi Xj + Bk
74	Set Xi to (Aj) + (Bk)	SXi Aj + Bk
75	Set Xi to (Aj) - (Bk)	SXi Aj - Bk
76	Set Xi to (Bj) + (Bk)	SXi Bj + Bk
77	Set Xi to (Bj) - (Bk)	SXi Bj - Bk

The long-add instructions are:

36	Integer sum of (Xj) and (Xk) to Xi	IXi Xj + Xk
37	Integer difference of (Xj) and (Xk) to Xi	IXi Xj - Xk

Compare/Move Sequence

The compare/move sequence controls data manipulation on a character basis. The compare/move instructions (also referred to as CMU instructions) are 60-bit instructions that use six support registers for source and result field CM addresses and character position offsets. The support registers load from the 60-bit instruction word. The compare/move instructions are:

464	Move indirect (Bj) + K	IM Bj + K
465	Move direct	DM
466	Compare collated	CC
467	Compare uncollated	CU

The support registers are:

- An 18-bit K1 register that specifies which relative CM address word contains the first character of the source data field.
- An 18-bit K2 register that specifies which relative CM address word contains the first character of the result field.
- A 4-bit C1 register that specifies the character position or offset of the first CM word of the source field.
- A 4-bit C2 register that specifies the character position or offset of the first CM word of the result field.
- Two 16-bit L registers (LA and LC) that specify the number of characters in the data field. The LA register is associated with K1, and the LC register is associated with K2. Instruction 464 uses 14 register bits. Instructions 465, 466, and 467 use only the lower eight register bits.

NOTE

CMU instructions are provided for compatibility with previous systems. For better performance, recompile jobs to avoid use of CMU instructions.

CYBER 170 Exchange Sequence

A CYBER 170 exchange sequence is the method used to swap jobs in and out of execution. When a CYBER 170 exchange jump instruction occurs, the CYBER 170 exchange sequence writes the contents of the current job's CP registers (described later in this chapter) into an area of central memory called a CYBER 170 exchange package. A CYBER 170 exchange package is associated with each job. It contains sufficient information to restart a job if the job is interrupted during execution and swapped out by a CYBER 170 exchange jump. To complete the sequence, CP registers for another job are read from its CYBER 170 exchange package and that job begins or resumes execution. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

Block Copy Sequence

The block copy sequence controls the transfer of data between CM and UEM. The number of words to be transferred is determined by the addition of K to the contents of Bj. The starting address for CM is formed by adding either the A0 register or certain bits of the X0 register to the RAC reference address. The starting address for UEM is formed by adding certain bits of the X0 register to the RAE reference address. The block copy instructions are:

011	Block copy Bj + K words from UEM to CM	RE Bj + K
012	Block copy Bj + K words from CM to UEM	WE Bj + K

Direct Read/Write Sequence

Instructions 014 and 015 perform single-word, direct read and write operations for UEM; and instructions 660 and 670 perform single-word, direct read and write operations for central memory.

014	Read one word from UEM at (Xk + RAE) into Xj	RXj Xk
015	Write one word from Xj to UEM at (Xk + RAE)	WXj Xk
660	Read central memory at (Xk) to Xj	CRXj Xk
670	Write Xj into central memory at (Xk)	CWXj Xk

Normal Jump Sequence

The normal jump sequence controls the execution of branch instructions 02 through 07. The 02 instruction performs an unconditional jump to the Bi register address plus K. The branch address is K with i equals zero. The 02 instruction is:

02	Jump to (Bi) + K	JP Bi + K

The conditional jump instructions 03 through 07 branch to address K if the jump condition is met.

These instructions are:

030	Branch to K if (Xj) = 0	ZR Xj, K
031	Branch to K if (Xj) = 0	NZ Xj, K
032	Branch to K if (Xj) is positive	PL Xj, K
033	Branch to K if (Xj) is negative	NG Xj, K
034	Branch to K if (Xj) is in range	IR Xj, K
035	Branch to K if (Xj) is out of range	OR Xj, K
036	Branch to K if (Xj) is definite	DF Xj, K
037	Branch to K if (Xj) is indefinite	ID Xj, K
04	Branch to K if (Bi) = (Bj)	EQ Bi, Bj, K
05	Branch to K if (Bi) ≠ (Bj)	NE Bi, Bj, K
06	Branch to K if (Bi) ≥ (Bj)	GE Bi, Bj, K
07	Branch to K if (Bi) < (Bj)	LT Bi, Bj, K

Return Jump Sequence

The return jump sequence controls the execution of three instructions.

00	Error exit to MA or program stop	PS
010	Return jump to K	RJ K
013	Central exchange jump to (Bj) + K or monitor exchange jump to MA	XJ BJ + K

Registers

The CP contains the operating and support registers which are located in the operand issue section (refer to figure 4-5).

The contents of these registers can be written into memory and reloaded from memory as a CYBER 170 exchange package by a single CP instruction (CYBER 170 exchange jump). Figure 5-1 shows the CYBER 170 exchange package.

The time a CYBER 170 exchange package resides in CP hardware is called an execution interval. During this interval, the contents of X, A, B, and P registers can be changed by CP instructions. The contents of other support registers change only as a result of a CYBER 170 exchange jump. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

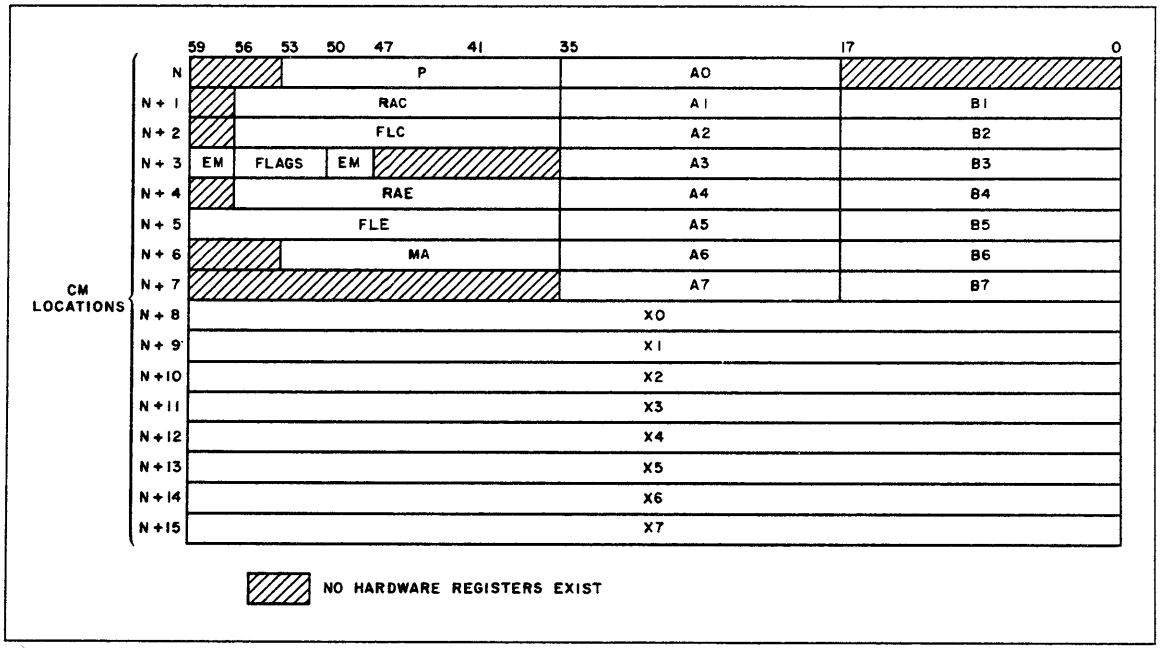


Figure 5-1. CYBER 170 Exchange Package

Operating Registers

The operating registers consist of operand (X), address (A), and index (B) registers. These registers minimize memory references for arithmetic operands and results.

X Registers

The CP contains eight 60-bit X registers, X0 through X7. The X0 register is used in the compare instructions to indicate if two fields of characters are equal. Also, the X0 register provides the relative UEM starting address in a block copy operation.

The X1 through X7 registers are primarily data handling registers for computation. X1 through X5 are used to input data from CM, and X6 and X7 are used to transmit data to CM.

Operands and results transfer between CM and the X registers as a result of placing CM addresses into corresponding A registers.

A Registers

The CP contains eight 18-bit A registers, A0 through A7. The A0 register serves as an intermediate register for the user's discretion. The A0 register is used in the compare collate instruction for the collate table address. Also, the A0 register provides the relative CM starting address in a block copy operation.

The A1 through A7 registers are essentially CM operand address registers associated one-for-one with the X registers. Placing a quantity into an address register (A1 through A5) causes a CM read reference to that address and transmits the CM word to the corresponding X register (X1 through X5). Similarly, placing a quantity into the A6 or A7 register causes the word in the corresponding X6 or X7 register to be written into that relative address of CM.

B Registers

The CP contains eight 18-bit B registers, B0 through B7. These registers are primarily indexing registers to control program execution. Program loop counts may also be incremented or decremented in these registers.

Program addresses may be modified on the way to an A register by adding or subtracting B register quantities. The B registers also hold shift counts for the nominal Bj shifts, the resultant exponent for the unpack, the operand exponent for the pack, and the resultant shift count from a normalize. The B0 register always contains positive zero which can be used as an operand. This register cannot hold results from instructions.

Support Registers

Eight support registers assist the operating registers during the execution of programs. The contents of the support registers are stored in CM, and their new contents are loaded from CM during a CYBER 170 exchange sequence. With the exception of the P register, the contents of the support registers cannot be altered during the execution interval of a CYBER 170 exchange package. When the execution interval completes, the data in the support registers is sent back to CM through a CYBER 170 exchange jump.

P Register

The 18-bit program address (P) register loads from CM during the first word of a CYBER 170 exchange sequence and contains the current program execution address. The register serves as a program address counter and holds the relative CM address for each program step.

RAC Register

The 21-bit CM reference address (RAC) register loads from CM during the second word of a CYBER 170 exchange sequence. An absolute CM address forms by adding RAC to a relative address determined by the instruction. The content of the P register is added to RAC to form the program address in CM. A P-equal-to-zero condition specifies relative address 0 and, therefore, (RAC). This CM location is reserved for recording error exit conditions and should not be used to store data or instructions.

FLC Register

The 21-bit CM field length (FLC) register loads from CM during the third word of a CYBER 170 exchange sequence. The FLC register defines the size of the field of the program in execution. Relative CM addresses are compared with FLC to check that the program is not going out of its allocated memory range.

EM Register

The 6-bit exit mode (EM) register loads from CM during the fourth word of a CYBER 170 exchange sequence. The EM register holds six exit mode selection bits that control individual error conditions for a program. Selected EM register bits cause the CP to error exit when the corresponding conditions occur. Any or all of the six bits can be set at one time. Clear EM register bits allow the CP to continue, without error processing, when most of the corresponding conditions occur. Refer to the error exit tables under Error Response in chapter 17 for specific cases. The exit mode selection bits appear in the exchange package as bits 48 through 50 and 57 through 59. The bits and their corresponding conditions are:

Mode Selection Bit	Significance
48	Address out of range
49	Infinite operand
50	Indefinite operand
57	Hardware error
58	Hardware error
59	Hardware error

Flag Register

The 6-bit flag register loads from CM during the fourth word of a CYBER 170 exchange sequence. The flag register holds six bits that function as control flags.

Bit	Condition
51	Hardware error bit.
52	Instruction stack (lookahead) purge flag. It set, extended purging of instruction lookahead registers is enabled. For further information, refer to Instruction Lookahead Purge Control under CP Programming in chapter 17.
53	CMU interrupted flag. If set, one of instructions 464 through 467 has been interrupted. The information necessary to resume operation has been saved.
54	Block copy flag. If set, block copy instructions (011, 012) use bits 30 through 50 of X0 rather than A0 to determine the CM address. For further information, refer to the descriptions of the block copy instructions in chapter 16.
55	Expanded addressing select flag. If set, UEM is operating in expanded addressing mode; if clear, UEM is operating in 24-bit standard addressing mode. For further information, refer to Addressing Modes under Memory Programming in chapter 17.
56	UEM enable flag. If set, UEM is available. This flag must be set to allow 011, 012, 014, and 015 instructions to access UEM.

RAE Register

The 21-bit UEM reference address (RAE) register loads from CM during the fifth word of a CYBER 170 exchange sequence. The lower 6 bits of this register are always zero. An absolute UEM address forms by adding RAE to the relative address which is determined by the instruction.

FLE Register

The 24-bit UEM field length (FLE) register loads from CM during the sixth word of a CYBER 170 exchange sequence. The lower 6 bits of this register are always zero. The FLE register defines the size of the field in UEM for the program in execution. Relative UEM addresses are compared with FLE.

MA Register

The 18-bit monitor address (MA) register loads from CM during the seventh word of a CYBER 170 exchange sequence. The MA register contains the absolute starting address of an exchange package which is used when executing a central exchange jump (013) instruction with the CYBER 170 monitor flag clear, or when honoring a monitor exchange jump to MA (262x) instruction with the CYBER 170 monitor flag clear. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

Execution Section

The execution section combines the operands into results, providing additional sequencing control where necessary.

Cache Memory

Cache memory is a high-speed buffer memory which is transparent to the user. It reduces effective CM access time by eliminating unnecessary CM references. When the CP first reads CM, a block of four words from CM (containing the requested word) is read rapidly into cache memory, these words may be instructions or data. On subsequent reading of any of these words, CM need not be accessed when these words are in cache memory. Often this is the case because the same data is read more than once, or because a loop of instructions is repeatedly executed. Cach memory is 2048 words or, optionally, 4096 words.

Addressing Section

An address adder calculates memory addresses for data and unconditional jump instructions.

Memory management hardware verifies that memory addresses are to access permitted memory areas. If this is the case, this hardware accesses cache memory and, if necessary, central memory.

Central Memory Control

Central memory control (CMC) provides an interface to CM for the CP and IOU. It is physically located in the CP cabinet. CMC includes:

- Ports and distributor
- SECDED logic
- Partial-write logic
- Memory control logic
- Maintenance registers

Central Memory

The CM performs the following functions.

- Without the Memory Upgrade Option, eight memory banks store from 524K to 2097K of 64-bit words (the leftmost 4 bits are undefined) and an 8-bit SECDED code.
- With the Memory Upgrade Option, eight memory banks store from 2097K to 16 776K of 64-bit words (the leftmost 4 bits are undefined) and an 8-bit SECDED code.
- The two ports make CM accessible to the CP and every PP.
- A bounds register limits access to CM from either or both ports.
- The SECDED generators generate the SECDED code bits stored with each word. SECDED checks circuits, corrects single-bit errors, and detects double-bit errors.
- The maintenance channel interface gives a PP in the IOU access to the CM maintenance registers for system initialization, corrective action, error reporting and diagnostics, and for setting the port bounds register.

Address Format Without Memory Upgrade Option

Figure 5-2 illustrates the address format without the Memory Upgrade Option.

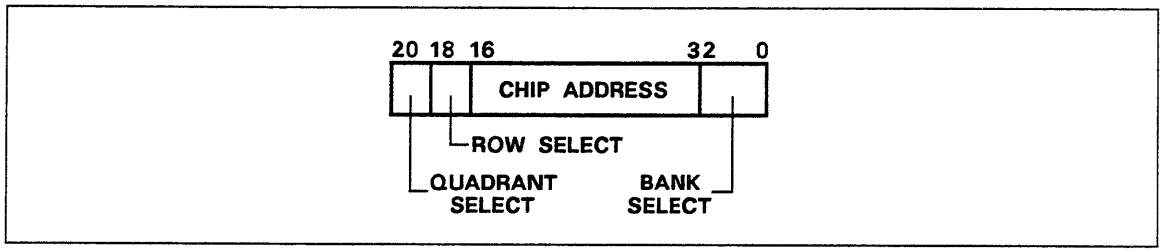


Figure 5-2. Address Format Without Memory Upgrade Option

The following list defines the address fields for figure 5-2.

- Bank Select specifies one of eight banks. Since the bank address is the lowest order 3 bits of the storage address, sequential addressing results in a phased-bank operation which allows a maximum data transfer rate of one word each clock period.
- Chip Address specifies the address of one word in 16K MOS memory chips for the selected bank.
- Row Select selects one of four word rows in a quadrant.
- Quadrant Select selects one of two quadrants (array packs). It is used only for storage units larger than 524K.

Address Format With Memory Upgrade Option

Figure 5-3 illustrates the address format with the Memory Upgrade Option.

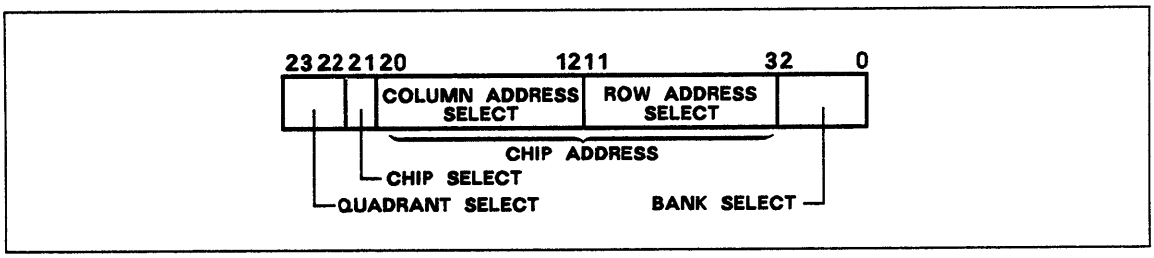


Figure 5-3. Address Format With Memory Upgrade Option

The following list defines the address fields for figure 5-3.

- Quadrant Select specifies one of four quadrants (array packs) within a bank.
- Chip Select, if set, enables the row address select to the upper half (720) of the 144 chips on memory boards in all eight memory banks. If clear, enables the lower half of the 144 chips on memory boards in all eight banks.
- Chip Address, which comprises column address select and row address select, specifies the address of one word on a chip for the selected bank and quadrant.
- Row Address Select specifies the row-select portion of the chip address on a chip.
- Column Address Select specifies the column-select portion of the chip address on a chip.
- Bank Select specifies one of eight banks.

CM Access and Cycle Times

The following paragraphs list CM access and cycle times. Without the Memory Upgrade Option, CM operates on an internal clock period of 64 nanoseconds. One 64-nanosecond clock period is referred to as a major cycle. With the Memory Upgrade Option, CM operates on an internal clock period of 32 nanoseconds. One 32-nanosecond clock period is referred to as a minor cycle.

	With CM Upgrade Option	Without CM Upgrade Option
Clock Period	32 ns (Minor cycle)	64 ns (Major cycle)

CM Access and Cycle Times Without Memory Upgrade Option

The CM access time for a read operation is 6 major cycles.

One bank cycle for a read or write operation is 7 major cycles. Cycle time for a partial write (read/modify/write) is 14 major cycles.

CM Access and Cycle Times With Memory Upgrade Option

The CM access time for a read operation is 10 minor cycles.

One bank cycle for a read or write operation is 12 minor cycles. Cycle time for a partial write (read/modify/write) is 24 minor cycles.

CM Ports and Priorities

A priority network resolves access conflicts on a rotating basis, preventing long-term lockout of any port. In case of simultaneous requests, the CP has priority. The CM also has a refresh mechanism which may consume a maximum of 7 percent (without the Memory Upgrade Option) and 4 percent (with the Memory Upgrade Option) of memory time. Refresh requests have priority over port requests. Refer to table 5-1 for maximum request lockout time.

Table 5-1. Port Priority

Port	Read or Write Requests
Refresh	1
Port 0	4
Port 1	5

Note:

Without the Memory Upgrade Option, 1 bank cycle equals 7 major cycles (448 nanoseconds). With the Memory Upgrade Option, 1 bank cycle equals 12 minor cycles (384 nanoseconds).

To present the information in this chapter in a structured format, this page has been left blank.

SECDED

The SECDED logic corrects single-bit errors during a CM read, permitting unimpeded computer operation. The SECDED logic prepares for the error correction by generating error correction code (ECC) bits for each data word and by storing these ECC bits in CM with the data word during the CM write. Table 5-2 lists the hexadecimal codes for all the combinations of syndrome bits with the number of the data bit assigned each code or a note categorizing the code. Then, during a CM read, CM performs the following SECDED sequence.

1. Read one CM word and generate new ECC bits for data portion of CM word.
2. Compare new ECC bits with CM word ECC bits.
3. If old and new ECC bits match, no error exists. Send data to requesting unit.
4. If bits do not match, generate syndrome bits from result of ECC compare.
5. Decode syndrome bits to determine if single or multiple bit failure.
6. If single bit failure, correct by inverting failing bit in data word. Send corrected word to requesting unit.
7. If multiple bit or other uncorrectable error, send uncorrectable error response code to CP or IOU. A PP in the IOU may then analyze the syndrome bits using the maintenance channel.

Table 5-2. SECDED Syndrome Codes/Corrected Bits

Code	Bit	Code	Bit	Code	Bit	Code	Bit
00	7	10	67[2]	20	66[2]	30	2/3[6]
01	71[2]	11	3	21	3	31	4
02	70[2]	12	3	22	3	32	4
03	6/7[6]	13	4	23	4	33	3
04	69[2]	14	3	24	3	34	4
05	3	15	4	25	4	35	3
06	3	16	4	26	4	36	3
07	24[1]	17	24[5]	27	28[5]	37	28[1]
08	68[2]	18	3	28	3	38	4
09	3	19	4	29	4	39	3
0A	3	1A	4	2A	4	3A	3
0B	16[1]	1B	16[5]	2B	20[5]	3B	20[1]
0C	4/5[6]	1C	4	2C	4	3C	3
0D	8[1]	1D	8[5]	2D	12[5]	3D	12[1]
0E	0[1]	1E	0[5]	2E	4[5]	3E	4[1]
0F	3	1F	4	2F	4	3F	3

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error with indicated bit(s) inverted.

6. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes.

7. No error detected.

(Continued)

Table 5-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
40	65[2]	50	3	60	3	70	56[1]
41	3	51	4	61	4	71	56[5]
42	3	52	4	62	4	72	60[5]
43	4	53	3	63	3	73	60[1]
44	3	54	4	64	4	74	58[5]
45	4	55	3	65	3	75	58[1]
46	4	56	3	66	3	76	62[1]
47	26[5]	57	26[1]	67	30[1]	77	30/62[5]
48	3	58	4	68	4	78	57[5]
49	4	59	3	69	3	79	57[1]
4A	4	5A	3	6A	3	7A	61[1]
4B	18[5]	5B	18[1]	6B	22[1]	7B	22/61[5]
4C	4	5C	3	6C	3	7C	59[1]
4D	10[5]	5D	10[1]	6D	14[1]	7D	14/59[5]
4E	2[5]	5E	2[1]	6E	6[1]	7E	6/63[5]
4F	4	5F	3	6F	3	7F	63[1]

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error with indicated bit(s) inverted.

(Continued)

Table 5-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
80	64[2]	90	3	A0	3	B0	48[1]
81	3	91	4	A1	4	B1	48[5]
82	3	92	4	A2	4	B2	52[5]
83	4	93	3	A3	3	B3	52[1]
84	3	94	4	A4	4	B4	50[5]
85	4	95	3	A5	3	B5	50[1]
86	4	96	3	A6	3	B6	54[1]
87	25[5]	97	25[1]	A7	29[1]	B7	29/54[5]
88	3	98	4	A8	4	B8	49[5]
89	4	99	3	A9	3	B9	49[1]
8A	4	9A	3	AA	3	BA	53[1]
8B	17[5]	9B	17[1]	AB	21[1]	BB	21/53[4]
8C	4	9C	3	AC	3	BC	51[1]
8D	9[5]	9D	9[1]	AD	13[1]	BD	13/51[5]
8E	1[5]	9E	1[1]	AE	5[1]	BE	5/55[5]
8F	4	9F	3	AF	3	BF	55[1]

1. Corrected single-bit error.
2. Syndrome code bit failed (single code bit set).
3. Double error or multiple error (even number of code bits set).
4. Multiple error reported as a single error.
5. Double error or multiple error with indicated bit(s) inverted.

(Continued)

Table 5-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
C0	0/1[5]	D0	40[1]	E0	32[1]	F0	2
C1	3	D1	40[4]	E1	32[4]	F1	3
C2	3	D2	44[4]	E2	36[4]	F2	3
C3	2	D3	44[1]	E3	36[1]	F3	2
C4	3	D4	42[4]	E4	34[4]	F4	3
C5	2	D5	42[1]	E5	34[1]	F5	2
C6	2	D6	46[1]	E6	38[1]	F6	2
C7	27[1]	D7	27/46[4]	E7	31/38[4]	F7	2
C8	3	D8	41[4]	E8	33[4]	F8	3
C9	2	D9	41[1]	E9	33[1]	F9	2
CA	2	DA	45[1]	EA	37[1]	FA	2
CB	19[1]	DB	19/45[4]	EB	23/37[4]	FB	23[1]
CC	2	DC	43[1]	EC	35[1]	FC	2
CD	11[1]	DD	11/43[4]	ED	15/35[4]	FD	15[1]
CE	3[1]	DE	3/47[4]	EE	7/39[4]	FE	7[1]
CF	2	DF	47[1]	EF	39[1]	FF	2

1. Corrected single-bit error.

2. Double error or multiple error (even number of code bits set).

3. Multiple error reported as a single error.

4. Double error or multiple error with indicated bit(s) inverted.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes.

CM Layout

Central memory contains an area that is reserved for special software called Virtual State software. Along with the hardware and microcode, this software handles the operations of Virtual State as described in chapter 17. Virtual State software is located at the higher end of memory. The remaining memory is available to the CYBER 170 State and may be allocated as central memory (accessible via RAC and FLC) or as unified extended memory (accessible via RAE and FLE and the 011, 012, 014, and 015 instructions). Refer to figure 5-4.

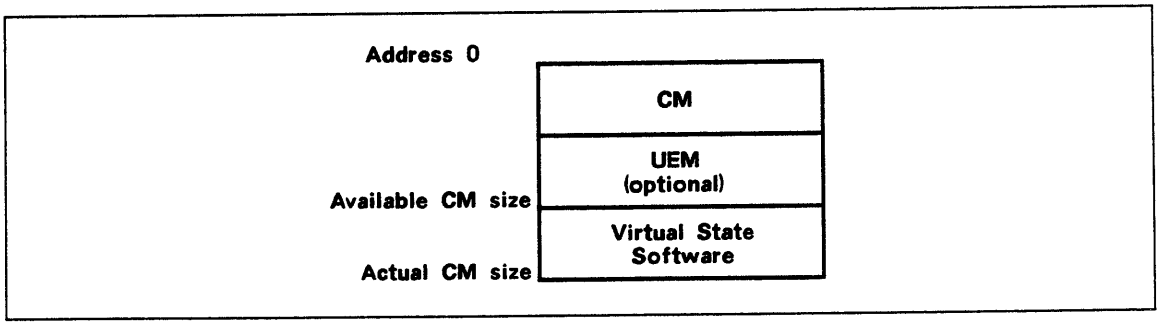


Figure 5-4. CM Layout

CM Bounds Register

The CM bounds register limits the write access to CM from specified ports. The ports are limited to the area between an upper and lower bound as specified in the CM bounds register. Bits in byte 0 specify the port(s) from which the write access is limited. The CM bounds register is set through the maintenance channel. For further information, refer to Maintenance Channel Programming in chapter 17.

Central Memory Reconfiguration

Central memory reconfiguration is a manually performed function that permits the computer operator to restructure the CM addresses so that a failing part of CM can be quickly locked out to provide a continuous block of usable CM. CM reconfiguration is accomplished by setting the switches on the memory unit to manipulate the upper address bits.

When a configuration switch is set forcing a CM address bit to a zero/one, the address range corresponding to the original installed memory accesses some parts of the reconfigured memory more than once. Addresses up to the rightmost forced bit, and half the addresses using the rightmost forced bit, cover a contiguous address space from location 0, which is the reconfigured memory. For further information, refer to chapter 6.

Input/Output Unit

The input/output unit (IOU) performs the functions required to locate, select, and initialize the external devices connected to the system, and controls the transfer of data between a selected device and CM. The IOU also performs system maintenance functions.

The IOU contains the following functional areas.

- Peripheral processor (PP)
- I/O channels
- Real-time clock
- Two-port multiplexer
- Maintenance channel
- CM access

Peripheral Processor

The basic IOU contains 10 PPs and can be expanded to 20 PPs in 5-PP increments. Each PP is a logically independent computer with its own memory. Each 5-PP group is organized into a multiplexing system which allows the PPs to share common hardware for arithmetic, logical, and I/O operations without losing independence. This multiplexing system comprises five ranks of registers termed a barrel. Each rank contains information related to the instruction being executed by one PP.

Each PP can communicate with the CP by issuing a CYBER 170 exchange request to a specific CYBER 170 exchange package associated with the issuing PP. In addition, a PP can also communicate with the CP via CM read and write operations. PPs can communicate with each other over the I/O channels and through CM.

Each PP executes programs alone or with other PPs to control data transfers between external devices and CM. These programs are comprised of instructions from the IOU instruction set and respond to requests issued through CM by the operating system. The programs translate generalized operating system requests into control functions for accessing the external devices and may also perform device scheduling and optimization. The programs use PP memory as a buffer for the data transfer between external devices and CM to isolate IOU data transfer from variations in CM transfer rate.

An IOU upgrade is available which is an optional concurrent input/output (CIO) subsystem consisting of five or ten PPs. Optional ISI and CYBER 170 direct-memory access (DMA) I/O channel adapters can be installed in the CIO.

Deadstart

A deadstart sequence allows the IOU to initialize itself. This deadstart sequence is initiated by the DEAD START switch on the deadstart panel or the DEAD START switch on the system console. The panel includes controls for assigning any PPM to PP0. For further information, refer to chapter 6.

Barrel and Slot

The barrel consists of the R, A, P, Q, and K registers, each one of which has five ranks 0 through 4. (Refer to figure 5-5.) Information in these registers moves from one rank to the next at a uniform 20-megahertz rate, providing a multiplexed system of five PPs, each operating at a 4-megahertz rate. The registers are stationary while the PPs rotate. For example, rank 4 registers contain PP0, PP1, PP2, PP3, and PP4 in succession, each consuming 50 nanoseconds of the total cycle time of 250 nanoseconds. Since PP memories operate at a slower rate, independent memory with its own address and data registers is provided for each PP.

Each time data enters the slot, a portion of the instruction for that data is executed. The slot performs tasks such as arithmetic and logic operations and program address manipulation. Complete execution of an instruction may require the R, A, P, Q, and K register quantities to go more than one trip around the barrel and through the slot.

The PPM may be referenced once each time the PP passes around the barrel and through the slot. During its slot time, the PP may also communicate with CM or with any of the I/O channels.

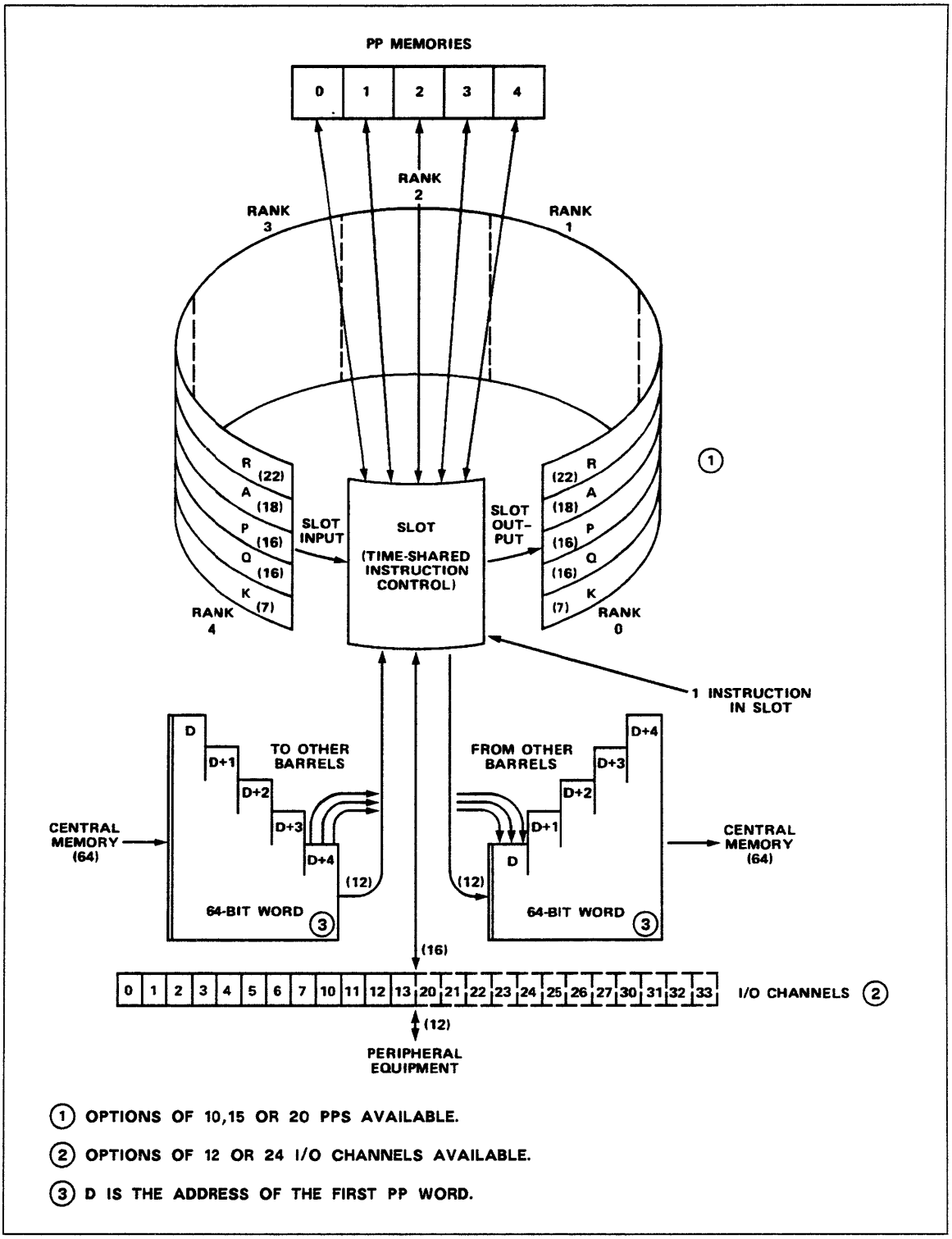


Figure 5-5. Barrel and Slot

PP Registers

The PP registers, which are discussed in the following paragraphs, are:

- R register
- A register
- P register
- Q register
- K register

R Register

The 28-bit R register, in conjunction with the A register, forms an absolute CM address for CM read/write instructions. When bit 17 of the A register is set, the absolute CM address is formed by appending six zeros to the lower end of the contents of the R register and adding to the result bits 0 through 16 of the contents of the A register (refer to figure 5-6).

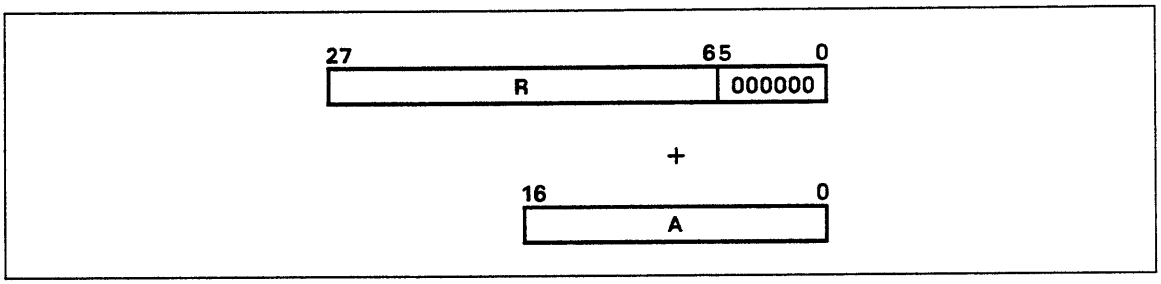


Figure 5-6. Formation of Absolute CM Address

A Register

The 18-bit A register holds one operand for arithmetic, logic, or selected I/O operations. The content of A may be an arithmetic operand, CM address, I/O function, or I/O data word. Various instructions operate on 6, 12, 16, or 18 bits of the A register.

When the A register is used as the CM address, parity is generated for transmission with the address to memory control. At deadstart, the A register is set to 10000 (octal). When bit 17 of the A register is clear, the A register is used as the CM address; however, when bit 17 is set, the R register is added to the A register (as described in the R register description) to obtain the absolute CM address for CM read/write instructions.

P Register

The 12-bit P register is the program address register, except during the execution of instructions 61, 63, 71, and 73. For these instructions, the P register contains the PPM address of the data transfer. At deadstart, the P register is set to zero.

Q Register

The 12-bit Q register holds data for several functions such as the address of the operand during direct addressing and indirect addressing, peripheral address of data used during one-word central read or write instructions, upper 6 bits during constant mode instructions, channel number on all I/O and channel instructions, shift count, and relative jump designator. At deadstart, each rank of the Q register is set to a corresponding PP number. Rank 0 is set to PP0, rank 2 is set to PP2, and so on.

K Register

The 12-bit K register is visible to the programmer through the maintenance channel only. This register holds the operation code field of an instruction for display on the IOU deadstart panel and for deadstart panel interrogation. When a PP is halted (idled), this register contains all ones.

PP Numbering

PPs are numbered in octal as follows:

Barrel	PPs
0	00 to 04
1	05 to 11
2	20 to 24
3	25 to 31

The deadstart sequence is used to determine PP numbering within a barrel. The sequence assigns barrel numbers according to the switch settings and, during the first minor cycle after deadstart, loads a zero into the Q register in barrel 0. This defines all the data in that rank of the barrel as belonging to PP0 and, since Q is the channel selector, assigns PP0 to channel 0. During the next minor cycle, Q loads with a one. This defines PP1 and assigns it to channel 1. This process occurs in parallel in all barrels until the IOU assigns each rank of each barrel with a PP number and a channel number. Reassignment can be done only during a deadstart.

PP Memory

Each PP has an independent 4K word memory. Each word contains 16 data bits with the upper 4 bits set to zero and 1 parity bit. PP0 executes the deadstart program from the deadstart panel during the deadstart operation. Therefore, PP memory 0 must be operational. A PP memory reconfiguration feature allows the user to restore IOU operation if the IOU detects a fault in the PP memory normally assigned to PP0.

To reconfigure, the operator assigns a good PP memory to PP0 and the operating system removes the failing PP memory. Computer operation can continue without the failing PP memory, and repairs can be made during scheduled maintenance. The system must be deadstarted to reconfigure PPMs.

I/O Channels

The I/O channels are comprised of an internal interface that allows common hardware and software to control the external devices, and an external interface that allows the IOU to communicate with the external devices using 12-bit data channels. The internal interface can transfer 16-bit data words between two PPs, or between a PP and an external device at a maximum rate of one word every 250 nanoseconds. This rate can be sustained for the maximum practical channel transfer (4096 words). During transfers between PPs, if the PPs are in the slot at the same time, the transfer rate is 500 nanoseconds.

Any PP can access any of the CYBER 170 bidirectional I/O channels. All PPs communicate with external devices through the independent I/O channels. Each channel may be connected to one or more pieces of external equipment, but only one piece of equipment can use a channel at one time. All channels can be active simultaneously. Available channels are listed as follows:

- Twenty-four CYBER 170 compatible I/O channels available with a maximum data transfer rate of 3 megabytes per second.
- An optional DMA-enchanced intelligent standard interface (ISI) channel adapter or CYBER 170 channel adapter that can be installed in any one of ten channel locations in the CIO cabinet. The adapters transfer data between the ISI or CYBER 170 channel and PP memory using standard I/O instructions. They also support DMA transfer in which data goes directly between CM and an external device without going through the PP. There are two types of CYBER 170 DMA transfers, fast and normal. Fast transfers are used with the Extended Semiconductor Memory-II (ESM-II), and normal transfers are used with other CYBER 170 external devices.

The display station controller (DSC) is attached to CYBER 170 channel 10_8. The DSC is the IOU interface between the PPs and the system console, servicing both the keyboard and the cathode-ray tube (CRT). It transmits function words and digital symbol size/position data to the system console, and receives digital character codes from the keyboard. It also receives digital symbol codes from the PPs and converts these to analog signals to the CRT.

Real-Time Clock

The real-time clock is a 12-bit free-running counter, incrementing at a 1-megahertz rate. It is permanently attached to channel 14_8. This channel may be read at any time as its active and full flags are always set.

Two-Port Multiplexer

The two-port multiplexer provides communication capability between a PP and two attached terminals. One port is reserved for maintenance purposes and the other port is reserved for future use. The two-port multiplexer is permanently attached to channel 15_8.

Maintenance Channel

The maintenance channel is used for initialization of the CP and CM maintenance registers and monitoring of error status.

The maintenance channel consists of the maintenance channel interface on channel 17_8, a maintenance access control in each system element, and a set of interconnecting cables.

Central Memory Access

Any PP can access CM. During a write from the IOU to CM, the IOU assembles five successive 12-bit PP words into a 64-bit CM word with the leftmost 4 bits undefined. During a CM read, the IOU disassembles the rightmost 60 bits of the 64-bit CM word into five PP words. To find the CM address, a PP reads the A register. If bit 17 of the A register is clear, the PP uses the contents of the A register for the CM address. If bit 17 of the A register is set, the PP adds the relocation address from the R register to the A register to form the CM address.

A maximum of 20 PPs in various stages of assembly/disassembly can simultaneously read CM words, and five PPs can write CM words.

Models 845 and 855 Operating Instructions 6

Controls and Indicators 6-1
Deadstart Panel Controls/Indicators 6-1
Central Memory Controls 6-6

Power-On and Power-Off Procedures 6-14

Operating Procedures 6-14
Control Checks 6-14
Deadstart Sequences 6-15

IOU Reconfiguration 6-16

Models 845 and 855 Operating Instructions 6

This chapter describes mainframe controls and indicators and the operating procedures which are hardware dependent. Software-dependent procedures are in system software reference manuals listed under Additional Related Manuals in About This Manual.

Controls and Indicators

This section describes IOU deadstart controls and indicators and CM configuration switches used by the system operator. Other controls used by maintenance personnel are described in the hardware operator's guide and the hardware maintenance manuals of the power distribution and warning system, the cooling system, and the system console listed in the system publication index in About This Manual.

Deadstart Panel Controls/Indicators

The deadstart panel (figure 6-1) is in the IOU. It contains PP register selection and display facilities, deadstart controls, error indicators, and a switch matrix, which is the source for a short PP program for initialization or troubleshooting. The switches, indicators, and their functions are listed in table 6-1.

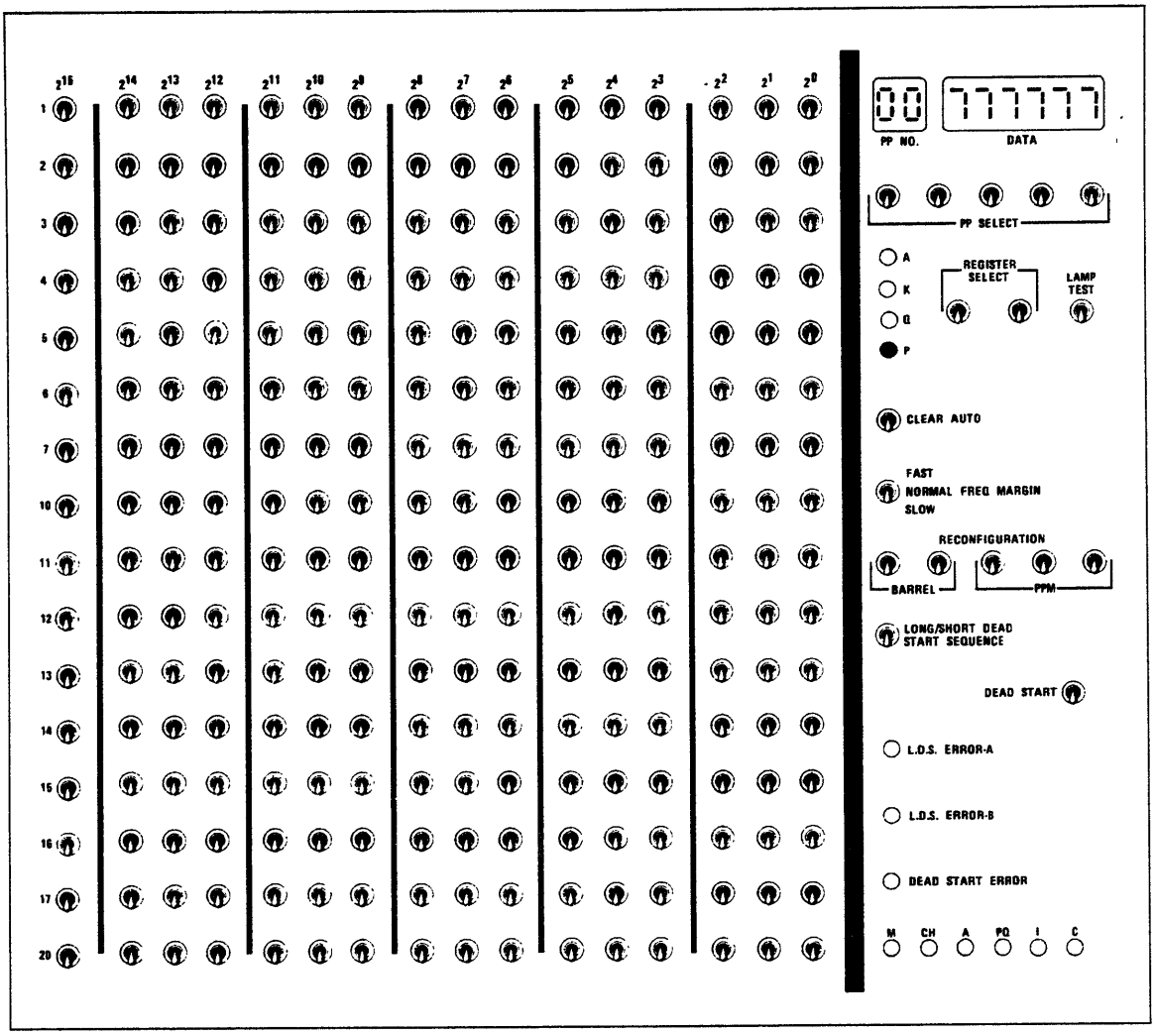


Figure 6-1. Deadstart Panel

Table 6-1. Deadstart Panel Controls/Indicators

Panel Nomenclature	Description	Function
20 through 215 by 1 through 208_8	Toggle switch matrix (two-position switches)	Provides a 16-word deadstart program for PP0. Switches 2^0 through 2^{11} set 12 bits for each of the program words, labeled 1 through 20 (octal). Switches 2^{12} through 2^{15} are set to zero. Up position sets bit. Down position clears bit.
PP NO	Octal display	Shows the PP selected by PP SELECT switches.
DATA	Octal display	Shows the content of the register selected by REGISTER SELECT switches.
PP SELECT	Toggle switches (two-position)	Selects the PP whose register is to be displayed.
REGISTER SELECT	Toggle switches (two-position)	Selects the register to be displayed (00 = P, 01 = Q, 10 = K, 11 = A).
A, K, Q, P	Indicators	One of these lights to indicate which register is selected by REGISTER SELECT switches.
LAMP TEST	Toggle switch (two-position)	Lights all indicators and display segments.
CLEAR AUTO	Toggle switch (two-position)	Allows manual clearing of auto-mode bit (bit 34 of the environment control register) to override possible auto-mode selection. This allows the selection of the PP and register from the deadstart panel if bit 34 is set.
FREQ MARGIN	Toggle switch (three-position)	Determines the frequency margin selected (FAST/NORMAL/SLOW). The setting of this switch is sensed only at deadstart time.

(Continued)

Table 6-1. Deadstart Panel Controls/Indicators *(Continued)*

Panel Nomenclature	Description	Function
RECONFIGURATION, BARREL	Toggle switches (two-position)	Selects the physical barrel which is logical barrel 0. All the other logical barrels are numbered from the selected physical barrel circularly. (If physical barrel 1 is selected by the switches, physical barrel 2 is logical barrel 1, and so on.)
RECONFIGURATION, PPM	Toggle switches (two-position)	Selects the physical PP memory which is logical PPM0. All the other PPMs in all barrels are numbered from the selected physical PPM circularly. If the switches are set to a value greater than four, no reconfiguration takes place.
LONG/SHORT DEAD START SEQUENCE	Toggle switch (two-position)	Selects the LONG/SHORT deadstart sequence. The setting of this switch is sensed only at deadstart.
DEAD START	Toggle switch (three-position, center is off)	Selects the fast or slow repetitive deadstart, which generates a master clear pulse every 250 or 4000 microseconds respectively. Up position selects fast deadstart; down position selects slow deadstart. (The single deadstart control pushbutton is on the system console.)
L.D.S. ERROR-A	Indicator	Remains lit when long deadstart branch tests are not completed within 10.25 microseconds.
L.D.S. ERROR-B	Indicator	Remains lit when a long deadstart sequence does not go to completion.
DEAD START ERROR	Indicator	Lights in case of long deadstart ROM address/data parity error.

(Continued)

Table 6-1. Deadstart Panel Controls/Indicators *(Continued)*

Panel Nomenclature	Description	Function	
M, CH, A, PQ, I, C	Indicators	Lights in case of hardware failures as follows:	
		M:	PP memory failure
		CH:	I/O channel failure
		A:	A barrel failure
		PQ:	P or Q barrel failure
		I:	Firmware or control failure
		C:	12/16 conversion failure

Central Memory Controls

The CM, without the Memory Upgrade Option contains four three-position configuration switches (figure 6-2). The CM with the Memory Upgrade Option contains six two-position configuration switches. Without the Memory Upgrade Option, these switches are located along the edge of a printed-circuit board located just to the right of the center post in the lower section of the memory cabinet (location F04). With the Memory Upgrade Option, these switches are located along the address interface pak switch in the A section of the memory cabinet.

The switches are used to eliminate CM sections with malfunctions. Without the Memory Upgrade Option, each switch, SW3 through SW6, forces one corresponding CM address bit, 23 through 20, either to a zero (switch down) or to a one (switch up). Refer to table 6-2. With the Memory Upgrade Option each switch, SW0 through SW5, inverts the corresponding CM address bit, 37 through 42. The inversion effectively moves blocks of bad memory to the highest memory block and moves blocks of good memory down, thereby providing a sequentially addressable block of error-free memory. Refer to table 6-3.

In case of CM malfunctions, the remaining good memory can be reconfigured so it is accessible by contiguous addresses from 0 to the maximum remaining address. This is accomplished by setting configuration switches (figure 6-2) as listed in tables 6-2 and 6-3. Refer to the hardware operator's guide listed in the system publication index in About This Manual for further information.

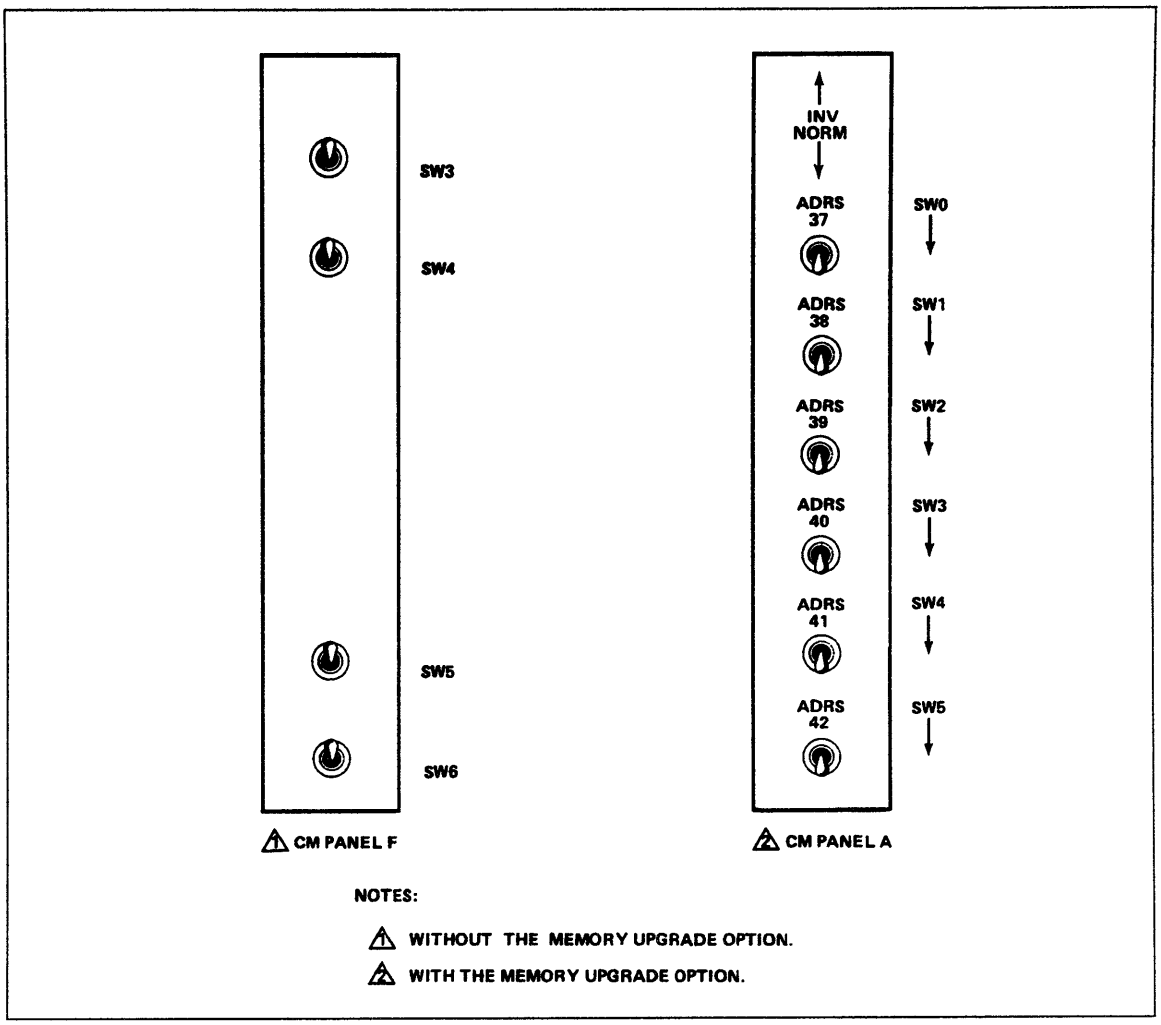


Figure 6-2. CM Configuration Switches

Table 6-2. Central Memory Reconfiguration (Without Memory Upgrade Option)

Original CM		Reconfigured CM				
Location of Failing CM[1]						
Words (Size)	**Address Range**	**Words (Size)**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**
524K (4 MB)	0-1 777 777	262K (2 MB)			0	X
		262K (2 MB)			1	X
1049K (8 MB)	0-3 777 777	524K (4 MB)		0	X	X
		524K (4 MB)		1	X	X
1573K (12 MB)	0-5 777 777	524K (4 MB)	0	0	X	X
		524K (4 MB)	0	1	X	X
		1049K (8 MB)	1	0	X	X
2097K (16 MB)	0-7 777 777	1049K (8 MB)	0	X	X	X
		1049K (8 MB)	1	X	X	X

1. CM remaining can be further reconfigured by setting additional configuration switches.

(Continued)

Table 6-2. Central Memory Reconfiguration (Without Memory Upgrade Option) *(Continued)*

Original CM		Reconfigured CM				
Reconfiguration Setting[2]						
Words (Size)	**Address Range**	**Words (Size)**	**SW3**	**SW4**	**SW5**	**SW6**
524K (4 MB)	0-1 777 777	262K (2 MB)	-	-	U	-
		262K (2 MB)	-	-	D	-
1049K (8 MB)	0-3 777 777	524K (4 MB)	-	U	-	-
		524K (4 MB)	-	D	-	-
1573K (12 MB)	0-5 777 777	524K (4 MB)	-	U	-	-
		524K (4 MB)	-	D	-	-
		1049K (8 MB)	D	-	-	-
2097K (16 MB)	0-7 777 777	1049K (8 MB)	U	-	-	-
		1049K (8 MB)	D	-	-	-

2. U equals up, D equals down, and dash (-) equals center position.

Table 6-3. Central Memory Reconfiguration (With Memory Upgrade Option)

Original CM		Reconfigured CM (Reconfiguration Settings)			
Words (Size)	Address Range	Error-Free Size	SW0 ADRS 37	SW1 ADRS 38	SW2 ADRS 39
2097K (16 MB)	0-7 777 777	1049K (8 MB)	D	D	D
4195K (32 MB)	0-17 777 777	2097K (16 MB)	D	D	U
8390K (64 MB)	0-37 777 777	4195K (32 MB)	D	U	D
16780K (128 MB)	0-77 777 777	8390K (64 MB)	U	D	D

Notes:

1. CM remaining can be further reconfigured to obtain larger contiguous blocks of error-free memory by setting additional configuration switches. See examples shown in figure 6-3.

2. U equals up, D equals down. Normal setting of all switches is down.

(Continued)

Table 6-3. Central Memory Reconfiguration (With Memory Upgrade Option) *(Continued)*

Original CM		Reconfigured CM (Reconfiguration Settings)			
Words (Size)	**Address Range**	**Error-Free Size**	**SW3 ADRS 40**	**SW4 ADRS 41**	**SW5 ADRS 42**
2097K (16 MB)	0-7 777 777	1049K (8 MB)	U	D	D
4195K (32 MB)	0-17 777 777	2097K (16 MB)	D	D	D
8390K (64 MB)	0-37 777 777	4195K (32 MB)	D	D	D
16780K (128 MB)	0-77 777 777	8390K (64 MB)	D	D	D

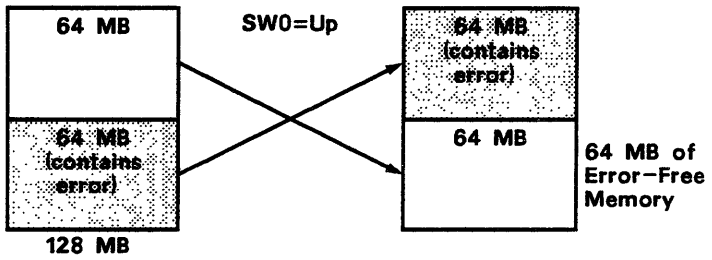
Notes:

1. CM remaining can be further reconfigured to obtain larger contiguous blocks of error-free memory by setting additional configuration switches. See examples shown in figure 6-3.

2. U equals up, D equals down. Normal setting of all switches is down.

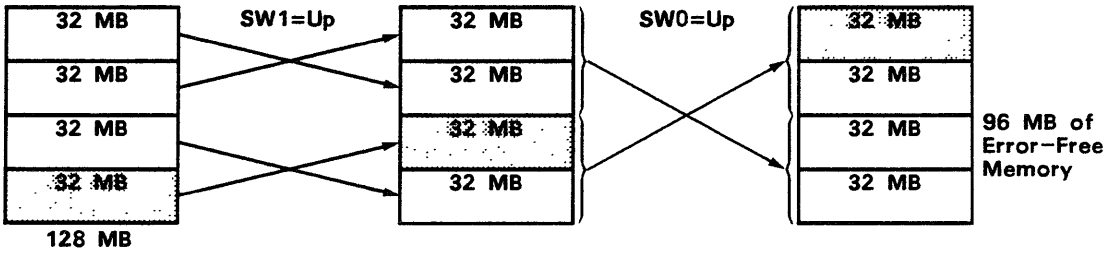
To present the information in this chapter in a structured format, this page has been left blank.

Set SW0 up to move block of memory containing error to upper half of memory.



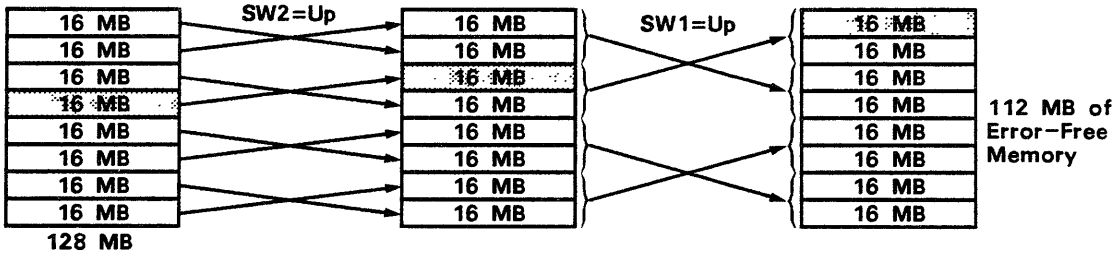
Error in lower 64-MB block of 128-MB memory.

Set SW1 up to move 32-MB block containing error to next higher 32 MB. Then set SW0 up to move block containing error to highest block of memory.



Error in lowest 32-MB block of 128-MB memory.

Set SW2 up to move 16-MB block containing error to next higher block. Then set SW1 up to move 32-MB block containing error to highest block of memory.



Error in lowest 16-MB block of 128-MB memory.

Figure 6-3. Reconfiguration Examples (With Memory Upgrade Option)

Power-On and Power-Off Procedures

In case of an emergency, use the system EMERGENCY OFF switch. The power-on and power-off procedures are described in the hardware operator's guide listed in the system publication index in About This Manual.

CAUTION

Improper application or removal of power may damage system circuits and/or air conditioning system. Power must be turned on/off by designated personnel only, except for the system EMERGENCY OFF switch. Use only for extreme emergency, not for normal shutdown.

Operating Procedures

Refer to the hardware operator's guide listed in the system publication index in About This Manual. The system is initialized by setting its control switches, and then by running either a long or short deadstart sequence (defined later in this section). After initialization, the keyboard is used to instruct the system further, under program control.

Control Checks

Before activating a long or short deadstart sequence, check the positions of deadstart panel switches against their intended use. These checks can be made by using table 6-4. The normal settings of these switches is as follows:

Switch	Position
CLEAR AUTO	Down
FREQ MARGIN	Center
RECONFIGURATION	All down
LONG/SHORT DEAD START SEQUENCE	Down for a short deadstart sequence
DEAD START	Center
All error lights	Not lit

Deadstart Sequences

In response to a deadstart signal from either the deadstart pushbutton on the system console or from the DEAD START switch on the deadstart panel, circuits in the IOU perform a deadstart sequence. Depending on the setting of the LONG/SHORT DEAD START SEQUENCE switch on the deadstart panel, either the long or the short deadstart sequence is performed. The short deadstart sequence is used when hardware integrity verification is not required. The long deadstart sequence performs all the tasks performed by the short deadstart sequence and some additional tasks. The main additional task is the running of a diagnostic program, from a read-only memory (ROM) in the IOU, on logical PP0. The diagnostic program takes approximately 1 minute to run.

Both deadstart sequences begin with a master clear which sets up all PPs, except logical PP0, for a 4096-word block input starting at PP location 0. The input into each PP is from the channel with the same number as the logical number of the PP concerned. The master clear also resets all external devices and sets maintenance channel connect code bit 52. The individual channels and registers are set as follows.

Channel	Active/ Inactive Flag	Full/Empty Flag	Channel Flag	Channel Error Flag
0	Inactive	Empty	Clear	Clear
10 (display controller)	Active	Empty	Clear	Clear
14 (real-time clock)	Active	Full	Set	Set
15 (two-port mux)	Active	Empty	Clear	Clear
17 (maintenance)	Active	Empty	Clear	Clear
Other installed channels	Active	Empty	Clear	Clear
Noninstalled channels	Inactive	Empty	Clear	Clear

The flags of channel 14 and of noninstalled channels are fixed by hardware and cannot be changed.

Register	Initialization[1]	Description
K	007100_8	Instruction display on deadstart panel
P	007777_8	Causes block input to start from location 0
A	$10{,}000_8$	Count of 4096 words
Q	0, 1, 2...	I/O channel numbers (PP0: 0, PP1: 1, and so on)

All registers in all barrels are set to these values, except the registers of PP0.

1. Leading zeros are not displayed on deadstart panel.

If the long deadstart sequence is being performed, hardware clears location 7777_8 in all PP memories and sets the P register of PP0 to 6000_8. PP0 starts performing a test program from a read-only memory in IOU and lights the deadstart panel L.D.S. ERROR-A and L.D.S. ERROR-B indicators. Indicator A remains lit unless the test program reaches location 6200_8 within 10.25 microseconds. Indicator B remains lit until the test program reaches location 7776_8. When this happens, the unique part of the long deadstart sequence ends with a master clear.

Next, both deadstart sequences clear PP0 location 0, write the settings of the deadstart panel matrix switches into PP0 memory locations 1_8 to 20_8, and clear PP0 location 21_8. PP0 then starts executing the program entered from the matrix switches, which is normally a bootstrap program to input more data from an assigned external device.

The short deadstart sequence does not disturb PP memory other than PP0 locations 0 to 21_8. Both deadstart sequences leave all PPs, except PP0, waiting for a block input or for action through the maintenance channel. After the block input is complete, each PP starts executing the program entered from whatever address was entered into location 0 of that PP.

IOU Reconfiguration

The logical PP numbers and hardware are assigned to physical PPs circularly from the settings of IOU deadstart panel RECONFIGURATION switches, which specify which physical barrel and PPM is PP0. If the PPM section of these switches is set to a value greater than 4, the value 0 is substituted. If the BARREL section of these switches is set to a value greater than the number of installed barrels, the value 0 is substituted. Thus, possible barrel numbering is as described in table 6-4.

NOTE

The minimum system option is 10 PPs.

Table 6-4. Barrel Numbering Table

Barrels Installed	Physical Barrel	Logical PPs in Physical Barrel with BARREL RECONFIGURATION Switch Values			
		0	1	2	3
4 Barrels (20 PPs)	0	0-4	25-31	20-24	5-11
	1	5-11	0-4	25-31	20-24
	2	20-24	5-11	0-4	25-31
	3	25-31	20-24	5-11	0-4
3 Barrels (15 PPs)	0	0-4	20-24	5-11	(0-4)
	1	5-11	0-4	20-24	(5-11)
	2	20-24	5-11	0-4	(20-24)
2 Barrels (10 PPs)	0	0-4	5-11	(0-4)	(0-4)
	1	5-11	0-4	(5-11)	(5-11)
1 Barrel (5 PPs)	0	0-4	(0-4)	(0-4)	(0-4)

Models 840, 850, and 860 System Description 7

Physical Characteristics 7-1

Functional Characteristics 7-3
- Central Processor 7-4
- Central Memory 7-5
- Input/Output Unit 7-6

Major System Component Descriptions 7-7
- Central Processor 7-7
  - Instruction Section 7-7
  - Registers 7-7
  - Execution Section 7-8
  - Cache Memory 7-8
  - Addressing Section 7-8
  - Central Memory Control 7-8
- Central Memory 7-9
- Input/Output Unit 7-10
- System Console 7-11

Models 840, 850, and 860 System Description 7

This chapter describes the physical and functional characteristics and major system components.

These high-speed computer systems are for both business and scientific applications. The systems include the following components.

- Central processor (CP)
- Central memory (CM)
- Input/output unit (IOU)

Physical Characteristics

The mainframe configuration for models 840 and 850 and the single-CP 860 include an interconnected three-section cabinet for the CP, CM, and IOU. (The system console is also required for system operation.) The model 860 supports an optional second CP, which is contained in an additional one-bay section. Refer to figures 7-1 and 7-2. The configuration in figure 7-2 contains the optional IOU-NIO/CIO.

Each cabinet section contains a logic chassis with plug-in circuit boards. The CP cabinet section comprises three attached subsections, each with separate power and cooling facilities. A separate cooling unit provides cooling for the CP subsections and CM. The IOU cabinet section has a self-contained cooling unit to cool the IOU logic chassis. Each cabinet section also contains an ac/dc control section with voltage margin testing facilities and dc power supplies. For additional cooling or power information, refer to the cooling system and power system manuals listed in the system publication index.

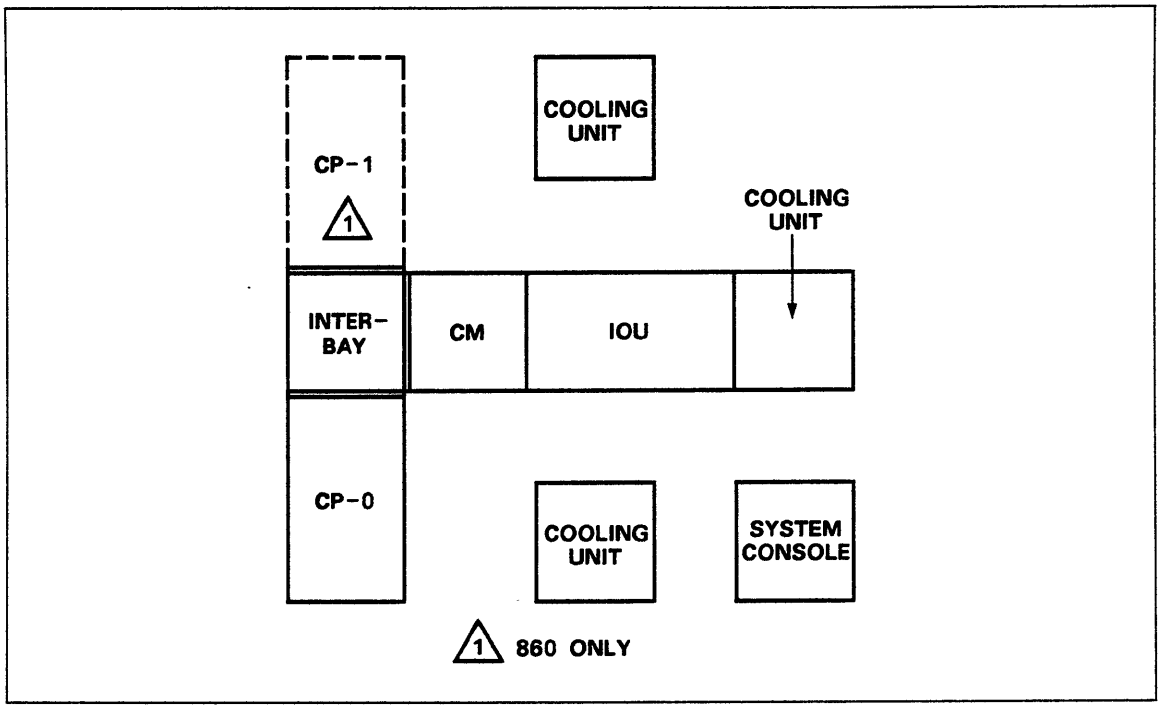


Figure 7-1. System Configuration

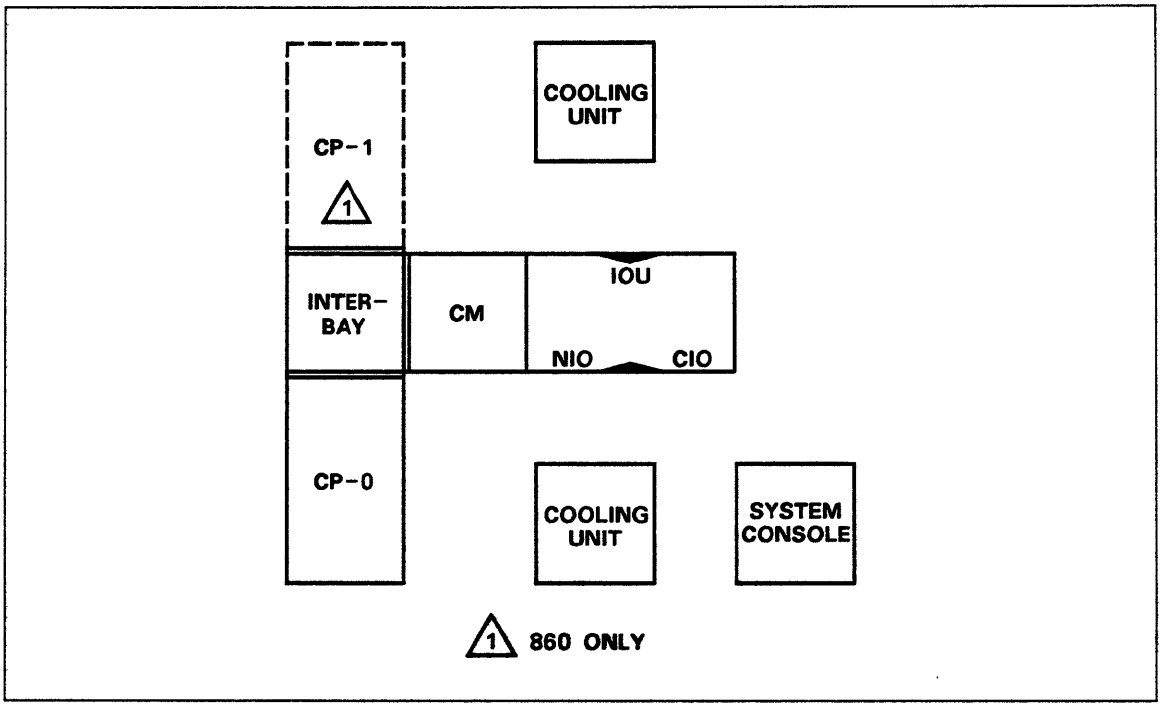


Figure 7-2. System Configuration With Optional IOU-NIO/CIO Upgrade Option

Functional Characteristics

To achieve high computation speeds ECL and large-scale integration (LSI) logic is used. High speed is also the objective of the CP design, which is based on the assumption that both data and instructions are, in most cases, accessed from successive memory locations. Accordingly, the CP prefetches both instructions and data expected to be used next while the current instruction is being processed.

The CP supports two states of operation.

Virtual State	Operates with virtual-memory byte addressing, using the Virtual State instruction set and data formats. Virtual State is the native operating state of the CP.
CYBER 170 State	Operates with real-memory word addressing, using the CYBER 170 State CP instruction set and data formats.

The Virtual State and CYBER 170 State environments may be present at the same time with the CP executing in either environment.

- NOS/VE is the operating system of Virtual State.
- NOS is the operating system of CYBER 170 State.

The semiconductor central memory is divided into eight independent banks. These banks may all be simultaneously in the process of completing read/write requests which are queued and distributed at ECL speeds. System input/output speeds are determined by the capabilities of existing external devices.

Central Processor

The CP has the following characteristics:

- 60-bit internal word.
- Eight 60-bit operand (X) registers.
- Eight 18-bit address (A) registers.
- Eight 18-bit index (B) registers.
- Two registers that isolate each user's central memory space (RAC, FLC).
- Two registers that isolate each user's extended memory space (RAE, FLE).
- Register exchange instructions (exchange jumps) for interrupting programs.
- Floating-point (FP) arithmetic (10-bit exponent plus sign bit, 48-bit coefficient plus sign bit). Some FP instructions use 96-bit (double-precision) coefficients.
- Integer arithmetic (60/18-bit operands).
- Character string compare/move facilities (6-bit characters).
- Packed instructions (15/30/60-bit instructions in 60-bit words).
- Synchronous internal logic.
- 64-nanosecond clock period.
- 2048-word cache buffer memory, option available for 4096-word cache.
- Instruction and branch instruction lookahead.
- Microcode control.
- Parity checking of all major data and address paths.
- Maintenance channel to IOU.

Central Memory

Timing references are from CMC/CPU interface.

- 72-bit data word (60 data bits, 8 single-error correction/double-error detection bits, and 4 unused bits).
- 2097K words (16 megabytes) of dynamic random-access memory, options available to 16 776K words (128 megabytes).
- Organization of eight independent banks.
- Two memory ports (located in the central processor cabinet).
- Bounds register to limit write access.
- 64-nanosecond clock period.
- Maximum data transfer rate of one word every 32 nanoseconds.
- 464-nanosecond read access time.
- 384-nanosecond read/write cycle time.
- 768-nanosecond partial write cycle time.
- Read and write data queuing capability.
- Single-error correction/double-error detection (SECDED) on stored data.
- Parity checking of all major data, address, and control paths.
- Unified-extended memory (UEM) which serves as extended memory within CM.

Input/Output Unit

The IOU has the following characteristics:

- Ten peripheral processors (PPs), 15-PP/20-PP options available. Each PP has 4K independent memory (PPM) comprised of 16-bit words with the upper 4 bits zero.
- Port to central memory.
- Bounds register to limit writes to central memory.
- Twelve 12-bit CYBER 170 channels to external devices, 24 channel option available.
- Real-time clock (channel 14_8).
- Display controller (CYBER 170 channel 10_8).
- Two-port multiplexer (channel 15_8).
- Maintenance channel (channel 17_8).
- Parity checking on all major data and address paths.
- Operating speed of 250 nanoseconds and a minor cycle of 50 nanoseconds.
- Optional concurrent input/output (CIO) PPs and direct-memory access (DMA) I/O channel adapter. Available only on systems with IOU-NIO/CIO cabinet upgrades.

Major System Component Descriptions

Central Processor

The CP hardware (figure 7-3) consists of the following:

- Instruction section
- Registers
- Execution section
- Cache memory
- Addressing section
- Central memory control

Central Processor 0 (CP-0) is capable of dual-state operation: Virtual State and CYBER 170 State. If a second central processor is added (CP-1), it is capable of only Virtual-State operation.

The CP is isolated from the IOU and is thus able to carry on computation or character manipulation unencumbered by I/O requirements.

Instruction Section

The instruction section directs the arithmetic and manipulative functions for instruction execution. The instruction section prefetches instruction words from memory and disassembles them into instructions.

Registers

Operating registers reduce storage accesses for operands used during the execution of an instruction. These registers are:

- Eight 60-bit X registers (X0 through X7) which hold operands used for computation.
- Eight 18-bit A registers (A0 through A7) which use A0 primarily for indexing and A1 through A7 for CM operand addressing.
- Eight 18-bit B registers (B0 through B7) which are primarily indexing registers to control program execution. The B0 register always contains all zeros.

Eight support registers support the operating registers during program execution. These registers are:

- 18-bit program address (P) register.
- 21-bit reference address for CM (RAC) register. This is a program's lower bound.
- 21-bit field length for CM (FLC) register. This is a program's upper bound.
- 6-bit exit mode (EM) register.
- 6-bit flag register.
- 21-bit reference address for UEM (RAE) register.
- 24-bit field length for UEM (FLE) register.
- 18-bit monitor address (MA) register.

The registers store data and control information, present operands to the execution section, and store results.

Execution Section

The execution section combines the operands to achieve the result.

Cache Memory

The cache memory consists of two sets of fast bipolar memory, capable of storing 2048 60-bit words. It can be expanded to four sets with a capacity of 4096 words. The memory addressing sections determine whether a requested word is in the cache memory. If it is not, they read four consecutive words from central memory into the cache memory.

Addressing Section

The addressing section checks memory addresses against the CP registers RAC, FLC, RAE, and FLE to ensure isolation of user memory space.

Central Memory Control

Central memory control (CMC) is integrated within the CP and controls the flow of data between CM and requesting system components.

Central Memory

The CM (figure 7-3) consists of the following:

- Eight memory banks
- Memory ports

The CM with the Memory Upgrade Option is a dynamic random access organized into eight independent banks.

A portion of CM can be reserved for use as extended memory. It is called unified-extended memory (UEM), and is referenced by the RAE and FLE registers. The memory ports are located in the central processor cabinet, with one port having a queuing buffer.

Input/Output Unit

The IOU (figure 7-3) consists of the following:

- Ten logically independent peripheral processors (PPs). Options are available to increase total to 15 or 20 PPs.
- Internal interface to 12 I/O channels. 24-channel option is available.
- External interfaces to I/O channels.
  - 11 or 23 CYBER 170 channel interfaces.
  - Display controller interface (CYBER 170 channel 10_8).
  - Real-time clock interface (channel 14_8).
  - Two-port multiplexer interface (channel 15_8).
  - Maintenance channel interface (channel 17_8).
- Interface to central memory.
- Bounds register to limit writes to CM.
- Optional CIO PPs and DMA I/O channel adapter. Available only on systems with IOU-NIO/CIO cabinet upgrades.

The PPs are organized in groups of five, called barrels. The PPs in a barrel time-share common hardware. Each PP has its own independent memory and communicates with all I/O channels and central memory.

System Console

The system console, required for system operation, provides a visual, alphanumeric readout for the computer. The receipt of symbol and position information from the computer enables displaying program information on a cathode-ray tube (CRT). The station also contains an alphanumeric keyboard which enables an operator to send data to the computer. The keyboard and CRT combination permits the computer operator to monitor and control system operation. Except for programming information in chapter 17, refer to the system console manual listed in additional related manuals in About This Manual for further information.

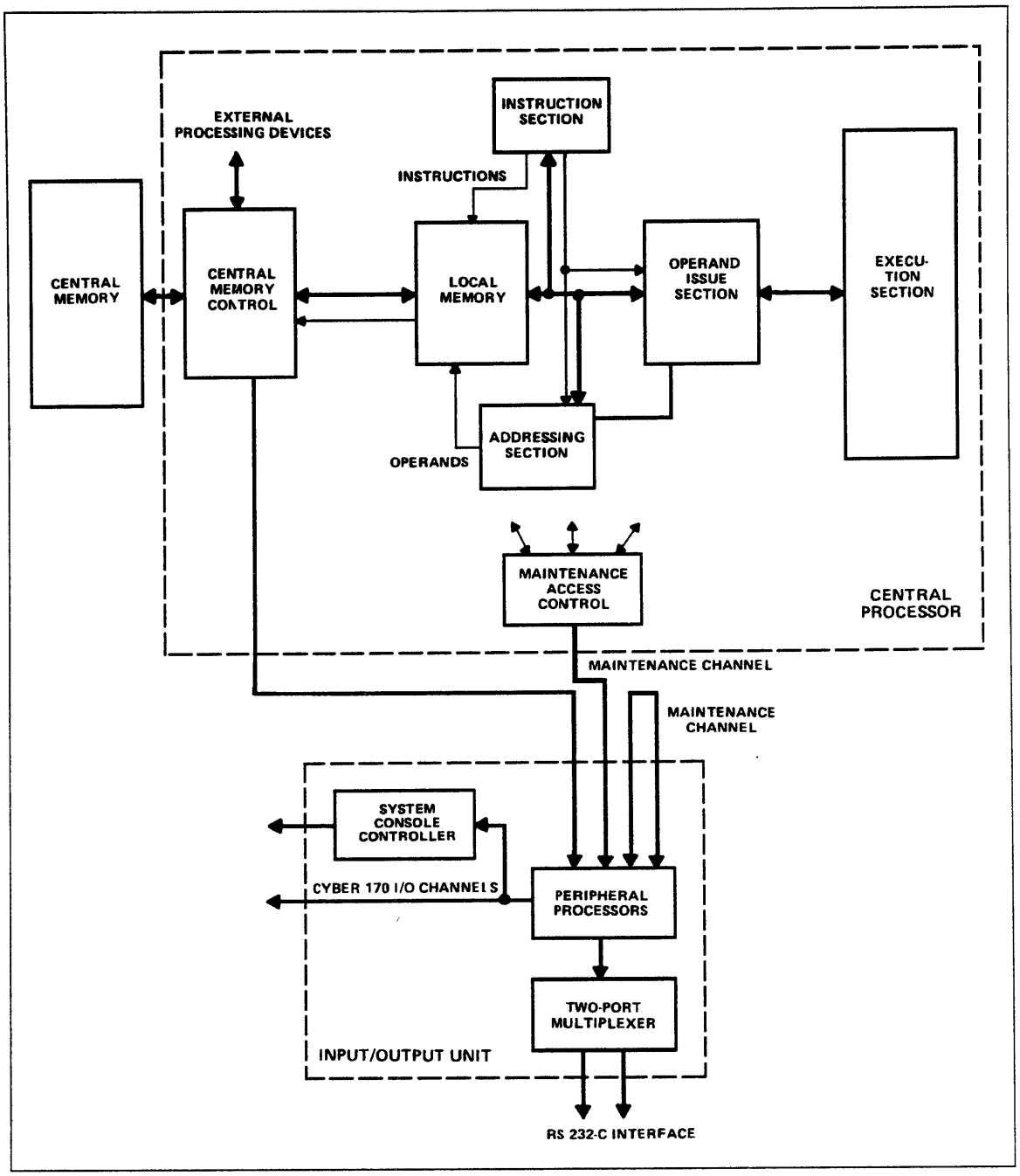


Figure 7-3. Computer System Block Diagram

Models 840, 850, and 860 Functional Descriptions 8

Central Processor 8-1
Instruction Section 8-1
Models 840 and 850 Instruction Lookahead 8-1
Model 860 Instruction Lookahead 8-1
Maintenance Access Control 8-1
Instruction Control Sequences 8-2
Boolean Sequence 8-2
Shift Sequence 8-3
Floating-Add Sequence 8-4
Floating-Multiply and Floating-Divide Sequence 8-4
Increment Sequence 8-4
Compare/Move Sequence 8-6
CYBER 170 Exchange Sequence 8-7
Block Copy Sequence 8-7
Direct Read/Write Sequence 8-7
Normal Jump Sequence 8-8
Return Jump Sequence 8-8
Registers 8-9
Operating Registers 8-10
X Registers 8-10
A Registers 8-10
B Registers 8-10
Support Registers 8-11
P Register 8-11
RAC Register 8-11
FLC Register 8-11
EM Register 8-12
Flag Register 8-12
RAE Register 8-13
FLE Register 8-13
MA Register 8-13
Execution Section 8-14
Cache Memory 8-14
Addressing Section 8-14
Central Memory Control 8-14

Central Memory 8-15
Address Format 8-15
CM Access and Cycle Times 8-16
CM Ports and Priorities 8-16
SECDED 8-17
CM Layout 8-22
CM Bounds Register 8-22
Central Memory Reconfiguration 8-22

Input/Output Unit 8-23
Peripheral Processor 8-23
Deadstart 8-24
Barrel and Slot 8-24
PP Registers 8-26
R Register 8-26
A Register 8-26

P Register 8-27
Q Register 8-27
K Register 8-27
PP Numbering 8-28
PP Memory 8-28
I/O Channels 8-29
Real-Time Clock 8-30
Two-Port Multiplexer 8-30
Maintenance Channel 8-30
Central Memory Access 8-30

Models 840, 850, and 860 Functional Descriptions 8

This chapter provides functional descriptions of the central processor (CP), central memory (CM), and input/output unit (IOU) as shown in the system block diagram in chapter 7. Functional descriptions for the system display station and the cooling system are in their respective manuals listed in the system publication index in About This Manual.

Central Processor

The CP consists of the instruction section, registers, the execution section, cache memory, central memory control, and the addressing section.

Instruction Section

The instruction section consists of logic for instruction control.

Models 840 and 850 Instruction Lookahead

The models 840 and 850 instruction lookahead hardware (ILH) prefetches a maximum of 12 instructions to make the next instruction immediately available when the execution of the previous instruction is complete. This is accomplished by reading instructions from cache/CM into a series of buffer ranks.

Model 860 Instruction Lookahead

The model 860 instruction lookahead hardware (ILH) prefetches a maximum of 12 instructions to make the next instruction immediately available when the execution of the previous instruction is complete. This is accomplished by reading instructions from cache/CM into a series of buffer ranks.

When ILH detects a conditional branch, it assumes that the branch condition will be met. ILH computes the branch target address and reads instructions from cache/CM starting at the target address. If the branch is taken, the buffer ranks contain the next executable instruction words. If the branch is not taken, the hardware purges the buffer ranks and resumes prefetching at the instruction word following the unsatisfied branch instruction.

Maintenance Access Control

The maintenance access control performs initialization and maintenance operations in the CP.

Instruction Control Sequences

The instruction control section performs instruction translation and control sequences. Each control sequence obtains the necessary instruction operands from the operating registers and provides the control signals for execution. Instructions read from CM are 60-bit instruction words that are in four 15-bit and 30-bit groups. The 15-bit groups are termed parcels with the first parcel (parcel 0) being the highest-order 15 bits of a 60-bit CM word. Second, third, and fourth parcels (parcels 1, 2, and 3) follow in order. The 30-bit groups contain two 15-bit parcels.

The instruction control sequences control the execution of one or more instructions of a common type. These sequences and associated instructions are briefly described in this section. For further information, refer to CP Instruction Descriptions in chapter 16.

Boolean Sequence

The Boolean sequence controls instructions that require bit-by-bit data manipulation. This includes both the logical and transmissive operations. The instructions requiring logical operations are:

11	Logical product (Xj) and (Xk) to Xi	BXi Xj * Xk
12	Logical sum of (Xj) and (Xk) to Xi	BXi Xj + Xk
13	Logical difference of (Xj) and (Xk) to Xi	BXi Xj - Xk
15	Logical product of (Xj) with complement of (Xk) to Xi	BXi -Xk * Xj
	Logical sum of (Xj) with complement of (Xk) to Xi	BXi -Xk + Xj
17	Logical difference of (Xj) with complement of (Xk) to Xi	BXi -Xk - Xj

The instructions requiring transmissive operations are:

10	Transmit (Xj) to Xi	BXi Xj
11	Transmit complement of (Xk) to Xi	BXi -Xk

Shift Sequence

The shift sequence controls instructions that require shifting the 60-bit field of data within the operand word. The shift instructions are:

20	Left shift (Xi) by jk	LXi jk
21	Right shift (Xi) by jk	AXi jk
22	Left shift (Xk) nominally (Bj) places to Xi	LXi Bj, Xk
23	Right shift (Xk) nominally (Bj) places to Xi	AXi Bj, Xk
43	Form mask of jk bits to Xi	MXi jk

The shift sequence also controls the pack and unpack instructions. In the packed floating format, the coefficient is contained in the lower 48 bits. The sign and biased exponents are contained in the upper 12 bits. The unpack instruction obtains the packed word from the Xk register, delivers the coefficient to the Xi register, and delivers the exponent to the Bj register. The unpack and pack instructions are:

26	Unpack (Xk) to Xi and Bj	UXi Bj, Xk
27	Pack (Xk) and (Bj) to Xi	PXi Bj, Xk

The shift sequence also controls the normalize operations. The coefficient portion of the operand is repositioned, and the exponent is adjusted so that the most significant bit of the coefficient is in the highest-order bit position of the coefficient, and the exponent is decreased by the number of bit positions shifted. The normalize instructions are:

24	Normalize (Xk) to Xi and Bj	NXi Bj, Xk
25	Round normalize (Xk) to Xi and Bj	ZXi Bj, Xk

Floating-Add Sequence

The floating-add sequence controls the operations necessary to form the 48-bit floating sum with a 12-bit exponent of the floating-point sum or difference of two floating-point operands. The floating-add instructions are:

30	Floating sum of (Xj) and (Xk) to Xi	FXi Xj + Xk
31	Floating difference of (Xj) and (Xk) to Xi	FXi Xj - Xk
32	Floating double-precision sum of (Xj) and (Xk) to Xi	DXi Xj + Xk
33	Floating double-precision difference of (Xj) and (Xk) to Xi	DXi Xj - Xk
34	Round floating sum of (Xj) and (Xk) to Xi	RXi Xj + Xk
35	Round floating difference of (Xj) and (Xk) to Xi	RXi Xj - Xk

Floating-Multiply and Floating-Divide Sequence

The floating-multiply and floating-divide sequence controls the operation of floating-multiply, floating-divide, and population-count instructions.

The multiply instructions are:

40	Floating product of (Xj) and (Xk) to Xi	FXi Xj * Xk
41	Round floating product of (Xj) and (Xk) to Xi	RXi Xj * Xk
42	Floating double-precision product of (Xj) and (Xk) to Xi	DXi Xj * Xk

The divide instructions are:

44	Floating divide (Xj) by (Xk) to Xi	FXi Xj/Xk
45	Round floating divide (Xj) by (Xk) to Xi	RXi Xj/Xk

The population-count instruction counts the number of one bits in a 60-bit operand. The instruction is:

47	Population count of (Xk) to Xi	CXi Xk

Increment Sequence

The increment sequence controls the ones complement addition and subtraction of 18-bit fixed-point operands for increment instructions 50 through 77. The sequence also controls the 60-bit ones complement sum and difference values for long-add instructions 36 and 37.

The increment instructions are:

50	Set Ai to (Aj) + K	SAi Aj + K
51	Set Ai to (Bj) + K	SAi Bj + K
52	Set Ai to (Xj) + K	SAi Xj + K
53	Set Ai to (Xj) + (Bk)	SAi Xj + Bk
54	Set Ai to (Aj) + (Bk)	SAi Aj + Bk
55	Set Ai to (Aj) - (Bk)	SAi Aj - Bk
56	Set Ai to (Bj) + (Bk)	SAi Bj + Bk
57	Set Ai to (Bj) - (Bk)	SAi Bj - Bk
60	Set Bi to (Aj) + K	SBi Aj + K
61	Set Bi to (Bj) + K	SBi Bj + K
62	Set Bi to (Xj) + K	SBi Xj + K
63	Set Bi to (Xj) + (Bk)	SBi Xj + Bk
64	Set Bi to (Aj) + (Bk)	SBi Aj + Bk
65	Set Bi to (Aj) - (Bk)	SBi Aj - Bk
66	Set Bi to (Bj) + (Bk)	SBi Bj + Bk
67	Set Bi to (Bj) - (Bk)	SBi Bj - Bk
70	Set Xi to (Aj) + K	SXi Aj + K
71	Set Xi to (Bj) + K	SXi Bj + K
72	Set Xi to (Xj) + K	SXi Xj + K
73	Set Xi to (Xj) + (Bk)	SXi Xj + Bk
74	Set Xi to (Aj) + (Bk)	SXi Aj + Bk
75	Set Xi to (Aj) - (Bk)	SXi Aj - Bk
76	Set Xi to (Bj) + (Bk)	SXi Bj + Bk
77	Set Xi to (Bj) - (Bk)	SXi Bj - Bk

The long-add instructions are:

36	Integer sum of (Xj) and (Xk) to Xi	IXi Xj + Xk
37	Integer difference of (Xj) and (Xk) to Xi	IXi Xj - Xk

Compare/Move Sequence

The compare/move sequence controls data manipulation on a character basis. The compare/move instructions (also referred to as CMU instructions) are 60-bit instructions that use six support registers for source and result field CM addresses and character position offsets. The support registers load from the 60-bit instruction word. The compare/move instructions are:

464	Move indirect (Bj) + K	IM Bj + K
465	Move direct	DM
466	Compare collated	CC
467	Compare uncollated	CU

The support registers are:

- An 18-bit K1 register that specifies which relative CM address word contains the first character of the source data field.
- An 18-bit K2 register that specifies which relative CM address word contains the first character of the result field.
- A 4-bit C1 register that specifies the character position or offset of the first CM word of the source field.
- A 4-bit C2 register that specifies the character position or offset of the first CM word of the result field.
- Two 16-bit L registers (LA and LC) that specify the number of characters in the data field. The LA register is associated with K1, and the LC register is associated with K2. Instruction 464 uses 14 register bits. Instructions 465, 466, and 467 use only the lower eight register bits.

NOTE

CMU instructions are provided for compatibility with previous systems. For better performance, recompile jobs to avoid use of CMU instructions.

CYBER 170 Exchange Sequence

A CYBER 170 exchange sequence is the method used to swap jobs in and out of execution. When a CYBER 170 exchange jump instruction occurs, the CYBER 170 exchange sequence writes the contents of the current job's CP registers (described later in this chapter) into an area of central memory called a CYBER 170 exchange package. A CYBER 170 exchange package is associated with each job. It contains sufficient information to restart a job if the job is interrupted during execution and swapped out by a CYBER 170 exchange jump. To complete the sequence, CP registers for another job are read from its CYBER 170 exchange package and that job begins or resumes execution. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

Block Copy Sequence

The block copy sequence controls the transfer of data between CM and UEM. The number of words to be transferred is determined by the addition of K to the contents of Bj. The starting address for CM is formed by adding either the A0 register or certain bits of the X0 register to the RAC reference address. The starting address for UEM is formed by adding certain bits of the X0 register to the RAE reference address. The block copy instructions are:

011	Block copy Bj + K words from UEM to CM	RE Bj + K
012	Block copy Bj + K words from CM to UEM	WE Bj + K

Direct Read/Write Sequence

Instructions 014 and 015 perform single-word, direct read and write operations for UEM; and instructions 660 and 670 perform single-word, direct read and write operations for central memory.

014	Read one word from UEM at (Xk + RAE) into Xj	RXj Xk
015	Write one word from Xj to UEM at (Xk + RAE)	WXj Xk
660	Read central memory at (Xk) to Xj	CRXj Xk
670	Write Xj into central memory at (Xk)	CWXj Xk

Normal Jump Sequence

The normal jump sequence controls the execution of branch instructions 02 through 07. The 02 instruction performs an unconditional jump to the Bi register address plus K. The branch address is K with i equals zero. The 02 instruction is:

02	Jump to (Bi) + K	JP Bi + K

The conditional jump instructions 03 through 07 branch to address K if the jump condition is met.

These instructions are:

030	Branch to K if (Xj) = 0	ZR Xj, K
031	Branch to K if (Xj) = 0	NZ Xj, K
032	Branch to K if (Xj) is positive	PL Xj, K
033	Branch to K if (Xj) is negative	NG Xj, K
034	Branch to K if (Xj) is in range	IR Xj, K
035	Branch to K if (Xj) is out of range	OR Xj, K
036	Branch to K if (Xj) is definite	DF Xj, K
037	Branch to K if (Xj) is indefinite	ID Xj, K
04	Branch to K if (Bi) = (Bj)	EQ Bi, Bj, K
05	Branch to K if (Bi) $\neq$ (Bj)	NE Bi, Bj, K
06	Branch to K if (Bi) $\geq$ (Bj)	GE Bi, Bj, K
07	Branch to K if (Bi) < (Bj)	LT Bi, Bj, K

Return Jump Sequence

The return jump sequence controls the execution of three instructions.

00	Error exit to MA or program stop	PS
010	Return jump to K	RJ K
013	Central exchange jump to (Bj) + K or monitor exchange jump to MA	XJ Bj + K

Registers

The CP contains the operating and support registers described in the following paragraphs. These registers are located in the operand issue section (refer to figure 7-3).

The contents of these registers can be written into memory and reloaded from memory as a CYBER 170 exchange package by a single CP instruction (CYBER 170 exchange jump). Figure 8-1 shows the CYBER 170 exchange package.

The time a CYBER 170 exchange package resides in CP hardware is called an execution interval. During this interval, the contents of X, A, B, and P registers can be changed by CP instructions. The contents of other support registers change only as a result of a CYBER 170 exchange jump. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

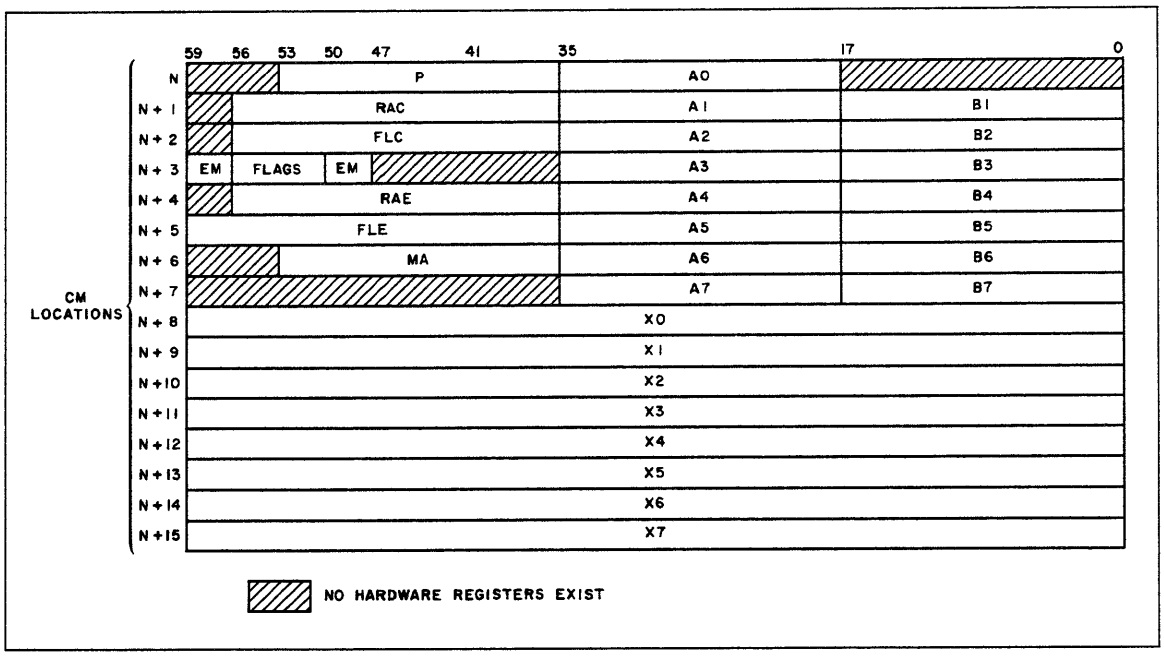


Figure 8-1. CYBER 170 Exchange Package

Operating Registers

The operating registers consist of operand (X), address (A), and index (B) registers. These registers minimize memory references for arithmetic operands and results.

X Registers

The CP contains eight 60-bit X registers, X0 through X7. The X0 register is used in the compare instructions to indicate if two fields of characters are equal. Also, the X0 register provides the relative UEM starting address in a block copy operation.

The X1 through X7 registers are primarily data handling registers for computation. X1 through X5 are used to input data from CM, and X6 and X7 are used to transmit data to CM.

Operands and results transfer between CM and the X registers as a result of placing CM addresses into corresponding A registers.

A Registers

The CP contains eight 18-bit A registers, A0 through A7. The A0 register serves as an intermediate register for the user's discretion. The A0 register is used in the compare collate instruction for the collate table address. Also, the A0 register provides the relative CM starting address in a block copy operation.

The A1 through A7 registers are essentially CM operand address registers associated one-for-one with the X registers. Placing a quantity into an address register (A1 through A5) causes a CM read reference to that address and transmits the CM word to the corresponding X register (X1 through X5). Similarly, placing a quantity into the A6 or A7 register causes the word in the corresponding X6 or X7 register to be written into that relative address of CM.

B Registers

The CP contains eight 18-bit B registers, B0 through B7. These registers are primarily indexing registers to control program execution. Program loop counts may also be incremented or decremented in these registers.

Program addresses may be modified on the way to an A register by adding or subtracting B register quantities. The B registers also hold shift counts for the nominal Bj shifts, the resultant exponent for the unpack, the operand exponent for the pack, and the resultant shift count from a normalize. The B0 register always contains positive zero which can be used as an operand. This register cannot hold results from instructions.

Support Registers

Eight support registers assist the operating registers during the execution of programs. The contents of the support registers are stored in CM, and their new contents are loaded from CM during a CYBER 170 exchange sequence. With the exception of the P register, the contents of the support registers cannot be altered during the execution interval of a CYBER 170 exchange package. When the execution interval completes, the data in the support registers is sent back to CM through a CYBER 170 exchange jump.

P Register

The 18-bit program address (P) register loads from CM during the first word of a CYBER 170 exchange sequence and contains the current program execution address. The register serves as a program address counter and holds the relative CM address for each program step.

RAC Register

The 21-bit CM reference address (RAC) register loads from CM during the second word of a CYBER 170 exchange sequence. An absolute CM address forms by adding RAC to a relative address determined by the instruction. The content of the P register is added to RAC to form the program address in CM. A P-equal-to-zero condition specifies relative address 0 and, therefore, (RAC). This CM location is reserved for recording error exit conditions and should not be used to store data or instructions.

FLC Register

The 21-bit CM field length (FLC) register loads from CM during the third word of a CYBER 170 exchange sequence. The FLC register defines the size of the field of the program in execution. Relative CM addresses are compared with FLC to check that the program is not going out of its allocated memory range.

EM Register

The 6-bit exit mode (EM) register loads from CM during the fourth word of a CYBER 170 exchange sequence. The EM register holds six exit-mode selection bits that control individual error conditions for a program. Selected EM register bits cause the CP to error exit when the corresponding conditions occur. Any or all of the six bits can be set at one time. Clear EM register bits allow the CP to continue, without error processing, when most of the corresponding conditions occur. Refer to the error exit tables under Error Response in chapter 17 for specific cases. The exit-mode selection bits appear in the exchange package as bits 48 through 50 and 57 through 59. The bits and their corresponding conditions are:

Mode Selection Bit	Significance
48	Address out of range
49	Infinite operand
50	Indefinite operand
57	Hardware error
58	Hardware error
59	Hardware error

Flag Register

The 6-bit flag register loads from CM during the fourth word of a CYBER 170 exchange sequence. The flag register holds six bits that function as control flags.

Bit	Condition
51	Hardware error bit.
52	Instruction stack (lookahead) purge flag. It set, extended purging of instruction lookahead registers is enabled. For further information, refer to Instruction Lookahead Purge Control under CP Programming in chapter 17.
53	CMU interrupted flag. If set, one of instructions 464 through 467 has been interrupted. The information necessary to resume operation has been saved.
54	Block copy flag. If set, block copy instructions (011, 012) use bits 30 through 50 of X0 rather than A0 to determine the CM address. For further information, refer to the descriptions of the block copy instructions in chapter 16.
55	Expanded addressing select flag. If set, UEM is operating in expanded addressing mode; if clear, UEM is operating in 24-bit standard addressing mode. For further information, refer to Addressing Modes under Memory Programming in chapter 17.
56	UEM enable flag. If set, UEM is available. This flag must be set to allow 011, 012, 014, and 015 instructions to access UEM.

RAE Register

The 21-bit UEM reference address (RAE) register loads from CM during the fifth word of a CYBER 170 exchange sequence. The lower six bits of this register are always zero. An absolute UEM address forms by adding RAE to the relative address which is determined by the instruction.

FLE Register

The 24-bit UEM field length (FLE) register loads from CM during the sixth word of a CYBER 170 exchange sequence. The lower six bits of this register are always zero. The FLE register defines the size of the field in UEM for the program in execution. Relative UEM addresses are compared with FLE.

MA Register

The 18-bit monitor address (MA) register loads from CM during the seventh word of a CYBER 170 exchange sequence. The MA register contains the absolute starting address of an exchange package which is used when executing a central exchange jump (013) instruction with the CYBER 170 monitor flag clear, or when honoring a monitor exchange jump to MA (262x) instruction with the CYBER 170 monitor flag clear. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

Execution Section

The execution section combines the operands into results, providing additional sequencing control where necessary.

Cache Memory

Cache memory is a high-speed buffer memory which is transparent to the user. It reduces effective CM access time by eliminating unnecessary CM references. When the CP first reads CM, a block of four words from CM (containing the rquested word) is read rapidly into cache memory. These words may be instructions or data. On subsequent reading of any of these words, CM need not be accessed when these words are in cache memory. Often this is the case because the same data is read more than once, or because a loop of instructions is repeatedly executed. Cache memory is 2048 words or, optionally, 4096 words.

Addressing Section

An address adder calculates memory addresses for data and unconditional jump instructions.

Memory management hardware verifies that memory addresses are to access permitted memory areas. If this is the case, this hardware accesses cache memory and, if necessary, central memory.

Central Memory Control

Central memory control (CMC) provides an interface to CM for the CP and IOU. It is physically located in the CP cabinet. CMC includes:

- Ports and distributor
- SECDED logic
- Partial-write logic
- Memory control logic
- Maintenance registers

Central Memory

The CM performs the following functions.

- Eight memory banks store from 2097K to 16 776K of 64-bit words (the leftmost 4 bits are undefined) and an 8-bit SECDED code.
- The two ports make CM accessible to the CP and every PP.
- A bounds register limits access to CM from either or both ports.
- The SECDED generators generate the SECDED code bits store with each word. SECDED checks circuits, corrects single-bit errors, and detects double-bit errors.
- The maintenance channel interface gives a PP in the IOU access to the CM maintenance registers for system initialization, corrective action, error reporting and diagnostics, and for setting the port bounds register.

Address Format

Figure 8-2 illustrates the address format.

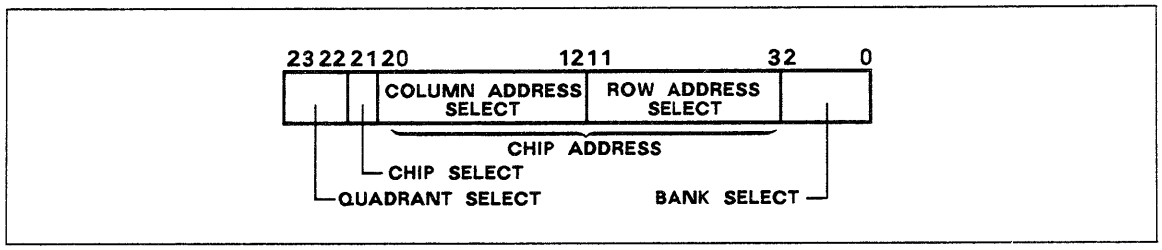


Figure 8-2. Address Format

The following list defines the address fields for figure 8-2.

- Quadrant Select specifies one of four quadrants (array packs) within a bank.
- Chip Select, if set, enables the row address select to the upper half (720) of the 144 chips on memory boards in all eight memory banks. If clear, enables the lower half of the 144 chips on memory boards in all eight banks.
- Chip Address, which comprises column address select and row address select, specifies the address of one word on a chip for the selected bank and quadrant.
- Row Address Select specifies the row-select portion of the chip address on a chip.
- Column Address Select specifies the column-select portion of the chip address on a chip.
- Bank Select specifies one of eight banks.

CM Access and Cycle Times

The CM access and cycle times operate on an internal clock period of 32 nanoseconds (minor cycle). The CM access time for a read operation is 320 nanoseconds (10 minor clock periods or 5 major cycles).

One bank cycle for a read or write operation is 384 nanoseconds (12 minor clock periods or 6 major cycles). Cycle time for a partial write (read/modify/write) is 768 nanoseconds (24 minor clock periods or 12 major cycles).

CM Ports and Priorities

A priority network resolves access conflicts on a rotating basis, preventing long-term lockout of any port. In case of simultaneous requests, the CP has priority. The CM also has a refresh mechanism which may consume a maximum of 4 percent of memory time. Refresh requests have priority over port requests. Refer to table 8-1 for maximum request lockout time in bank cycles.

Table 8-1. Port Priority

Port	Read or Write Requests
Refresh	1
Port 0	4
Port 1	5

Note:

One bank cycle equals 5 clock periods which equals 384 nanoseconds.

SECDED

The SECDED logic corrects single-bit errors during a CM read, permitting unimpeded computer operation. The SECDED logic prepares for the error correction by generating error correction code (ECC) bits for each data word, and by storing these ECC bits in CM with the data word during the CM write.

Table 8-2 lists the hexadecimal codes for all combinations of syndrome bits with the number of the data bit assigned each code or a note categorizing the code. Then, during a CM read, CM performs the following SECDED sequence.

1. Read one CM word and generate new ECC bits for data portion of CM word.
2. Compare new ECC bits with CM word ECC bits.
3. If old and new ECC bits match, no error exists. Send data to requesting unit.
4. If bits do not match, generate syndrome bits from result of ECC compare.
5. Decode syndrome bits to determine if single or multiple bit failure.
6. If single bit failure, correct by inverting failing bit in data word. Send corrected word to requesting unit.
7. If multiple bit or other uncorrectable error, send uncorrectable error response code to CP or IOU. A PP in the IOU may then analyze the syndrome bits using the maintenance channel.

Table 8-2. SECDED Syndrome Codes/Corrected Bits

Code	Bit	Code	Bit	Code	Bit	Code	Bit
00	7	10	67[2]	20	66[2]	30	2/3[6]
01	71[2]	11	3	21	3	31	4
02	70[2]	12	3	22	3	32	4
03	6/7[6]	13	4	23	4	33	3
04	69[2]	14	3	24	3	34	4
05	3	15	4	25	4	35	3
06	3	16	4	26	4	36	3
07	24[1]	17	24[5]	27	28[5]	37	28[1]
08	68[2]	18	3	28	3	38	4
09	3	19	4	29	4	39	3
0A	3	1A	4	2A	4	3A	3
0B	16[1]	1B	16[5]	2B	20[5]	3B	20[1]
0C	4/5[6]	1C	4	2C	4	3C	3
0D	8[1]	1D	8[5]	2D	12[5]	3D	12[1]
0E	0[1]	1E	0[5]	2E	4[5]	3E	4[1]
0F	3	1F	4	2F	4	3F	3

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error with indicated bit(s) inverted.

6. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes.

7. No error detected.

(Continued)

Table 8-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
40	65[2]	50	3	60	3	70	56[1]
41	3	51	4	61	4	71	56[5]
42	3	52	4	62	4	72	60[5]
43	4	53	3	63	3	73	60[1]
44	3	54	4	64	4	74	58[5]
45	4	55	3	65	3	75	58[1]
46	4	56	3	66	3	76	62[1]
47	26[5]	57	26[1]	67	30[1]	77	30/62[5]
48	3	58	4	68	4	78	57[5]
49	4	59	3	69	3	79	57[1]
4A	4	5A	3	6A	3	7A	61[1]
4B	18[5]	5B	18[1]	6B	22[1]	7B	22/61[5]
4C	4	5C	3	6C	3	7C	59[1]
4D	10[5]	5D	10[1]	6D	14[1]	7D	14/59[5]
4E	2[5]	5E	2[1]	6E	6[1]	7E	6/63[5]
4F	4	5F	3	6F	3	7F	63[1]

1. Corrected single-bit error.
2. Syndrome code bit failed (single code bit set).
3. Double error or multiple error (even number of code bits set).
4. Multiple error reported as a single error.
5. Double error or multiple error with indicated bit(s) inverted.

(Continued)

Table 8-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
80	64[2]	90	3	A0	3	B0	48[1]
81	3	91	4	A1	4	B1	48[5]
82	3	92	4	A2	4	B2	52[5]
83	4	93	3	A3	3	B3	52[1]
84	3	94	4	A4	4	B4	50[5]
85	4	95	3	A5	3	B5	50[1]
86	4	96	3	A6	3	B6	54[1]
87	25[5]	97	25[1]	A7	29[1]	B7	29/54[5]
88	3	98	4	A8	4	B8	49[5]
89	4	99	3	A9	3	B9	49[1]
8A	4	9A	3	AA	3	BA	53[1]
8B	17[5]	9B	17[1]	AB	21[1]	BB	21/53[4]
8C	4	9C	3	AC	3	BC	51[1]
8D	9[5]	9D	9[1]	AD	13[1]	BD	13/51[5]
8E	1[5]	9E	1[1]	AE	5[1]	BE	5/55[5]
8F	4	9F	3	AF	3	BF	55[1]

1. Corrected single-bit error.
2. Syndrome code bit failed (single code bit set).
3. Double error or multiple error (even number of code bits set).
4. Multiple error reported as a single error.
5. Double error or multiple error with indicated bit(s) inverted.

(Continued)

Table 8-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
C0	0/1[5]	D0	40[1]	E0	32[1]	F0	2
C1	3	D1	40[4]	E1	32[4]	F1	3
C2	3	D2	44[4]	E2	36[4]	F2	3
C3	2	D3	44[1]	E3	36[1]	F3	2
C4	3	D4	42[4]	E4	34[4]	F4	3
C5	2	D5	42[1]	E5	34[1]	F5	2
C6	2	D6	46[1]	E6	38[1]	F6	2
C7	27[1]	D7	27/46[4]	E7	31/38[4]	F7	2
C8	3	D8	41[4]	E8	33[4]	F8	3
C9	2	D9	41[1]	E9	33[1]	F9	2
CA	2	DA	45[1]	EA	37[1]	FA	2
CB	19[1]	DB	19/45[4]	EB	23/37[4]	FB	23[1]
CC	2	DC	43[1]	EC	35[1]	FC	2
CD	11[1]	DD	11/43[4]	ED	15/35[4]	FD	15[1]
CE	3[1]	DE	3/47[4]	EE	7/39[4]	FE	7[1]
CF	2	DF	47[1]	EF	39[1]	FF	2

1. Corrected single-bit error.

2. Double error or multiple error (even number of code bits set).

3. Multiple error reported as a single error.

4. Double error or multiple error with indicated bit(s) inverted.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes.

CM Layout

Central memory contains an area that is reserved for special software called Virtual State software. Along with the hardware and microcode, this software handles the operations of Virtual State as described in chapter 17. Virtual State software is located at the higher end of memory. The remaining memory is available to the CYBER 170 State and may be allocated as central memory (accessible via RAC and FLC) or as unified extended memory (accessible via RAE and FLE and the 011, 012, 014, and 015 instructions). Refer to figure 8-3.

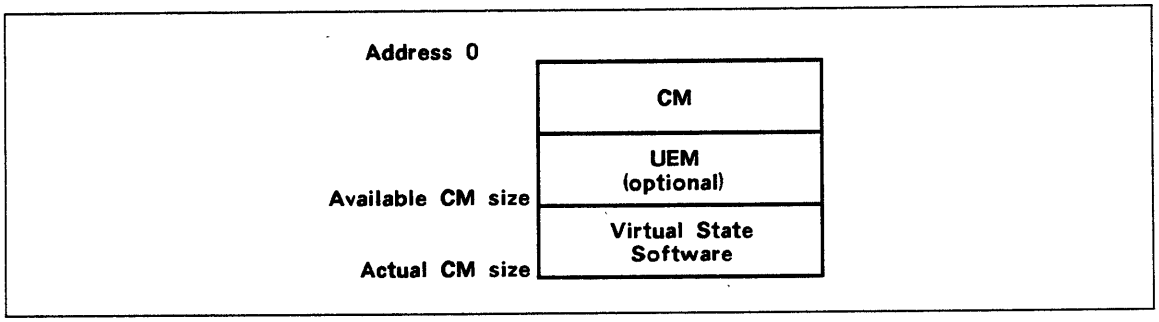


Figure 8-3. CM Layout

CM Bounds Register

The CM bounds register limits the write access to CM from specified ports. The ports are limited to the area between an upper and lower bound as specified in the CM bounds register. Bits in byte 0 specify the port(s) from which the write access is limited. The CM bounds register is set through the maintenance channel. For further information, refer to Maintenance Channel Programming in chapter 17.

Central Memory Reconfiguration

Central memory reconfiguration is a manually performed function that permits the computer operator to restructure the CM addresses so that a failing part of CM can be quickly locked out to provide a continuous block of usable CM. CM reconfiguration is accomplished by setting the switches on the memory unit to manipulate the upper address bits.

When a configuration switch is set forcing a CM address bit to a zero/one, the address range corresponding to the original installed memory accesses some parts of the reconfigured memory more than once. Addresses up to the rightmost forced bit, and half the addresses using the rightmost forced bit, cover a contiguous address space from location 0, which is the reconfigured memory. For further information, refer to chapter 9.

Input/Output Unit

The input/output unit (IOU) performs the functions required to locate, select, and initialize the external devices connected to the system, and controls the transfer of data between a selected device and CM. The IOU also performs system maintenance functions.

The IOU contains the following functional areas.

- Peripheral processor (PP)
- I/O channels
- Real-time clock
- Two-port multiplexer
- Maintenance channel
- CM access

Peripheral Processor

The basic IOU contains 10 PPs and can be expanded to 20 PPs in 5-PP increments. Each PP is a logically independent computer with its own memory. Each 5-PP group is organized into a multiplexing system which allows the PPs to share common hardware for arithmetic, logical, and I/O operations without losing independence. This multiplexing system comprises five ranks of registers termed a barrel. Each rank contains information related to the instruction being executed by one PP.

Each PP can communicate with the CP by issuing a CYBER 170 exchange request to a specific CYBER 170 exchange package associated with the issuing PP. In addition, a PP can also communicate with the CP via CM read and write operations. PPs can communicate with each other over the I/O channels and through CM.

Each PP executes programs alone or with other PPs to control data transfers between external devices and CM. These programs are comprised of instructions from the IOU instruction set and respond to requests issued through CM by the operating system. The programs translate generalized operating system requests into control functions for accessing the external devices and may also perform device scheduling and optimization. The programs use PP memory as a buffer for the data transfer between external devices and CM to isolate IOU data transfer from variations in CM transfer rate.

An IOU upgrade is available which is an optional concurrent input/output (CIO) subsystem consisting of five or ten PPs. Optional intelligent standard interface (ISI), intelligent peripheral interface (IPI), and CYBER 170 DMA (direct memory access) I/O channel adapters can be installed in the CIO.

Deadstart

A deadstart sequence allows the IOU to initialize itself. This deadstart sequence is initiated by the DEAD START switch on the deadstart panel. The panel includes controls for assigning any PPM to PP0. For further information, refer to chapter 9.

Barrel and Slot

The barrel consists of the R, A, P, Q, and K registers, each one of which has five ranks 0 through 4. (Refer to figure 8-4.) Information in these registers moves from one rank to the next at a uniform 20-megahertz rate, providing a multiplexed system of five PPs, each operating at a 4-megahertz rate. The registers are stationary while the PPs rotate. For example, rank 4 registers contain PP0, PP1, PP2, PP3, and PP4 in succession, each consuming 50 nanoseconds of the total cycle time of 250 nanoseconds. Since PP memories operate at a slower rate, independent memory with its own address and data registers is provided for each PP.

Each time data enters the slot, a portion of the instruction for that data is executed. The slot performs tasks such as arithmetic and logic operations and program address manipulation. Complete execution of an instruction may require the R, A, P, Q, and K register quantities to go more than one trip around the barrel and through the slot.

The PPM may be referenced once each time the PP passes around the barrel and through the slot. During its slot time, the PP may also communicate with CM or any of the I/O channels.

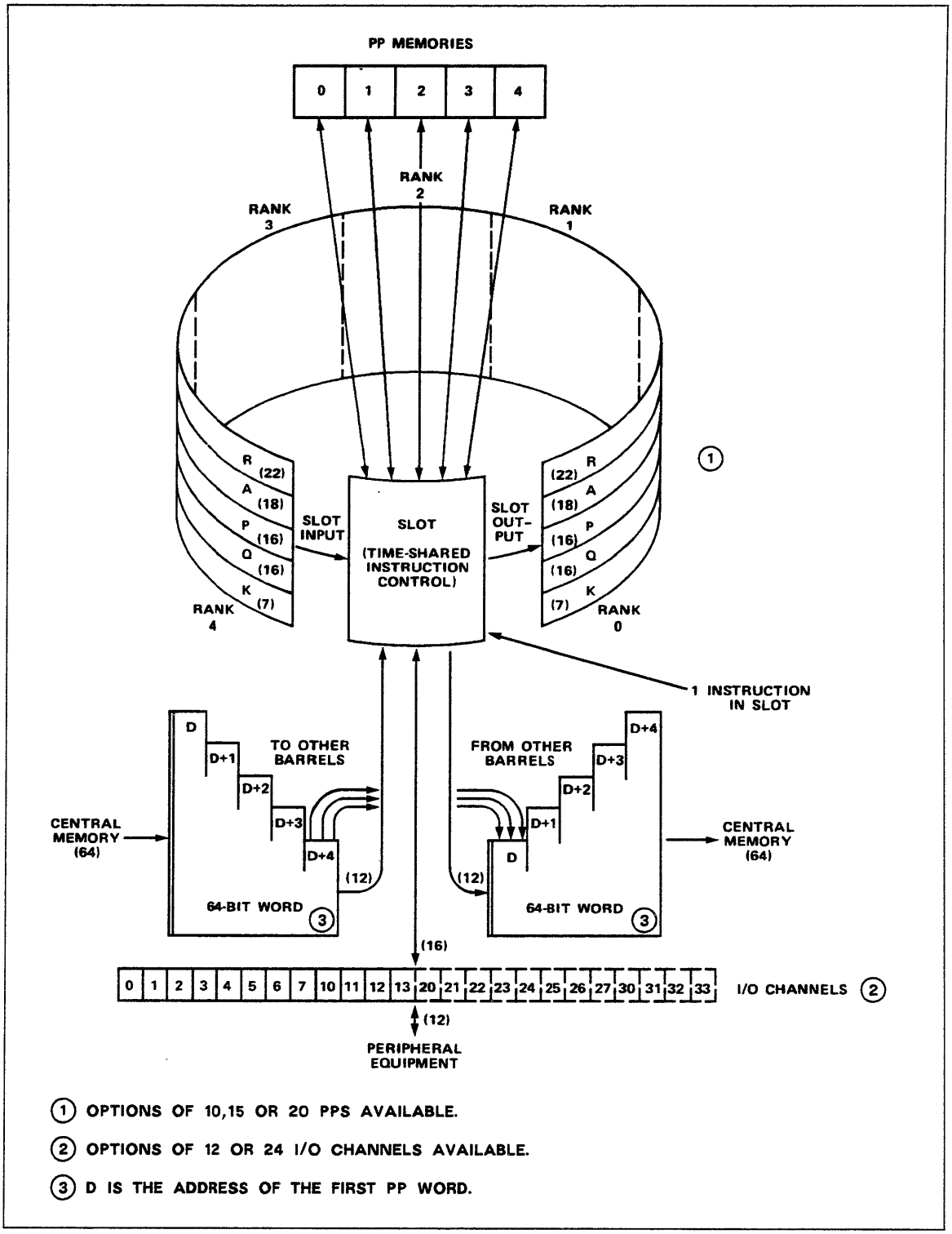


Figure 8-4. Barrel and Slot

PP Registers

The PP registers, which are discussed in the following paragraphs, are:

- R register
- A register
- P register
- Q register
- K register

R Register

The 28-bit R register, in conjunction with the A register, forms an absolute CM address for CM read/write instructions. When bit 17 of the A register is set, the absolute CM address is formed by appending six zeros to the lower end of the contents of the R register and adding to the result bits 0 through 16 of the contents of the A register (figure 8-5).

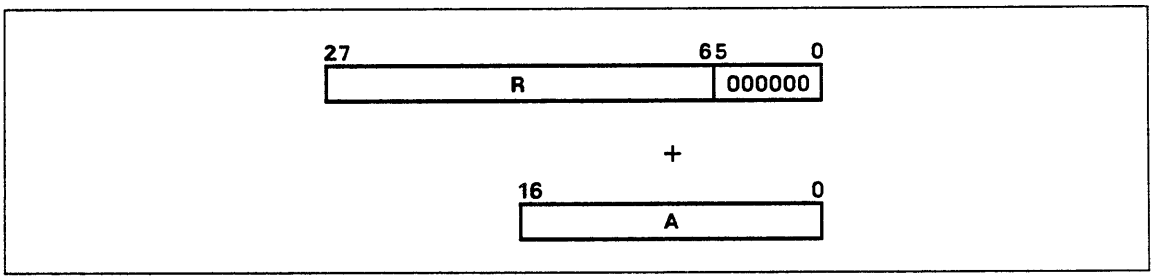


Figure 8-5. Formation of Absolute CM Address

A Register

The 18-bit A register holds one operand for arithmetic, logic, or selected I/O operations. The content of A may be an arithmetic operand, CM address, I/O function, or I/O data word. Various instructions operate on 6, 12, 16, or 18 bits of the A register.

When the A register is used as the CM address, parity is generated for transmission with the address to memory control. At deadstart, the A register is set to 10000_8. When bit 17 of the A register is clear, the A register is used as the CM address; however, when bit 17 is set, the R register is added to the A register (as described in the R register description) to obtain the absolute CM address for CM read/write instructions.

P Register

The 12-bit P register is the program address register, except during the execution of instructions 61, 63, 71, and 73. For these instructions, the P register contains the PPM address of the data transfer. At deadstart, the P register is set to zero.

Q Register

The 12-bit Q register holds data for several functions such as the address of the operand during direct addressing and indirect addressing, peripheral address of data used during one-word central read or write instructions, upper 6 bits during constant mode instructions, channel number on all I/O and channel instructions, shift count, and relative jump designator. At deadstart, each rank of the Q register is set to a corresponding PP number. Rank 0 is set to PP0, rank 2 is set to PP2, and so on.

K Register

The 12-bit K register is visible to the programmer through the maintenance channel only. This register holds the operation code field of an instruction for display on the IOU deadstart panel and for deadstart panel interrogation. When a PP is halted (idled), this register contains all ones.

PP Numbering

PPs are numbered in octal as follows:

Barrel	PPs
0	00 to 04
1	05 to 11
2	20 to 24
3	25 to 31

The deadstart sequence is used to determine PP numbering within a barrel. The sequence assigns barrel numbers according to the switch settings and, during the first minor cycle after deadstart, loads a zero into the Q register in barrel 0. This defines all the data in that rank of the barrel as belonging to PP0 and since Q is the channel selector, assigns PP0 to channel 0. During the next minor cycle, Q loads with a one. This defines PP1 and assigns it to channel 1. This process occurs in parallel in all barrels until the IOU assigns each rank of each barrel with a PP number and a channel number. Reassignment can be done only during a deadstart.

PP Memory

Each PP has an independent 4K word memory. Each word contains 16 data bits with the upper 4 bits set to zero, and 1 parity bit. PP0 executes the deadstart program from the deadstart panel during the deadstart operation. Therefore, PP memory 0 must be operational. A PP memory reconfiguration feature allows the user to restore IOU operation if the IOU detects a fault in the PP memory normally assigned to PP0.

To reconfigure, the operator assigns a good PP memory to PP0 and the operating system removes the failing PP memory. Computer operation can continue without the failing PP memory, and repairs can be made during scheduled maintenance. The system must be deadstarted to reconfigure PPMs.

I/O Channels

The I/O channels are comprised of an internal interface that allows common hardware and software to control the external devices, and an external interface that allows the IOU to communicate with the external devices using 12-bit data channels. The internal interface can transfer 16-bit data words between two PPs, or between a PP and an external device at a maximum rate of one word every 250 nanoseconds. This rate can be sustained for the maximum practical channel transfer (4096 words). During transfers between PPs, if the PPs are in the slot at the same time, the transfer rate is 500 nanoseconds.

Any PP can access any of the CYBER 170 bidirectional I/O channels. All PPs communicate with external devices through the independent I/O channels. Each channel may be connected to one or more pieces of external equipment, but only one piece of equipment can use a channel at one time. All channels can be active simultaneously. Available channels are listed as follows:

- Twenty-four CYBER 170 campatible I/O channels available with a maximum data transfer rate of 3 megabytes/second.
- An optional DMA-enchanced intelligent standard interface (ISI) channel adapter, intelligent peripheral interface (IPI) channel adapter, or CYBER 170 channel adapter that can be installed in any one of ten channel locations in the CIO cabinet. The adapters transfer data between the ISI, IPI, or CYBER 170 channel and PP memory using standard I/O instructions. They also support DMA transfer in which data goes directly between CM and an external device without going through the PP. There are two types of CYBER 170 DMA transfers, fast and normal. Fast transfers are used with the Extended Semiconductor Memory-II (ESM-II), and normal transfers are used with other CYBER 170 external devices.

The display station controller (DSC) is attached to CYBER 170 channel 10_8. The DSC is the IOU interface between the PPs and the system console, servicing both the keyboard and the cathode-ray tube (CRT). It transmits function words and digital symbol size/position data to the system console and receives digital character codes from the keyboard. It also receives digital symbol codes from the PPs and converts these to analog signals to the CRT.

Real-Time Clock

The real-time clock is a 12-bit free-running counter, incrementing at a 1-megahertz rate. It is permanently attached to channel 14_8. This channel may be read at any time as its active and full flags are always set.

Two-Port Multiplexer

The two-port multiplexer provides communication capability between a PP and two attached terminals. One port is reserved for maintenance purposes and the other port is reserved for future use. The two-port multiplexer is permanently attached to channel 15_8.

Maintenance Channel

The maintenance channel is used for initialization of the CP and CM maintenance registers and monitoring of error status.

The maintenance channel consists of the maintenance channel interface on channel 17_8, a maintenance access control in each system element, and a set of interconnecting cables.

Central Memory Access

Any PP can access CM. During a write from the IOU to CM, the IOU assembles five successive 12-bit PP words into a 64-bit CM word with the leftmost 4 bits undefined. During a CM read, the IOU disassembles the rightmost 60 bits of the 64-bit CM word into five PP words. To find the CM address, a PP reads the A register. If bit 17 of the A register is clear, the PP uses the contents of the A register for the CM address. If bit 17 of the A register is set, the PP adds the relocation address from the R register to the A register to form the CM address.

A maximum of 20 PPs in various stages of assembly/disassembly can simultaneously read CM words, and five PPs can write CM words.

Models 840, 850, and 860 Operating Instructions 9

Controls and Indicators 9-1
Deadstart Panel Controls/Indicators 9-1
Central Memory Controls 9-6

Power-On and Power-Off Procedures 9-10

Operating Procedures 9-10
Control Checks 9-10
Deadstart Sequences 9-11

IOU Reconfiguration 9-12

Models 840, 850, and 860 Operating Instructions 9

This chapter describes mainframe controls and indicators and the operating procedures which are hardware dependent. Software-dependent procedures are in system software reference manuals listed under Additional Related Manuals in About This Manual.

Controls and Indicators

This section describes IOU deadstart controls and indicators and CM configuration switches used by the system operator. Other controls used by maintenance personnel are described in the hardware operator's guide and the hardware maintenance manuals of the power distribution and warning system, the cooling system, and the system console listed in the system publication index in About This Manual.

Deadstart Panel Controls/Indicators

The deadstart panel (figure 9-1) is in the IOU. It contains PP register selection and display facilities, deadstart controls, error indicators, and a switch matrix, which is the source for a short PP program for initialization or troubleshooting. The switches, indicators, and their functions are listed in table 9-1.

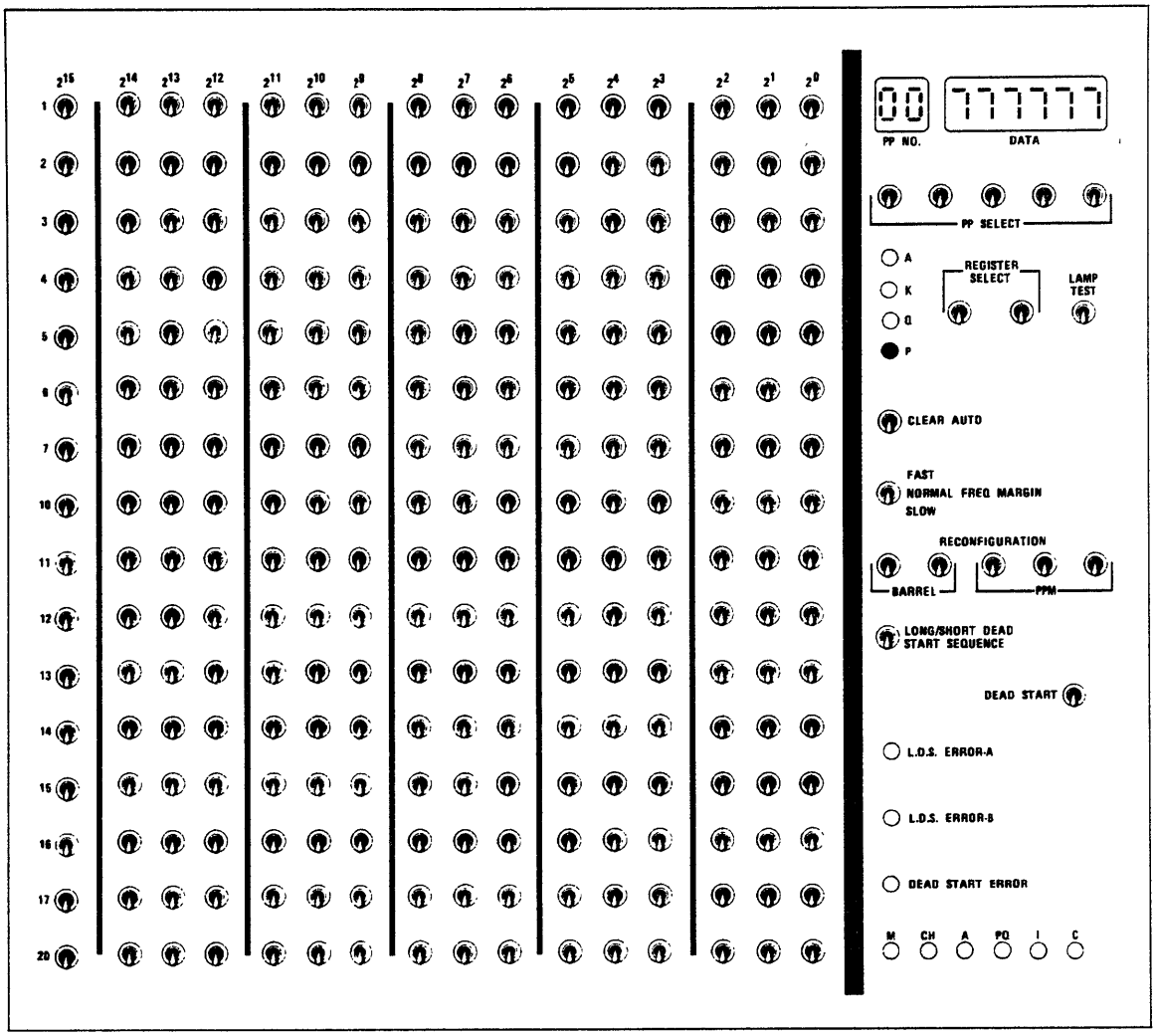


Figure 9-1. Deadstart Panel

Table 9-1. Deadstart Panel Controls/Indicators

Panel Nomenclature	Description	Function
20 through 215 by 1 through 208_8	Toggle switch matrix (two-position switches)	Provides a 16-word deadstart program for PP0. Switches 2^0 through 2^{11} set 12 bits for each of the program words, labeled 1 through 20 (octal). Switches 2^{12} through 2^{15} are set to zero. Up position sets bit. Down position clears bit.
PP NO	Octal display	Shows the PP selected by PP SELECT switches.
DATA	Octal display	Shows the content of the register selected by REGISTER SELECT switches.
PP SELECT	Toggle switches (two-position)	Selects the PP whose register is to be displayed.
REGISTER SELECT	Toggle switches (two-position)	Selects the register to be displayed (00 = P, 01 = Q, 10 = K, 11 = A).
A, K, Q, P	Indicators	One of these lights to indicate which register is selected by REGISTER SELECT switches.
LAMP TEST	Toggle switch (two-position)	Lights all indicators and display segments.
CLEAR AUTO	Toggle switch (two-position)	Allows manual clearing of auto-mode bit (bit 34 of the environment control register) to override possible auto-mode selection. This allows the selection of the PP and register from the deadstart panel if bit 34 is set.
FREQ MARGIN	Toggle switch (three-position)	Determines the frequency margin selected (FAST/NORMAL/SLOW). The setting of this switch is sensed only at deadstart time.

(Continued)

Table 9-1. Deadstart Panel Controls/Indicators *(Continued)*

Panel Nomenclature	Description	Function
RECONFIGURATION, BARREL	Toggle switches (two-position)	Selects the physical barrel which is logical barrel 0. All the other logical barrels are numbered from the selected physical barrel circularly. (If physical barrel 1 is selected by the switches, physical barrel 2 is logical barrel 1, and so on.)
RECONFIGURATION, PPM	Toggle switches (two-position)	Selects the physical PP memory which is logical PPM0. All the other PPMs in all barrels are numbered from the selected physical PPM circularly. If the switches are set to a value greater than four, no reconfiguration takes place.
LONG/SHORT DEAD START SEQUENCE	Toggle switch (two-position)	Selects the LONG/SHORT deadstart sequence. The setting of this switch is sensed only at deadstart.
DEAD START	Toggle switch (three-position, center is off)	Selects the fast or slow repetitive deadstart, which generates a master clear pulse every 250 or 4000 microseconds respectively. Up position selects fast deadstart; down position selects slow deadstart. (The single deadstart control pushbutton is on the system console.)
L.D.S. ERROR-A	Indicator	Remains lit when long deadstart branch tests are not completed within 10.25 microseconds.
L.D.S. ERROR-B	Indicator	Remains lit when a long deadstart sequence does not go to completion.
DEAD START ERROR	Indicator	Lights in case of long deadstart ROM address/data parity error.

(Continued)

Table 9-1. Deadstart Panel Controls/Indicators *(Continued)*

Panel Nomenclature	Description	Function
M, CH, A, PQ, I, C	Indicators	Lights in case of hardware failures as follows: M: PP memory failure CH: I/O channel failure A: A barrel failure PQ: P or Q barrel failure I: Firmware or control failure C: 12/16 conversion failure

Central Memory Controls

The CM contains six two-position configuration switches (figure 9-2). These switches are located along the address interface pak switch in the A section of the memory cabinet. The switches are used to eliminate CM sections with malfunctions.

Each switch, SW0 through SW5, inverts the corresponding CM address bit 37 through 42. The inversion effectively moves blocks of bad memory to the highest memory block and moves blocks of good memory down, thereby providing a sequentially addressable block of error-free memory. Refer to table 9-2.

In case of CM malfunctions, the remaining good memory can be reconfigured so it is accessible by contiguous addresses from 0 to the maximum remaining address. This is accomplished by setting configuration switches (figure 9-2) as listed in table 9-2. Refer to the hardware operator's guide listed in the system publication index for further information.

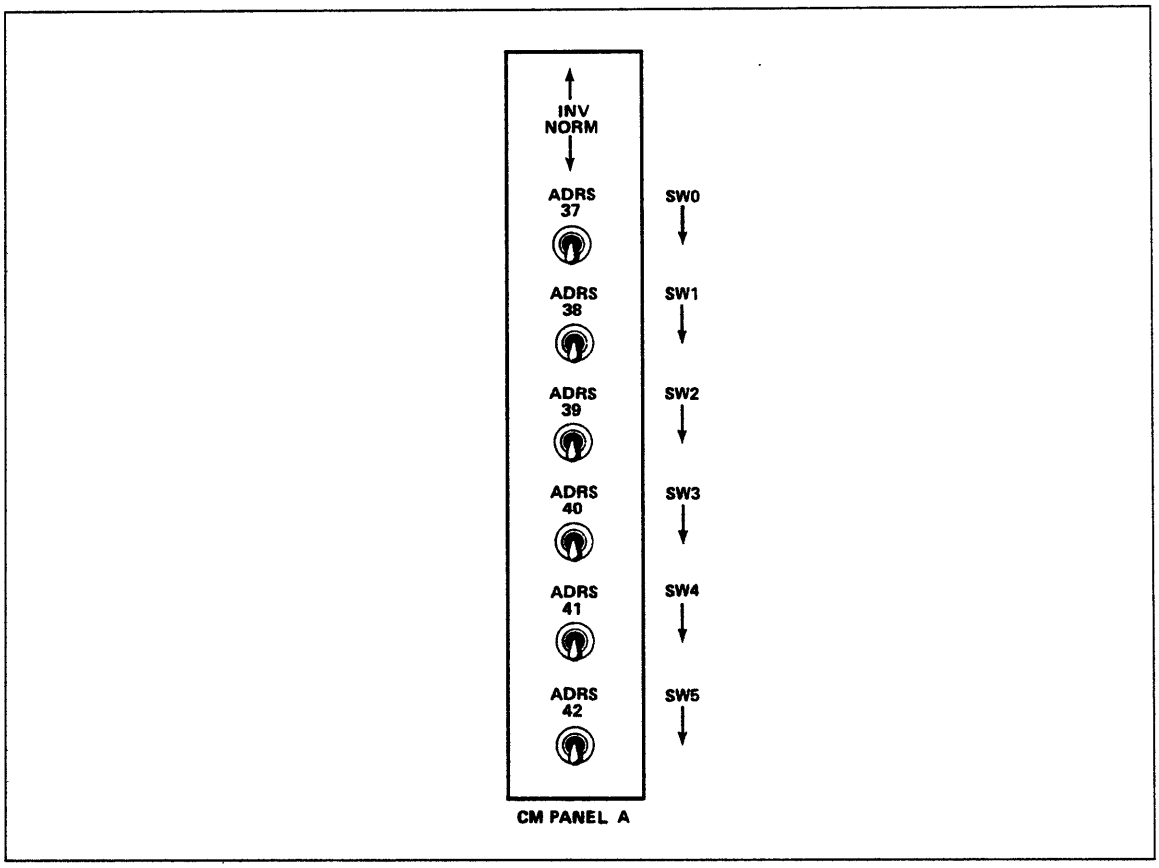


Figure 9-2. CM Configuration Switches

Table 9-2. Central Memory Reconfiguration

Original CM		Reconfigured CM (Reconfiguration Settings)			
Words (Size)	Address Range	Error-Free Size	SW0 ADRS 37	SW1 ADRS 38	SW2 ADRS 39
2097K (16 MB)	0-7 777 777	1049K (8 MB)	D	D	D
4195K (32 MB)	0-17 777 777	2097K (16 MB)	D	D	U
8390K (64 MB)	0-37 777 777	4195K (32 MB)	D	U	D
16780K (128 MB)	0-77 777 777	8390K (64 MB)	U	D	D

Notes:

1. CM remaining can be further reconfigured to obtain larger contiguous blocks of error-free memory by setting additional configuration switches. See examples shown in figure 9-3.

2. U equals up, D equals down. Normal setting of all switches is down.

(Continued)

Table 9-2. Central Memory Reconfiguration *(Continued)*

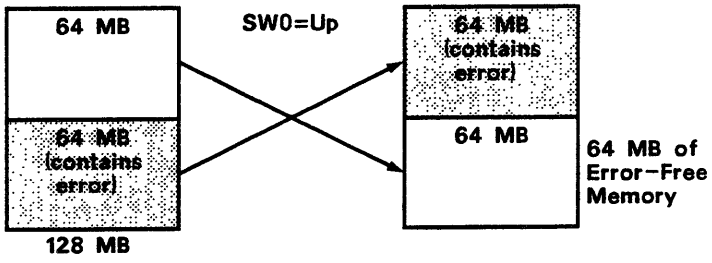
Original CM		Reconfigured CM (Reconfiguration Settings)			
Words (Size)	**Address Range**	**Error-Free Size**	**SW0 ADRS 40**	**SW1 ADRS 41**	**SW2 ADRS 42**
2097K (16 MB)	0-7 777 777	1049K (8 MB)	U	D	D
4195K (32 MB)	0-17 777 777	2097K (16 MB)	D	D	D
8390K (64 MB)	0-37 777 777	4195K (32 MB)	D	D	D
16780K (128 MB)	0-77 777 777	8390K (64 MB)	D	D	D

Notes:

1. CM remaining can be further reconfigured to obtain larger contiguous blocks of error-free memory by setting additional configuration switches. See examples shown in figure 9-3.

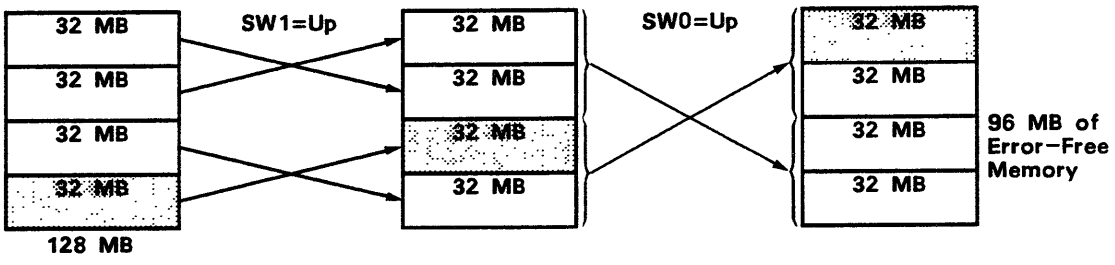
2. U equals up, D equals down. Normal setting of all switches is down.

Set SW0 up to move block of memory containing error to upper half of memory.



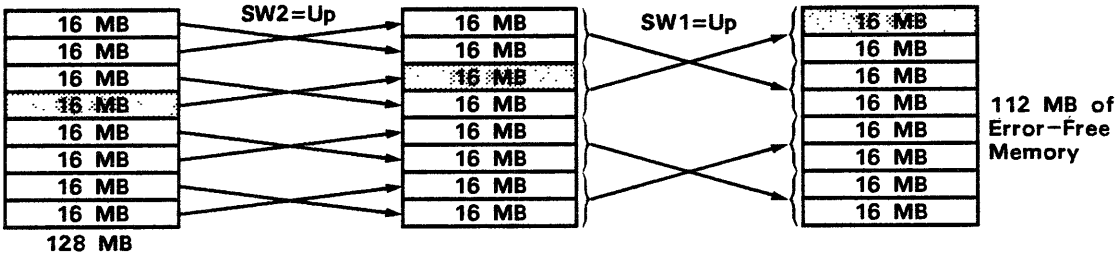
Error in lower 64-MB block of 128-MB memory.

Set SW1 up to move 32-MB block containing error to next higher 32 MB. Then set SW0 up to move block containing error to highest block of memory.



Error in lowest 32-MB block of 128-MB memory.

Set SW2 up to move 16-MB block containing error to next higher block. Then set SW1 up to move 32-MB block containing error to highest block of memory.



Error in lowest 16-MB block of 128-MB memory.

Figure 9-3. Reconfiguration Examples

Power-On and Power-Off Procedures

In case of an emergency, use the system EMERGENCY OFF switch. The power-on and power-off procedures are described in the hardware operator's guide listed in the system publication index.

CAUTION

Improper application or removal of power may damage system circuits and/or air conditioning system. Power must be turned on/off by designated personnel only, except for the system EMERGENCY OFF switch. Use only for extreme emergency, not for normal shutdown.

Operating Procedures

Refer to the hardware operator's guide listed in the system publication index. The system is initialized by setting its control switches, and then by running either a long or short deadstart sequence (defined later in this section). After initialization, the keyboard is used to instruct the system further, under program control.

Control Checks

Before activating a long or short deadstart sequence, check the positions of deadstart panel switches against their intended use. These checks can be made by using table 9-3. The normal settings of these switches is as follows:

Switch	Position
CLEAR AUTO	Down
FREQ MARGIN	Center
RECONFIGURATION	All down
LONG/SHORT DEAD START SEQUENCE	Down for a short deadstart sequence
DEAD START	Center
All error lights	Not lit

Deadstart Sequences

In response to a deadstart signal from either the deadstart pushbutton on the system console, or from the DEAD START switch on the deadstart panel, circuits in the IOU perform a deadstart sequence. Depending on the setting of the LONG/SHORT DEAD START SEQUENCE switch on the deadstart panel, either the long or the short deadstart sequence is performed. The short deadstart sequence is used when hardware integrity verification is not required. The long deadstart sequence performs all the tasks performed by the short deadstart sequence and some additional tasks. The main additional task is the running of a diagnostic program, from a read-only memory (ROM) in the IOU, on logical PP0. The diagnostic program takes approximately 1 minute to run.

Both deadstart sequences begin with a master clear which sets up all PPs, except logical PP0, for a 4096-word block input starting at PP location 0. The input into each PP is from the channel with the same number as the logical number of the PP concerned. The master clear also resets all external devices and sets maintenance channel connect code bit 52. The individual channels and registers are set as follows:

Channel	Active/ Inactive Flag	Full/ Empty Flag	Channel Flag	Channel Error Flag
0	Inactive	Empty	Clear	Clear
10 (display controller)	Active	Empty	Clear	Clear
14 (real-time clock)	Active	Full	Set	Set
15 (two-port mux)	Active	Empty	Clear	Clear
17 (maintenance)	Active	Empty	Clear	Clear
Other installed channels	Active	Empty	Clear	Clear
Noninstalled channels	Inactive	Empty	Clear	Clear

The flags of channel 14 and of noninstalled channels are fixed by hardware and cannot be changed.

Register	Initialization[1]	Description
K	007100_8	Instruction display on deadstart panel
P	007777_8	Causes block input to start from location 0
A	$10{,}000_8$	Count of 4096 words
Q	0, 1, 2...	I/O channel numbers (PP0: 0, PP1: 1, and so on)

All registers in all barrels are set to these values, except the registers of PP0.

1. Leading zeros are not displayed on deadstart panel.

If the long deadstart sequence is being performed, hardware clears location 7777_8 in all PP memories and sets the P register of PP0 to 6000_8. PP0 starts performing a test program from a read-only memory in IOU and lights the deadstart panel L.D.S. ERROR-A and L.D.S. ERROR-B indicators. Indicator A remains lit unless the test program reaches location 6200_8 within 10.25 microseconds. Indicator B remains lit until the test program reaches location 7776_8. When this happens, the unique part of the long deadstart sequence ends with a master clear.

Next, both deadstart sequences clear PP0 location 0, write the settings of the deadstart panel matrix switches into PP0 memory locations 1_8 to 20_8, and clear PP0 location 21_8. PP0 then starts executing the program entered from the matrix switches, which is normally a bootstrap program to input more data from an assigned external device.

The short deadstart sequence does not disturb PP memory other than PP0 locations 0 to 21_8. Both deadstart sequences leave all PPs, except PP0, waiting for a block input, or for action through the maintenance channel. After the block input is complete, each PP starts executing the program entered from whatever address was entered into location 0 of that PP.

IOU Reconfiguration

The logical PP numbers and hardware are assigned to physical PPs circularly from the settings of IOU deadstart panel RECONFIGURATION switches, which specify which physical barrel and PPM is PP0. If the PPM section of these switches is set to a value greater than 4, the value 0 is substituted. If the BARREL section of these switches is set to a value greater than the number of installed barrels, the value 0 is substituted. Thus, possible barrel numbering is as described in table 9-3.

NOTE

The minimum system option is 10 PPs.

Table 9-3. Barrel Numbering Table

Barrels Installed	Physical Barrel	Logical PPs in Physical Barrel with BARREL RECONFIGURATION Switch Values			
		0	**1**	**2**	**3**
4 Barrels (20 PPs)	0	0-4	25-31	20-24	5-11
	1	5-11	0-4	25-31	20-24
	2	20-24	5-11	0-4	25-31
	3	25-31	20-24	5-11	0-4
3 Barrels (15 PPs)	0	0-4	20-24	5-11	(0-4)
	1	5-11	0-4	20-24	(5-11)
	2	20-24	5-11	0-4	(20-24)
2 Barrels (10 PPs)	0	0-4	5-11	(0-4)	(0-4)
	1	5-11	0-4	(5-11)	(5-11)
1 Barrel (5 PPs)	0	0-4	(0-4)	(0-4)	(0-4)

Model 990 System Description 10

Physical Characteristics 10-1

Functional Characteristics 10-3
Central Processor 10-4
Central Memory 10-5
Input/Output Unit 10-6

Major System Component Descriptions 10-7
Central Processor 10-7
Instruction Section 10-7
Registers 10-7
Execution Section 10-8
Cache Memory 10-8
Addressing Section 10-8
Central Memory Control 10-8
Central Memory 10-9
Input/Output Unit 10-10
System Console 10-11

Model 990 System Description 10

This chapter describes the physical and functional characteristics and major system components.

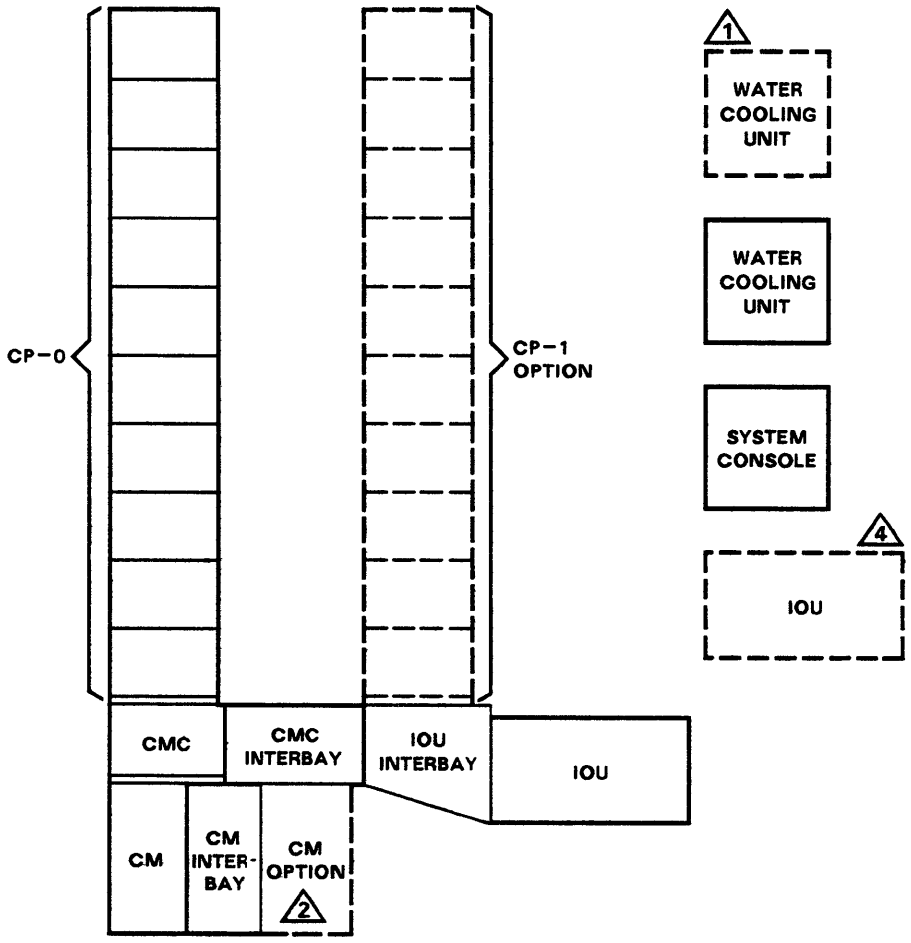
This high-speed computer systems is used for both business and scientific applications. The system includes the following components.

- Central processor (CP)
- Central memory (CM)
- Input/output unit (IOU)

Physical Characteristics

The system configuration (figure 10-1) includes interconnected CP, CM, and IOU cabinet sections that compose the system cabinet. (The system console is also required for system operation.) Each cabinet section contains a logic chassis with plug-in circuit boards. The CP consists of 10 sections, plus a single section for central memory control (CMC). With the dual-CP option, an additional identical cabinet contains the second CP. (The second CP does not support CYBER 170 State operation.) The CM consists of one section with four memory cages. The IOU consists of two sections and performs initialization and maintenance functions. A second IOU is installed in a separate cabinet for a dual-IOU option.

The CP, CM, and IOU sections each contain an ac/dc control section with voltage margin testing facilities and dc power supplies. A standalone cooling unit provides cooling for the CP, CM, and IOU logic chassis.



1. Optional water cooling unit required for CP-1.

2. Optional CM requires additional water cooling unit if water cooling unit for CP-1 is not installed.

3. Refer to the appropriate site preparation manual for specific water cooling unit requirements.

4. Second IOU required for dual-IOU option.

Figure 10-1. System Configuration

Functional Characteristics

Emitter-coupled logic (ECL) and large-scale integration (LSI) logic are used to achieve high computation speeds. High speed is also the objective of the CP design, which is based on the capability to execute many operations concurrently.

The CP supports two states of operation.

Virtual State	Operates with virtual-memory byte addressing, using the Virtual State instruction set and data formats. Virtual State is the native operating state of the CP.
CYBER 170 State	Operates with real-memory word addressing, using the CYBER 170 State CP instruction set and data formats.

The Virtual State and CYBER 170 State environments may be present at the same time with the CP executing in either environment.

- NOS/VE is the operating system of Virtual State.
- NOS is the operating system of CYBER 170 State.

The bipolar central memory is divided into eight independent banks to minimize conflicts between central memory requests. System input/output speeds are determined by the capabilities of existing external devices.

Central Processor

The CP has the following characteristics:

- 60-bit internal word.
- Eight 60-bit operand (X) registers.
- Eight 18-bit address (A) registers.
- Eight 18-bit index (B) registers.
- Two registers that isolate each user's extended memory space (RAC, FLC).
- Two registers that isolate each user's extended memory space (RAE, FLE).
- Register exchange instructions (exchange jumps) for interrupting programs.
- Floating-point (FP) arithmetic (10-bit exponent plus sign bit, 48-bit coefficient plus sign bit). Some FP instructions use 96-bit (double precision) coefficients.
- Integer arithmetic (60/18-bit operands).
- Character string compare/move facilities (6-bit characters).
- Packed instructions (15/30/60-bit instructions in 60-bit words).
- Synchronous internal logic.
- 16-nanosecond clock period.
- 4096-word cache buffer memory.
- Instruction and branch instruction lookahead.
- Microcode control.
- Parity checking of selected data and address paths.
- Maintenance channel to IOU.

Central Memory

The CM has the following characteristics:

- 72-bit data word (60 data bits, 8 single-error correction/double-error detection bits, and 4 unused bits).
- 1048K words of bipolar memory, options available to 4194K words in 1048K-word increments.
- Organization of eight independent banks.
- Memory ports (located in the central processor cabinet).
- Bounds register to limit write access.
- 64-nanosecond clock period.
- Maximum data transfer rate of four words every 16 nanoseconds.
- 80-nanosecond read access time.
- 64-nanosecond read/write cycle time.
- 192-nanosecond partial-write cycle time.
- Read and write data queuing capability.
- Single-error correction/double-error detection (SECDED) on stored data.
- Parity checking of all major data, address, and control paths.
- Unified-extended memory (UEM) which serves as extended memory within CM.

Input/Output Unit

The IOU has the following characteristics:

- A nonconcurrent input/output (NIO) subsystem consisting of 20 CYBER 170 compatible peripheral processors (PPs). Each PP has an 8K x 16-bit word independent memory (PPM) degradeable to 4K x 16 bits (refer to the CIP User's Handbook listed under Additional Related Manuals in About This Manual for information on degrading PPMs).
- A concurrent input/output (CIO) subsystem consisting of five or ten PPs. Each PP has an 8K x 16-bit PPM. In addition, there is an optional direct-memory access (DMA)-enhanced intelligent standard interface (ISI) channel adapter, an optional DMA intelligent peripheral interface (IPI) channel adapter, and an optional CYBER 170 DMA channel adapter. These adapters support DMA transfer between CM and an external device as well as standard I/O data transfer. An adapter can be installed in any one of ten channel locations in the CIO cabinet. There are two types of CYBER 170 DMA transfers, fast and normal. Fast transfers are used with the Extended Semiconductor Memory-II (ESM-II), and normal transfers are used with other CYBER 170 external devices.
- Execution of 12- or 16-bit PP code.
- Port to central memory.
- Twenty-four CYBER 170 compatible I/O channels available with a maximum data transfer rate of 3 megabytes/second.
- Interface to real-time clock, display controller, and two-port multiplexer.
- Bounds register controlling write access to CM.
- SECDED data verification on all PP memories.
- Parity checking on all major data and address paths.
- Maintenance channels giving PPs access to CP, CM, and IOU registers to perform system initialization and maintenance functions.
- Operating speed of 250 nanoseconds and a minor cycle of 50 nanoseconds.

Major System Component Descriptions

Central Processor

The CP hardware (figure 10-2) consists of the following:

- Instruction section
- Registers
- Execution section
- Cache memory
- Addressing section
- Central memory control

Central Processor 0 (CP-0) is capable of dual-state operation: Virtual State and CYBER 170 State. If a second central processor is added (CP-1), it is capable of only Virtual State operation.

The CP is isolated from the IOU and is thus able to carry on computation of character manipulation unencumbered by I/O requirements.

Instruction Section

The instruction section directs the arithmetic and manipulative functions for instruction execution. The instruction section prefetches instruction words from memory and disassembles them into instructions.

Registers

Operating registers reduce storage accesses for operands used during the execution of an instruction. These registers are:

- Eight 60-bit X registers (X0 through X7) which hold operands used for computation.
- Eight 18-bit A registers (A0 through A7) which use A0 primarily for indexing and A1 through A7 for CM operand addressing.
- Eight 18-bit B registers (B0 through B7) which are primarily indexing registers to control program execution. The B0 register always contains all zeros.

Eight support registers support the operating registers during program execution. These registers are:

- 18-bit program address (P) register.
- 21-bit reference address for CM (RAC) register. This is a program's lower bound.
- 21-bit field length for CM (FLC) register. This is a program's upper bound.
- 6-bit exit mode (EM) register.
- 6-bit flag register.
- 21-bit reference address for UEM (RAE) register.
- 24-bit field length for UEM (FLE) register.
- 18-bit monitor address (MA) register.

The registers store data and control information, present operands to the execution section, and store results.

The operating and support registers reside in the register unit and process state registers section.

Execution Section

The execution section combines the operands to achieve the result.

Cache Memory

The cache memory consists of 4096 words. The memory addressing sections determine whether a requested word is in the cache memory. If it is not, they read four consecutive words from central memory into the cache memory.

Addressing Section

The addressing section checks memory addresses against the CP registers RAC, FLC, RAE, and FLE to ensure isolation of user memory space.

Central Memory Control

Central memory control (CMC) is integrated within the CP and controls the flow of data between CM and requesting system components.

Central Memory

The CM (figure 10-2) consists of the following:

- Eight memory banks
- Memory ports

The CM is a bipolar memory organized into eight independent banks.

A portion of CM can be reserved for use as extended memory. It is called unified extended memory (UEM) and is referenced by the RAE and FLE registers. The UEM operates in 24-bit standard addressing mode. All memory ports have queuing buffers. The memory ports are located in the central processor cabinet.

The following CM option is available.

Option	Model	Description
18855	001	Adds 7M 64K-words to increase size from 1M to 8M words (8 MB to 64 MB). Use of Upgrade Option 18850 allows additional expansion up to 16M words (128 MB). Memory sizes above 2M words (16 MB) are supported by NOS/VE only.

Input/Output Unit

The IOU (figure 10-2) consists of:

- Twenty logically independent nonconcurrent input/output (NIO) peripheral processors (PPs). Options are available to increase the total to 25 or 30 PPs.
- Five or ten logically independent concurrent input/output (CIO) PPs and DMA channel adapters.
- Internal interface to 24 I/O channels. Options are available to increase the total to 34 channels.
- External interfaces to I/O channels:
  - 11 or 23 CYBER 170 channel interfaces.
  - Display controller interface (CYBER 170 channel 10_8).
  - Real-time clock interface (channel 14_8).
  - Two-port multiplexer interface (channel 15_8).
  - Maintenance channel interface (channel 17_8).
- Interface to CM.
- Bounds register to limit writes and exchanges to CM.

The PPs are organized in groups of five, called barrels. The PPs in a barrel time-share common hardware. Each PP has its own 4K or 8K independent memory, and communicates with all I/O channels and with CM.

System Console

The system console, required for system operation, provides a visual, alphanumeric readout for the computer. The receipt of symbol and position information from the computer enables displaying program information on a cathode-ray tube (CRT). The station also contains an alphanumeric keyboard which enables an operator to send data to the computer. The keyboard and CRT combination permits the computer operator to monitor and control system operation. Except for programming information in chapter 17, refer to the system console manual listed in additional related manuals in About This Manual for further information.

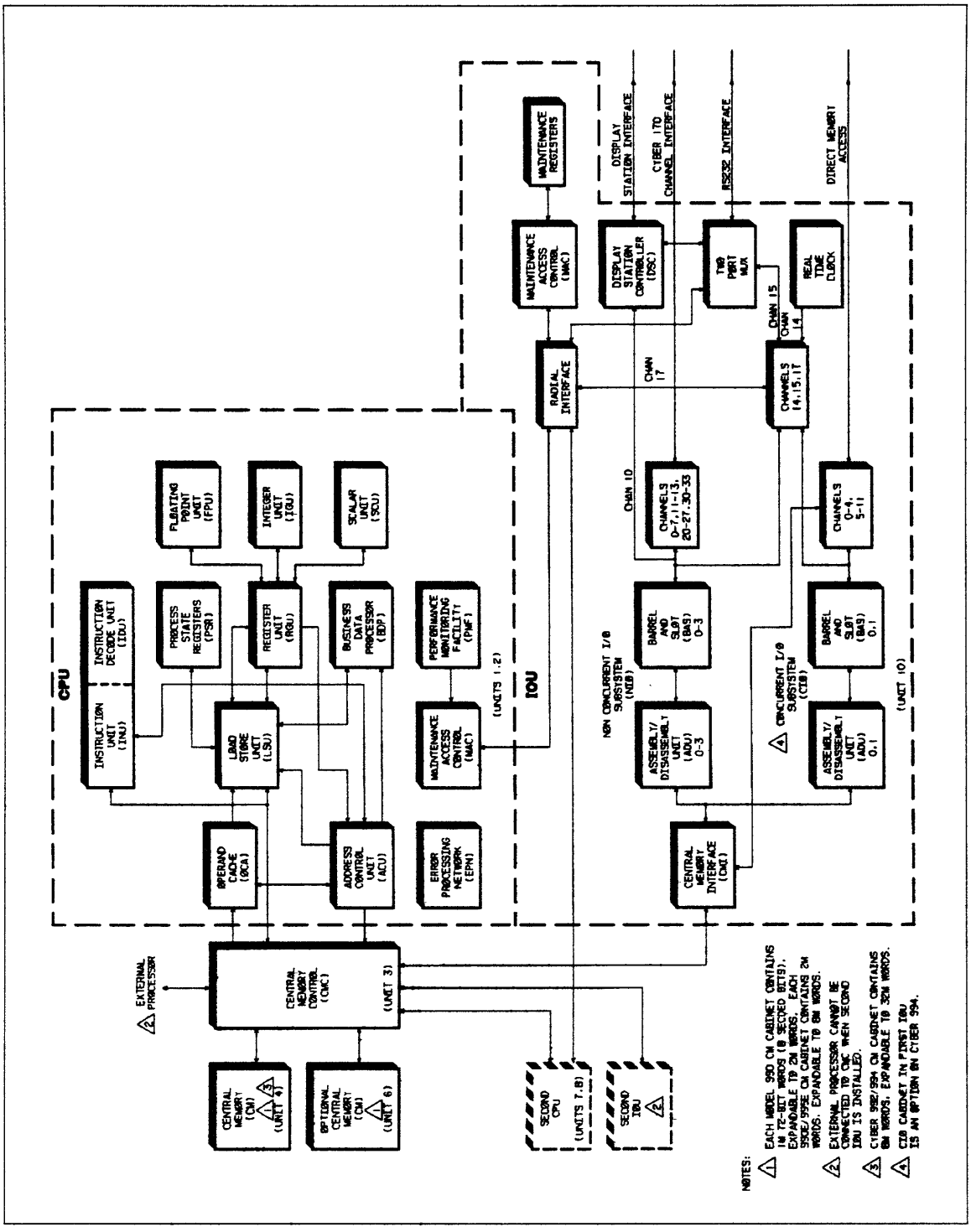


Figure 10-2. Computer System Block Diagram

Model 990 Functional Descriptions 11

Central Processor 11-1
Instruction Section 11-1
Instruction Lookahead 11-1
Maintenance Access Control 11-1
Instruction Control Sequences 11-2
Boolean Sequence 11-2
Shift Sequence 11-3
Floating-Add Sequence 11-4
Floating-Multiply and Floating-Divide Sequence 11-4
Increment Sequence 11-4
Compare/Move Sequence 11-6
CYBER 170 Exchange Sequence 11-7
Block Copy Sequence 11-7
Direct Read/Write Sequence 11-7
Normal Jump Sequence 11-8
Return Jump Sequence 11-8
Registers 11-9
Operating Registers 11-10
X Registers 11-10
A Registers 11-10
B Registers 11-10
Support Registers 11-11
P Register 11-11
RAC Register 11-11
FLC Register 11-11
EM Register 11-12
Flag Register 11-12
RAE Register 11-13
FLE Register 11-13
MA Register 11-13
Execution Section 11-14
Cache Memory 11-14
Addressing Section 11-14
Central Memory Control 11-14

Central Memory 11-15
Address Format 11-15
CM Access and Cycle Times 11-16
CM Ports and Priorities 11-16
SECDED 11-17
CM Layout 11-22
CM Bounds Register 11-22
Central Memory Reconfiguration 11-22

Input/Output Unit 11-23
Peripheral Processor 11-24
Deadstart 11-26
Barrel and Slot 11-26
PP Registers 11-28
R Register 11-28
A Register 11-28
Q Register 11-29
P Register 11-29

K Register 11-29
PP Numbering 11-30
PP Memory 11-30
I/O Channels 11-31
Display Station Controller 11-31
Real-Time Clock 11-31
Two-Port Multiplexer 11-31
Maintenance Channel 11-32
IOU Maintenance Registers 11-32
Element Identifier (EID) Register 11-33
Environment Control (EC) Register 11-33
Fault Status (FS) Registers 11-33
Fault Status Mask Register 11-33
Options Installed (OI) Register 11-34
OS Bounds Register 11-34
Status Summary Register 11-34
Test Mode (TM) Register 11-34
Central Memory Access by PPs 11-34

Model 990 Functional Descriptions 11

This chapter provides functional descriptions of the central processor (CP), central memory (CM), and input/output unit (IOU) as shown in the system block diagram in chapter 10. Functional descriptions for the system display station and the cooling system are in their respective manuals listed in the system publication index in About This Manual.

Central Processor

The CP consists of the instruction section, registers, execution section, cache memory, addressing section, and central memory control.

Instruction Section

The instruction section consists of logic for instruction control.

Instruction Lookahead

The instruction lookahead hardware (ILH) speeds up instruction processing by stacking prefetched instructions to make them immediately available for execution. It also accurately predicts program branching based on the recent history of each conditional branch.

To maintain a continuous flow of instructions, the instruction section prefetches instruction words ahead of the instruction being read and stores them in the 64-word instruction buffer stack (IBS). This high-speed buffer is set in the instruction stream between CM and the CP execution section. When an instruction is requested for execution, IBS checks whether that instruction is in the stack. If in the stack, the instruction proceeds to instruction decode and initiation. If not in the stack, the instruction is fetched from CM and placed in the IBS. A least-recently-used replacement algorithm determines the new instruction's position in the IBS.

If the instruction issued is a conditional branch, the instruction section predicts the branch taken or not taken based on the recent history of the branch. It then prefetches instructions along that path before the branch outcome is known.

The actual result of the branch determines whether the conditionally issued instructions may execute to completion. If a branch prediction is correct, the instruction section enables the modifying of registers and CM from issued instructions. If a branch prediction is incorrect, hardware purges the issued instructions along the incorrect branch and issues instructions along the correct branch. In this case, registers and CM appear as if no instructions were issued after the incorrectly predicted branch. As a precaution to incorrect branch prediction, the instruction section will not issue subsequent branch instructions until the prior branch instruction is resolved.

Maintenance Access Control

The maintenance access control performs initialization and maintenance operations in the CP.

Instruction Control Sequences

The instruction control section performs instruction translation and control sequences. Each control sequence obtains the necessary instruction operands from the operating registers and provides the control signals for execution. Instructions read from CM are 60-bit instruction words that are in four 15-bit groups, two 30-bit groups, or a combination of 15-bit and 30-bit groups. The 15-bit groups are termed parcels with the first parcel (parcel 0) being the highest-order 15 bits of a 60-bit CM word. Second, third, and fourth parcels (parcels 1, 2, and 3) follow in order. The 30-bit groups contain two 15-bit parcels.

The instruction control sequences control the execution of one or more instructions of a common type. These sequences and associated instructions are briefly described in this chapter. For further information, refer to CP Instruction Descriptions in chapter 16.

Boolean Sequence

The Boolean sequence controls instructions that require bit-by-bit data manipulation. This includes both the logical and transmissive operations. The instructions requiring logical operations are:

11	Logical product (Xj) and (Xk) to Xi	BXi Xj * Xk
12	Logical sum of (Xj) and (Xk) to Xi	BXi Xj + Xk
13	Logical difference of (Xj) and (Xk) to Xi	BXi Xj - Xk
15	Logical product of (Xj) with complement of (Xk) to Xi	BXi -Xk * Xj
	Logical sum of (Xj) with complement of (Xk) to Xi	BXi -Xk + Xj
17	Logical difference of (Xj) with complement of (Xk) to Xi	BXi -Xk - Xj

The instructions requiring transmissive operations are:

10	Transmit (Xj) to Xi	BXi Xj
11	Transmit complement of (Xk) to Xi	BXi -Xk

Shift Sequence

The shift sequence controls instructions that require shifting the 60-bit field of data within the operand word. The shift instructions are:

20	Left shift (Xi) by jk	LXi jk
21	Right shift (Xi) by jk	AXi jk
22	Left shift (Xk) nominally (Bj) places to Xi	LXi Bj, Xk
23	Right shift (Xk) nominally (Bj) places to Xi	AXi Bj, Xk
43	Form mask of jk bits to Xi	MXi jk

The shift sequence also controls the pack and unpack instructions. In the packed floating format, the coefficient is contained in the lower 48 bits. The sign and biased exponents are contained in the upper 12 bits. The unpack instruction obtains the packed word from the Xk register, delivers the coefficient to the Xi register, and delivers the exponent to the Bj register. The unpack and pack instructions are:

26	Unpack (Xk) to Xi and Bj	UXi Bj, Xk
27	Pack (Xk) and (Bj) to Xi	PXi Bj, Xk

The shift sequence also controls the normalize operations. The coefficient portion of the operand is repositioned, and the exponent is adjusted so that the most significant bit of the coefficient is in the highest-order bit position of the coefficient, and the exponent is decreased by the number of bit positions shifted. The normalize instructions are:

24	Normalize (Xk) to Xi and Bj	NXi Bj, Xk
25	Round normalize (Xk) to Xi and Bj	ZXi Bj, Xk

Floating-Add Sequence

The floating-add sequence controls the operations necessary to form the 48-bit floating sum with a 12-bit exponent of the floating-point sum or difference of two floating-point operands. The floating-add instructions are:

30	Floating sum of (Xj) and (Xk) to Xi	FXi Xj + Xk
31	Floating difference of (Xj) and (Xk) to Xi	FXi Xj - Xk
32	Floating double-precision sum of (Xj) and (Xk) to Xi	DXi Xj + Xk
33	Floating double-precision difference of (Xj) and (Xk) to Xi	DXi Xj - Xk
34	Round floating sum of (Xj) and (Xk) to Xi	RXi Xj + Xk
35	Round floating difference of (Xj) and (Xk) to Xi	RXi Xj - Xk

Floating-Multiply and Floating-Divide Sequence

The floating-multiply and floating-divide sequence controls the operation of floating-multiply, floating-divide, and population-count instructions.

The multiply instructions are:

40	Floating product of (Xj) and (Xk) to Xi	FXi Xj * Xk
41	Round floating product of (Xj) and (Xk) to Xi	RXi Xj * Xk
42	Floating double-precision product of (Xj) and (Xk) to Xi	DXi Xj * Xk

The divide instructions are:

44	Floating divide (Xj) by (Xk) to Xi	FXi Xj/Xk
45	Round floating divide (Xj) by (Xk) to Xi	RXi Xj/Xk

The population-count instruction counts the number of one bits in a 60-bit operand. The instruction is:

47	Population count of (Xk) to Xi	CXi Xk

Increment Sequence

The increment sequence controls the ones complement addition and subtraction of 18-bit fixed-point operands for increment instructions 50 through 77. The sequence also controls the 60-bit ones complement sum and difference values for long-add instructions 36 and 37.

The increment instructions are:

50	Set Ai to (Aj) + K	SAi Aj + K
51	Set Ai to (Bj) + K	SAi Bj + K
52	Set Ai to (Xj) + K	SAi Xj + K
53	Set Ai to (Xj) + (Bk)	SAi Xj + Bk
54	Set Ai to (Aj) + (Bk)	SAi Aj + Bk
55	Set Ai to (Aj) - (Bk)	SAi Aj - Bk
56	Set Ai to (Bj) + (Bk)	SAi Bj + Bk
57	Set Ai to (Bj) - (Bk)	SAi Bj - Bk
60	Set Bi to (Aj) + K	SBi Aj + K
61	Set Bi to (Bj) + K	SBi Bj + K
62	Set Bi to (Xj) + K	SBi Xj + K
63	Set Bi to (Xj) + (Bk)	SBi Xj + Bk
64	Set Bi to (Aj) + (Bk)	SBi Aj + Bk
65	Set Bi to (Aj) - (Bk)	SBi Aj - Bk
66	Set Bi to (Bj) + (Bk)	SBi Bj + Bk
67	Set Bi to (Bj) - (Bk)	SBi Bj - Bk
70	Set Xi to (Aj) + K	SXi Aj + K
71	Set Xi to (Bj) + K	SXi Bj + K
72	Set Xi to (Xj) + K	SXi Xj + K
73	Set Xi to (Xj) + (Bk)	SXi Xj + Bk
74	Set Xi to (Aj) + (Bk)	SXi Aj + Bk
75	Set Xi to (Aj) - (Bk)	SXi Aj - Bk
76	Set Xi to (Bj) + (Bk)	SXi Bj + Bk
77	Set Xi to (Bj) - (Bk)	SXi Bj - Bk

The long-add instructions are:

36	Integer sum of (Xj) and (Xk) to Xi	IXi Xj + Xk
37	Integer difference of (Xj) and (Xk) to Xi	IXi Xj - Xk

Compare/Move Sequence

The compare/move sequence controls data manipulation on a character basis. The compare/move instructions (also referred to as CMU instructions) are 60-bit instructions that use six support registers for source and result field CM addresses and character position offsets. The support registers load from the 60-bit instruction word. The compare/move instructions are:

464	Move indirect (Bj) + K	IM Bj + K
465	Move direct	DM
466	Compare collated	CC
467	Compare uncollated	CU

The support registers are:

- An 18-bit K1 register that specifies which relative CM address word contains the first character of the source data field.
- An 18-bit K2 register that specifies which relative CM address word contains the first character of the result field.
- A 4-bit C1 register that specifies the character position or offset of the first CM word of the source field.
- A 4-bit C2 register that specifies the character position or offset of the first CM word of the result field.
- Two 16-bit L registers (LA and LC) that specify the number of characters in the data field. The LA register is associated with K1, and the LC register is associated with K2. Instruction 464 uses 14 register bits. Instructions 465, 466, and 467 use only the lower 8 register bits.

NOTE

CMU instructions are provided for compatibility with previous systems. For better performance, recompile jobs to avoid use of CMU instructions.

CYBER 170 Exchange Sequence

A CYBER 170 exchange sequence is the method used to swap jobs in and out of execution. When a CYBER 170 exchange jump instruction occurs, the CYBER 170 exchange sequence writes the contents of the current job's CP registers (described later in this chapter) into an area of central memory called a CYBER 170 exchange package. A CYBER 170 exchange package is associated with each job. It contains sufficient information to restart a job if the job is interrupted during execution and swapped out by a CYBER 170 exchange jump. To complete the sequence, CP registers for another job are read from its CYBER 170 exchange package and that job begins or resumes execution. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

Block Copy Sequence

The block copy sequence controls the transfer of data between CM and UEM. The number of words to be transferred is determined by the addition of K to the contents of Bj. The starting address for CM is formed by adding either the A0 register or certain bits of the X0 register to the RAC reference address. The starting address for UEM is formed by adding certain bits of the X0 register to the RAE reference address. The block copy instructions are:

011	Block copy Bj + K words from UEM to CM	RE Bj + K
012	Block copy Bj + K words from CM to UEM	WE Bj + K

Direct Read/Write Sequence

Instructions 014 and 015 perform single-word, direct read and write operations for UEM; and instructions 660 and 670 perform single-word, direct read and write operations for central memory.

014	Read one word from UEM at (Xk + RAE) into Xj	RXj Xk
015	Write one word from Xj to UEM at (Xk + RAE)	WXj Xk
660	Read central memory at (Xk) to Xj	CRXj Xk
670	Write Xj into central memory at (Xk)	CWXj Xk

Normal Jump Sequence

The normal jump sequence controls the execution of branch instructions 02 through 07. The 02 instruction performs an unconditional jump to the Bi register address plus K. The branch address is K with i equals zero. The 02 instruction is:

02	Jump to (Bi) + K	JP Bi + K

The conditional jump instructions 03 through 07 branch to address K if the jump condition is met.

These instructions are:

030	Branch to K if (Xj) = 0	ZR Xj, K
031	Branch to K if (Xj) = 0	NZ Xj, K
032	Branch to K if (Xj) is positive	PL Xj, K
033	Branch to K if (Xj) is negative	NG Xj, K
034	Branch to K if (Xj) is in range	IR Xj, K
035	Branch to K if (Xj) is out of range	OR Xj, K
036	Branch to K if (Xj) is definite	DF Xj, K
037	Branch to K if (Xj) is indefinite	ID Xj, K
04	Branch to K if (Bi) = (Bj)	EQ Bi, Bj, K
05	Branch to K if (Bi) ≠ (Bj)	NE Bi, Bj, K
06	Branch to K if (Bi) ≥ (Bj)	GE Bi, Bj, K
07	Branch to K if (Bi) < (Bj)	LT Bi, Bj, K

Return Jump Sequence

The return jump sequence controls the execution of three instructions.

00	Error exit to MA or program stop	PS
010	Return jump to K	RJ K
013	Central exchange jump to (Bj) + K or monitor exchange jump to MA	XJ Bj + K

Registers

The CP contains the operating and support registers described in the following paragraphs. These registers are located in the register unit and process state registers sections (refer to figure 10-2).

The contents of these registers can be written into memory and reloaded from memory as a CYBER 170 exchange package by a single CP instruction (CYBER 170 exchange jump). Figure 11-1 shows the CYBER 170 exchange package.

The time a CYBER 170 exchange package resides in CP hardware is called an execution interval. During this interval, the contents of X, A, B, and P registers can be changed by CP instructions. The contents of other support registers change only as a result of a CYBER 170 exchange jump. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

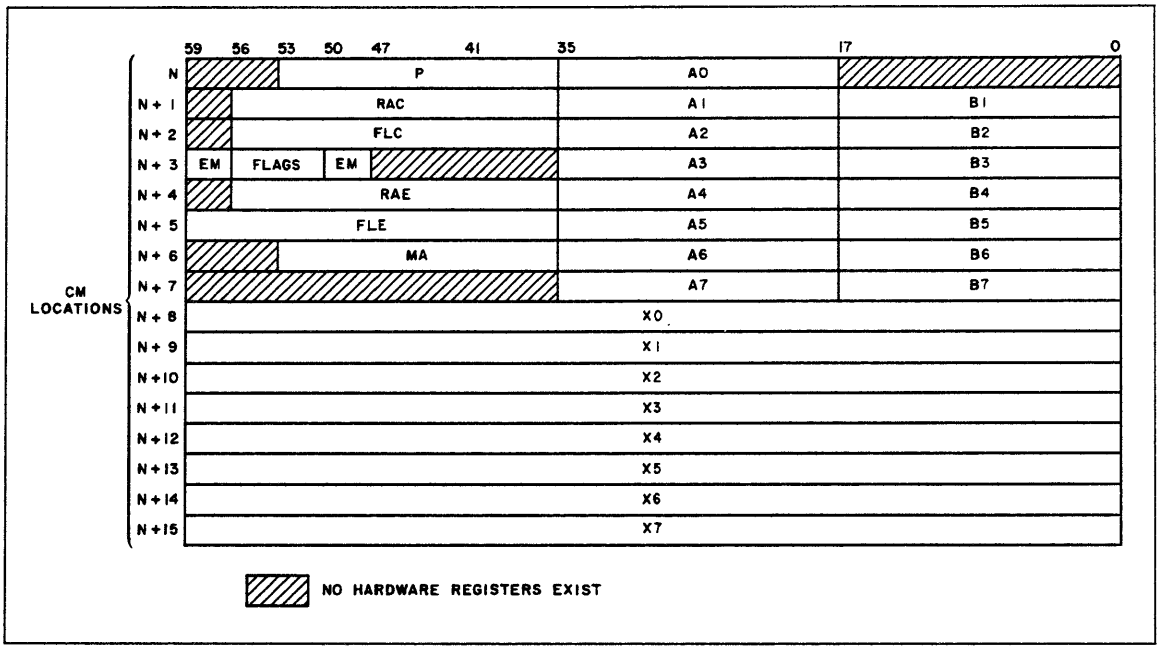


Figure 11-1. CYBER 170 Exchange Package

Operating Registers

The operating registers consist of operand (X), address (A), and index (B) registers. These registers minimize memory references for arithmetic operands and results.

X Registers

The CP contains eight 60-bit X registers, X0 through X7. The X0 register is used in the compare instructions to indicate if two fields of characters are equal. Also, the X0 register provides the relative UEM starting address in a block copy operation.

The X1 through X7 registers are primarily data handling registers for computation. X1 through X5 are used to input data from CM, and X6 and X7 are used to transmit data to CM.

Operands and results transfer between CM and the X registers as a result of placing CM addresses into corresponding A registers.

A Registers

The CP contains eight 18-bit A registers, A0 through A7. The A0 register serves as an intermediate register for the user's discretion. The A0 register is used in the compare collate instruction for the collate table address. Also, the A0 register provides the relative CM starting address in a block copy operation.

The A1 through A7 registers are essentially CM operand address registers associated one-for-one with the X registers. Placing a quantity into an address register (A1 through A5) causes a CM read reference to that address and transmits the CM word to the corresponding X register (X1 through X5). Similarly, placing a quantity into the A6 or A7 register causes the word in the corresponding X6 or X7 register to be written into that relative address of CM.

B Registers

The CP contains eight 18-bit B registers, B0 through B7. These registers are primarily indexing registers to control program execution. Program loop counts may also be incremented or decremented in these registers.

Program addresses may be modified on the way to an A register by adding or subtracting B register quantities. The B registers also hold shift counts for the nominal Bj shifts, the resultant exponent for the unpack, the operand exponent for the pack, and the resultant shift count from a normalize. The B0 register always contains positive zero which can be used as an operand. This register cannot hold results from instructions.

Support Registers

Eight support registers assist the operating registers during the execution of programs. The contents of the support registers are stored in CM, and their new contents are loaded from CM during a CYBER 170 exchange sequence. With the exception of the P register, the contents of the support registers cannot be altered during the execution interval of a CYBER 170 exchange package. When the execution interval completes, the data in the support registers is sent back to CM through a CYBER 170 exchange jump.

P Register

The 18-bit program address (P) register loads from CM during the first word of a CYBER 170 exchange sequence and contains the current program execution address. The register serves as a program address counter and holds the relative CM address for each program step.

RAC Register

The 21-bit CM reference address (RAC) register loads from CM during the second word of a CYBER 170 exchange sequence. An absolute CM address forms by adding RAC to a relative address determined by the instruction. The content of the P register is added to RAC to form the program address in CM. A P-equal-to-zero condition specifies relative address 0 and, therefore, (RAC). This CM location is reserved for recording error exit conditions and should not be used to store data or instructions.

FLC Register

The 21-bit CM field length (FLC) register loads from CM during the third word of a CYBER 170 exchange sequence. The FLC register defines the size of the field of the program in execution. Relative CM addresses are compared with FLC to check that the program is not going out of its allocated memory range.

EM Register

The 6-bit exit mode (EM) register loads from CM during the fourth word of a CYBER 170 exchange sequence. The EM register holds six exit-mode selection bits that control individual error conditions for a program. Selected EM register bits cause the CP to error exit when the corresponding conditions occur. Any or all of the six bits can be set at one time. Clear EM register bits allow the CP to continue, without error processing, when most of the corresponding conditions occur. Refer to the error exit tables under Error Response in chapter 17 for specific cases. The exit mode selection bits appear in the exchange package as bits 48 through 50 and 57 through 59. The bits and their corresponding conditions are:

Mode Selection Bit	Significance
48	Address out of range
49	Infinite operand
50	Indefinite operand
57	Hardware error
58	Hardware error
59	Hardware error

Flag Register

The 6-bit flag register loads from CM during the fourth word of a CYBER 170 exchange sequence. The flag register holds six bits that function as control flags.

Bit	Condition
51	Hardware error bit.
52	Instruction stack (lookahead) purge flag. If set, extended purging of instruction lookahead registers is enabled. For further information, refer to Instruction Lookahead Purge Control under CP Programming in chapter 17.
53	CMU interrupted flag. If set, one of instructions 464 through 467 has been interrupted. The information necessary to resume operation has been saved.
54	Block copy flag. If set, block copy instructions (011, 012) use bits 30 through 50 of X0 rather than A0 to determine the CM address. For further information, refer to the descriptions of the block copy instructions in chapter 16.
55	Expanded addressing select flag. If set, UEM is operating in expanded addressing mode; if clear, UEM is operating in 24-bit standard addressing mode. For further information, refer to Addressing Modes under Memory Programming in chapter 17.
56	UEM enable flag. If set, UEM is available. This flag must be set to allow 011, 012, 014, and 015 instructions to access UEM.

RAE Register

The 21-bit UEM reference address (RAE) register loads from CM during the fifth word of a CYBER 170 exchange sequence. The lower six bits of this register are always zero. An absolute UEM address forms by adding RAE to the relative address which is determined by the instruction.

FLE Register

The 24-bit UEM field length (FLE) register loads from CM during the sixth word of a CYBER 170 exchange sequence. The lower six bits of this register are always zero. The FLE register defines the size of the field in UEM for the program in execution. Relative UEM addresses are compared with FLE.

MA Register

The 18-bit monitor address (MA) register loads from CM during the seventh word of a CYBER 170 exchange sequence. The MA register contains the absolute starting address of an exchange package which is used when executing a central exchange jump (013) instruction with the CYBER 170 monitor flag clear, or when honoring a monitor exchange jump to MA (262x) instruction with the CYBER 170 monitor flag clear. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

Execution Section

The execution section combines the operands into results providing additional sequencing control where necessary.

Cache Memory

Cache memory is a high-speed buffer memory which is transparent to the user. It reduces effective CM access time by eliminating unnecessary CM references. When the CP first reads CM, a block of four words from CM (containing the requested word) is read rapidly into cache memory. These words include all data except instructions. On subsequent reading of any of these words, CM need not be accessed when these words are in cache memory. Often this is the case because the same data is read more than once, or because a loop of instructions is repeatedly executed.

Addressing Section

An address adder calculates memory addresses for data and unconditional jump instructions.

Memory management hardware verifies that memory addresses are to access permitted memory areas. If this is the case, this hardware accesses cache memory and, if necessary, central memory.

Central Memory Control

Central memory control (CMC) provides an interface to CM for the CP and IOU. It is physically located in the CP cabinet. CMC includes:

- Ports and distributor
- SECDED logic
- Partial-write logic
- Memory control logic
- Maintenance registers

Central Memory

The CM performs the following functions.

- A memory cabinet contains eight banks, storing 2M 64-bit words (the leftmost 4 bits are undefined) and an 8-bit SECDED code. A maximum of two memory cabinets allows a total of 4M 64-bit words.
- The two ports make CM accessible to the CP and every PP.
- A bounds register limits access to CM from either or both ports.
- The SECDED generators generate the SECDED code bits stored with each word. SECDED checks circuits, corrects single-bit errors, and detects double-bit errors.
- The maintenance channel interface gives a PP in the IOU access to the CM maintenance registers for system initialization, corrective action, error reporting and diagnostics, and for setting the port bounds register.

Address Format

Figure 11-2 illustrates the address format.

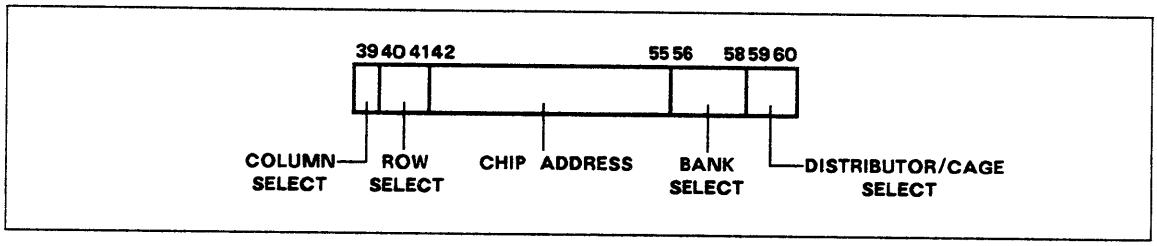


Figure 11-2. Address Format

The following list defines the address fields for figure 11-2.

- Column Select specifies one of two columns.
- Row Select specifies one of four word rows in a bank. (Each word row corresponds to a set of chips in a bank.)
- Chip Address specifies the address of one word in 16K bipolar memory chips for the selected bank.
- Bank Select specifies one of eight banks in a cage.
- Distributor/Cage Select specifies one of four cages in a column. (The cage number in a column corresponds to the distributor number in central memory control.)

CM Access and Cycle Times

The CM access time for a single-word read operation is 208 nanoseconds (13 clock periods). The CM access time for a four-word block read is 256 nanoseconds (16 clock periods).

Cycle time for a normal read or write operation is 64 nanoseconds (4 clock periods). Cycle time for a partial write, read/set/clear/lock, and exchange is 192 nanoseconds (12 clock periods).

The CM bounds register, located in CMC, limits CM write access for the selected ports to an area between two addresses specified in this register. The CM bounds register is set through the maintenance channel (refer to Maintenance Channel Programming in chapter 17).

CM Ports and Priorities

A priority network resolves access conflicts on a rotating basis, preventing long-term lockout of any port. In case of simultaneous requests, the CP has priority.

Refresh requests have priority over port requests. Refer to table 11-1 for maximum request lockout time in bank cycles.

Table 11-1. Port Priority

Port	Read or Write Requests
Refresh	1
Port 0	4
Port 1	5

Note:

One bank cycle equals 4 clock periods (64 nanoseconds).

SECDED

The SECDED logic corrects single-bit errors during a CM read, permitting unimpeded computer operation. The SECDED logic prepares for the error correction by generating error correction code (ECC) bits for each data word, and by storing these ECC bits in CM with the data word during the CM write. Table 11-2 lists the hexadecimal codes for all the combinations of syndrome bits with the number of the data bit assigned each code or a note categorizing the code. Then, during a CM read, CM performs the following SECDED sequence.

1. Read one CM word and generate new ECC bits for data portion of CM word.
2. Compare new ECC bits with CM word ECC bits.
3. If old and new ECC bits match, no error exists. Send data to requesting unit.
4. If bits do not match, generate syndrome bits from result of ECC compare.
5. Decode syndrome bits to determine if single or multiple bit failure.
6. If single bit failure, correct by inverting failing bit in data word. Send corrected word to requesting unit.
7. If multiple bit or other uncorrectable error, send uncorrectable error response code to CP or IOU. A PP in the IOU may then analyze the syndrome bits using the maintenance channel.

Table 11-2. SECDED Syndrome Codes/Corrected Bits

Code	Bit	Code	Bit	Code	Bit	Code	Bit
00	6	10	67[2]	20	66[2]	30	2/3[5]
01	71[2]	11	3	21	3	31	4
02	70[2]	12	3	22	3	32	4
03	6/7[5]	13	4	23	4	33	3
04	69[2]	14	3	24	3	34	4
05	3	15	4	25	4	35	3
06	3	16	4	26	4	36	3
07	24[1]	17	5	27	5	37	28[1]
08	68[2]	18	3	28	3	38	4
09	3	19	4	29	4	39	3
0A	3	1A	4	2A	4	3A	3
0B	16[1]	1B	5	2B	5	3B	20[1]
0C	4/5[5]	1C	4	2C	4	3C	3
0D	8[1]	1D	5	2D	5	3D	12[1]
0E	0[1]	1E	5	2E	5	3E	4[1]
0F	3	1F	4	2F	4	3F	3

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes indicated.

6. No error detected.

(Continued)

Table 11-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
40	65[2]	50	3	60	3	70	56[1]
41	3	51	4	61	4	71	5
42	3	52	4	62	4	72	5
43	4	53	3	63	3	73	60[1]
44	3	54	4	64	4	74	5
45	4	55	3	65	3	75	58[1]
46	4	56	3	66	3	76	62[1]
47	5	57	26[1]	67	30[1]	77	5
48	3	58	4	68	4	78	5
49	4	59	3	69	3	79	57[1]
4A	4	5A	3	6A	3	7A	61[1]
4B	5	5B	18[1]	6B	22[1]	7B	5
4C	4	5C	3	6C	3	7C	59[1]
4D	10[5]	5D	10[1]	6D	14[1]	7D	5
4E	5	5E	2[1]	6E	6[1]	7E	5
4F	4	5F	3	6F	3	7F	63[1]

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes indicated.

(Continued)

Table 11-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
80	64[2]	90	3	A0	3	B0	48[1]
81	3	91	4	A1	4	B1	5
82	3	92	4	A2	4	B2	5
83	4	93	3	A3	3	B3	52[1]
84	3	94	4	A4	4	B4	5
85	4	95	3	A5	3	B5	50[1]
86	4	96	3	A6	3	B6	54[1]
87	5	97	25[1]	A7	29[1]	B7	5
88	3	98	4	A8	4	B8	5
89	4	99	3	A9	3	B9	49[1]
8A	4	9A	3	AA	3	BA	53[1]
8B	5	9B	17[1]	AB	21[1]	BB	5
8C	4	9C	3	AC	3	BC	51[1]
8D	5	9D	9[1]	AD	13[1]	BD	5
8E	5	9E	1[1]	AE	5[1]	BE	5
8F	4	9F	3	AF	3	BF	55[1]

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes indicated.

(Continued)

Table 11-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
C0	0/1[5]	D0	40[1]	E0	32[1]	F0	3
C1	4	D1	5	E1	5	F1	4
C2	4	D2	5	E2	5	F2	4
C3	3	D3	44[1]	E3	36[1]	F3	3
C4	4	D4	5	E4	5	F4	4
C5	3	D5	42[1]	E5	34[1]	F5	3
C6	3	D6	46[1]	E6	38[1]	F6	3
C7	27[1]	D7	5	E7	5	F7	31[1]
C8	4	D8	5	E8	5	F8	4
C9	3	D9	41[1]	E9	33[1]	F9	3
CA	3	DA	45[1]	EA	37[1]	FA	3
CB	19[1]	DB	5	EB	5	FB	23[1]
CC	3	DC	43[1]	EC	35[1]	FC	3
CD	11[1]	DD	5	ED	5	FD	15[1]
CE	3[1]	DE	5	EE	5	FE	7[1]
CF	3	DF	47[1]	EF	39[1]	FF	3

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes indicated.

CM Layout

Central memory contains an area that is reserved for special software called Virtual State software. Along with the hardware and microcode, this software handles the operations of Virtual State as described in chapter 17. Virtual State software is located at the higher end of memory. The remaining memory is available to the CYBER 170 State and may be allocated as central memory (accessible via RAC and FLC) or as unified extended memory (accessible via RAE and FLE and the 011, 012, 014, and 015 instructions). Refer to figure 11-3.

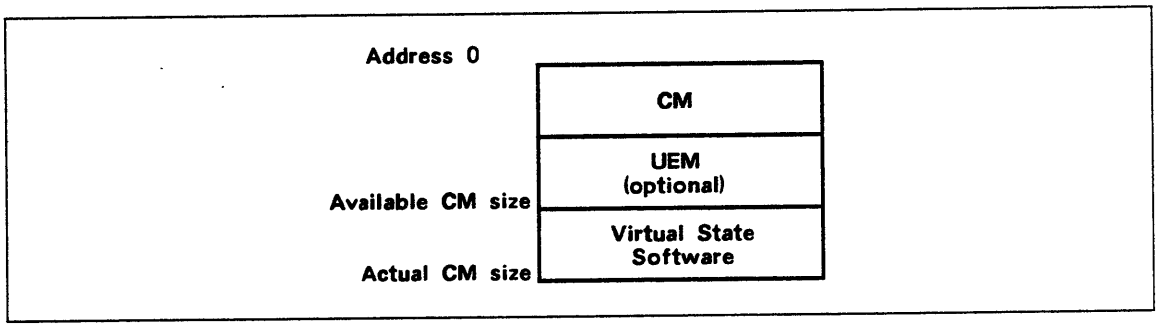


Figure 11-3. CM Layout

CM Bounds Register

The CM bounds register limits the write access to CM from specified ports. The ports are limited to the area between an upper and lower bound as specified in the CM bounds register. Bits in byte 0 specify the port(s) from which the write access is limited. The CM bounds register is set through the maintenance channel. For further information, refer to Maintenance Channel Programming in chapter 17.

Central Memory Reconfiguration

Central memory reconfiguration is a manually performed function that permits the computer operator to restructure the CM addresses so that a failing part of CM can be quickly locked out to provide a continuous block of usable CM. CM reconfiguration is accomplished by setting the switches on the memory unit to manipulate the upper address bits.

When a configuration switch is set forcing a CM address bit to a zero/one, the address range corresponding to the original installed memory accesses some parts of the reconfigured memory more than once. Addresses up to the rightmost forced bit, and half the addresses using the rightmost forced bit, cover a contiguous address space from location 0, which is the reconfigured memory. For further information, refer to chapter 12.

Input/Output Unit

The IOU consists of the nonconcurrent input/output (NIO) subsystem and the concurrent input/output (CIO) subsystem. The NIO consists of 20 PPs, each having 8K x 16-bit words of memory and a repertoire of 122 instructions. All 20 PPs share access to 24 I/O channels. The PPs are partitioned into four barrels of five PPs.

The CIO consists of one or two barrels each containing five PPs per barrel. Each barrel of the CIO has a group of five direct-memory access (DMA) I/O channels. These channel groups are dedicated to a barrel and no inter-barrel communication is possible on DMA channels.

An optional DMA-enhanced intelligent standard interface (ISI) channel adapter, intelligent peripheral interface (IPI) channel adapter, or CYBER 170 channel adapter can be installed in any one of 10 channel locations in the CIO cabinet. The adapter transfers data between the ISI, IPI or CYBER 170 channel and PP memory using standard I/O intructions. It also supports DMA transfer in which data flows directly between the CM and an external device without going through the PP.

There are two types of CYBER 170 DMA transfers, fast and normal. Fast transfers are used with the Extended Semiconductor Memory-II (ESM-II), and normal transfers are used with other CYBER 170 external devices.

This data flow is called DMA since it accesses memory directly. DMA allows a higher I/O bandwidth and also allows the PP execution unit to operate independently from the I/O channel during data transfers (concurrently). This concurrency gives the PP more instruction cycle time to process I/O requests from the CP.

The IOU performs the functions necessary to locate, select, and initialize the external devices connected to the system and controls the transfer of data between a selected device and CM. The IOU also performs system maintenance functions.

The IOU contains the following functional areas:

- Peripheral processors
- I/O channels
- Display Station Controller
- Real-Time Clock
- Two-Port Multiplexer
- Maintenance channel
- CM access

Peripheral Processor

The basic IOU contains 20 NIO PPs and can be expanded to include five or ten CIO PPs. Each PP is a logically independent computer with its own memory.

Each 5-PP group is organized into a multiplexing system which allows the PPs to share common hardware for arithmetic, logical, and I/O operations without losing independence.

This multiplexing system comprises five ranks of registers termed a barrel. Each rank contains information related to the instruction being executed by one PP.

Within the NIO subsystem:

- Any PP in a barrel may communicate with any other PP over any of the 24 I/O channels.
- The channels are numbered from 0 through 11_8 and from 20_8 through 31_8.
- Inter-PP communication is always on a 16-bit boundary.
- The NIO subsystem supports CYBER 170 peripheral equipment (12 bits) using CYBER 170 channel modules.
- Special I/O instructions are provided to convert 12-bit channel words to 16-bit PP words and vice versa.

Within the CIO subsystem:

- PPs within a barrel may communicate with any other PP in the same barrel on any of the five dedicated I/O channels.
- Communication between barrel or with the NIO subsystem must be done via channels 15_8 or 17_8.
- Inter-PP communication is always on a 16-bit boundary. The CIO channels are 0 through 11_8.

Each PP can communicate with:

- Other PPs over the I/O channels (subject to restrictions stated above).
- The CP via CM read and write operations.
- The CP (in CYBER 170 State operation) by issuing a CYBER 170 State exchange request to a specific CYBER 170 State exchange package associated with the issuing PP.

Each PP can also cause an interrupt condition with the CP operating in either Virtual State or CYBER 170 State.

Each PP executes programs alone or with other PPs to control data transfers between external devices and CM. These programs, called I/O drivers, comprised IOU instructions combined to interact with operating system requests issued through CM.

The I/O drivers translate generalized operating system requests into control functions for accessing the external devices and may also perform device scheduling and optimization.

The I/O drivers use PP memory as a buffer for the data transfer between external devices and CM to isolate IOU data transfers from variations in CM transfer rate.

The optional DMA-enhanced ISI, IPI, and CYBER 170 channels can access CM directly and do not require buffer memories.

Deadstart

A deadstart sequence allows the IOU to initialize itself. A deadstart sequence is initiated, depending on the terminal, as follows:

- CC545 display station console -- press the DEADSTART switch and either the S key or CR.
- CC634-B display terminal -- press CTRL-G, CTRL-R, and either the S key or CR.
- CC598-B system console -- press either CTRL-G or CTRL-F2 and either the S key or CR.

For further information regarding the deadstart sequence and options, refer to CYBER Initialization Package (CIP) Reference Manual listed in About This Manual.

Barrel and Slot

The barrel consists of the A, K, P, Q, and R registers, each one of which has five ranks numbered 0 through 4 (figure 11-4).

Information in these registers is transferred from one rank to the next at a uniform 20-MHz rate, providing a multiplexed system of five PPs, each operating at a 4-MHz rate. The registers are stationary while the PPs rotate. For example, rank 4 registers contain PP0, PP1, PP2, PP3, and PP4 in succession, each consuming 50 nanoseconds of the total cycle time of 250 nanoseconds.

Since PP memories operate at a slower rate, independent memory with its own address and data registers is provided for each PP.

Each time data enters the slot, a portion of the instruction for that data is executed. The slot performs tasks such as arithmetic and logic operations and program address manipulation. Complete execution of an instruction may require the A, K, P, Q, and R register quantities to go more than one trip around the barrel and through the slot.

The PPM may be referenced once each time the PP passes around the barrel and through the slot. During its slot time, the PP may also communicate with CM or with any of the I/O channels.

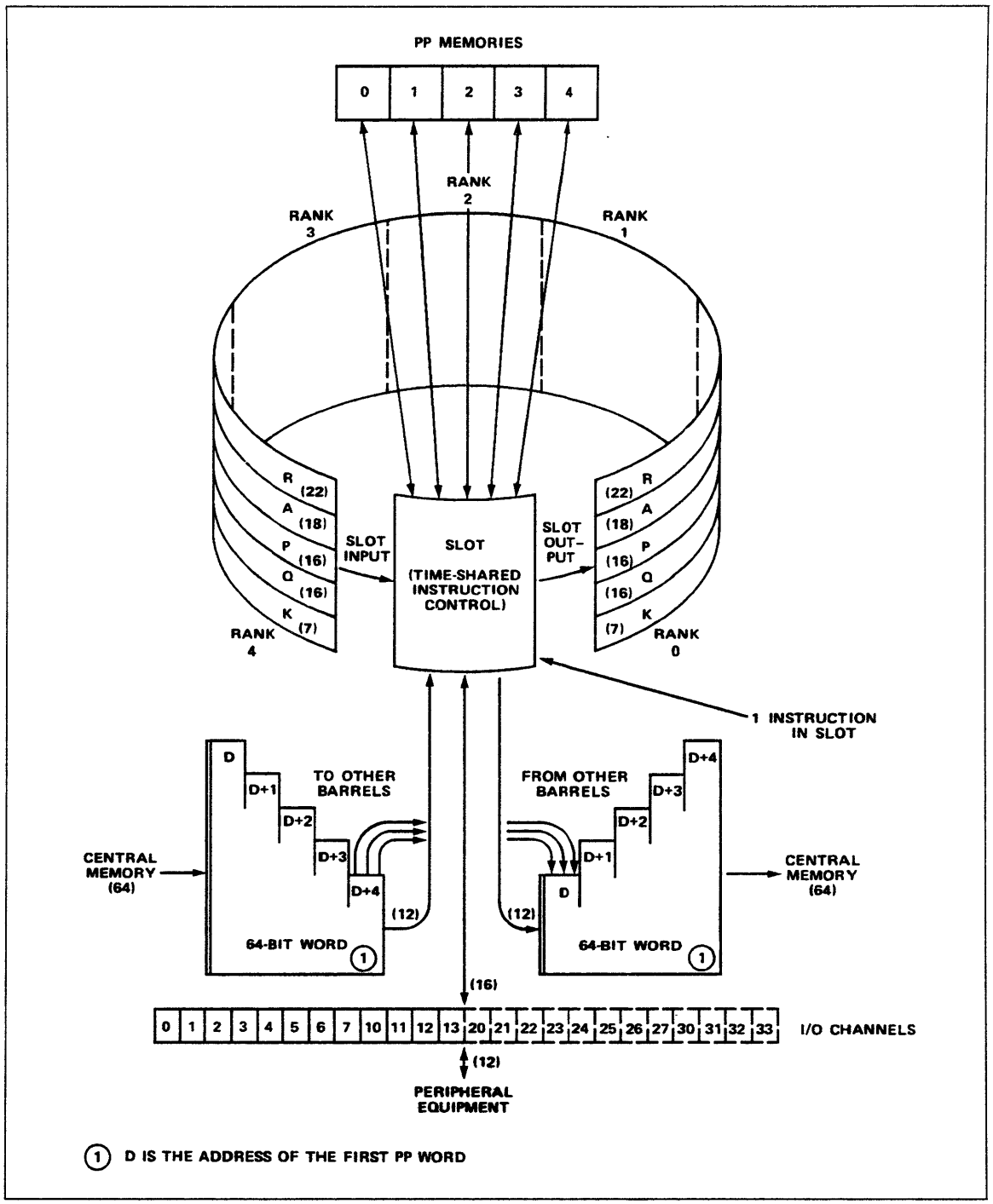


Figure 11-4. Barrel and Slot

PP Registers

The PP registers, which are discussed in the following paragraphs, are:

- R register
- A register
- P register
- Q register
- K register

R Register

The 22-bit R register, in conjunction with the A register, forms an absolute CM address for CM read/write instruction (refer to Central Memory Access by PPs later in this chapter).

A Register

The 18-bit A register contains one of two operands for arithmetic and logic operations. The content of A may be:

- Arithmetic operand
- A CM address or part of a CM address
- An I/O function
- An I/O data word
- Word count for a block I/O or CM transfer

Various instructions operate on 6, 12, 16, or 18 bits of the A register. Calculation results are always placed in the A register, although some instructions also write the result into PP memory.

When the A register provides the CM address, parity is generated with the address for transmission to memory control. When the A register provides data or function words for I/O activities, channel parity is always generated on 16 bits of A.

At deadstart, the A register is set to 10000_8 for the standard 20 PPs and to 20000_8 for the optional 10 PPs.

P Register

The P register operates in two different modes. In 4K mode, P is a 12-bit register; and in 8K mode, P is a 16-bit register. In 8K mode, the PP memory uses only the least significant 13 bits.

The P register is the PP program address counter. Also, during block I/O and CM transfers, the P register temporarily contains the PP memory address of the data transfer. At deadstart, the P register is set to 7777_8 for the NIO PPs and is set to 1_8 for the CIO PPs.

Q Register

The Q register operates in two different modes. In 4K mode, Q is a 12-bit register; and in 8K mode, Q is a 16-bit register. In 8K mode, the PP memory uses only the least significant 13 bits.

The Q register may hold the following data:

- Operand address for direct and indirect addressing.
- Peripheral address of data used during single-word CM read/write instructions.
- Shift count.
- Word count for CM block transfers.
- Upper six bits during constant mode PP instructions.
- Target address for relative jump.
- Channel number for all I/O and channel instructions.

At deadstart, each rank of the Q register is set to a corresponding PP number. Rank 0 is set to PP0, rank 2 is set to PP2, and so on.

K Register

The 7-bit K register is visible to the programmer through the maintenance channel and the IOU deadstart display. This register holds the operation code field of an instruction for display and is used for maintenance purposes. When a PP is halted (idled), this register contains all ones.

PP Numbering

PPs are numbered in octal as follows:

Barrel	PPs
0	N00 through N04
1	N05 through N11
2	N20 through N24
3	N25 through N31
0	C00 through C04
1	C05 through C11

The deadstart sequence decodes a program stored in the IOU microprocessor RAM to determine PP numbering within a barrel. The sequence:

- Assigns barrel numbers according to the program.
- Loads a zero into the Q register in barrel 0 during the first minor cycle after deadstart.

This defines all the data in that rank of the barrel as belonging to PP0 and, since Q is the channel selector, assigns PP0 to channel 0.

During the next minor cycle, Q loads with a one. This defines PP1 and assigns it to channel 1.

This process occurs in parallel in all barrels until the IOU assigns each rank of the barrel with a PP number and a channel number. Reassignment can only be done at deadstart. For further information on PP reassignments, refer to the CIP User's Handbook listed under Additional Related Manuals in About This Manual.

PP Memory

Each NIO PP has an independent 8K-word memory degradeable to 4K; each word contains 16 data bits and 6 bits of SECDED code. Each CIO PP has an independent 8K-word memory; each word contains 16 data bits and six bits of SECDED code.

PP0 reads the deadstart program from the microprocessor RAM during the deadstart operation. Therefore, PP memory 0 must be operational. A PP memory reconfiguration feature allows the user to restore IOU operation if the IOU detects a fault in the PP memory normally assigned to PP0.

To reconfigure, the operator assigns a good PP memory to PP0, and the operating system removes the failing PP memory. Computer operation can continue without the failing PP memory, and repairs can be made during scheduled maintenance. The system must be deadstarted to reconfigure PPs.

I/O Channels

The I/O channels are composed of:

- An internal interface that allows common hardware and software to control the external devices, and
- An external interface that allows the IOU to communicate with the external devices using 12-bit data channels. The internal interface can transfer 16-bit data words between two PPs or between a PP and an external device at a maximum rate of 1 word every 250 ns.

This rate can be sustained for the maximum practical channel transfer (4096 words). During transfers between PPs, if the PPs are in the slot at the same time, the transfer rate is 500 ns.

Any PP can access any of the CYBER 170 bidirectional I/O channels. All PPs communicate with external devices through the independent I/O channels. Each channel may be connected to one or more pieces of external equipment, but only one piece of equipment can use a channel at one time. All channels can be active simultaneously. Available channels are listed as follows:

- Twenty-four CYBER 170 compatible I/O channels or 10 CIO channels available with a maximum data transfer rate of 3 megabytes per second.
- An optional DMA-enhanced ISI channel adapter, IPI channel adapter, or CYBER 170 channel adapter that can be installed in any one of 10 channel locations in the CIO cabinet. The adapters transfer data between the ISI, IPI, or CYBER 170 channels and PP memory using standard I/O instructions. They also support DMA transfer in which data goes directly between CM and an external device without going through the PP. There are two types of CYBER 170 DMA transfers, fast and normal. Fast transfers are used with the ESM-II, and normal transfers are used with other CYBER 170 external devices.

Display Station Controller

The display station controller (DSC) is the IOU interface between the PPs and the display station servicing both the keyboard and the cathode-ray tube (CRT). The DSC transmits function words and digital symbol size/position data to the display station and receives digital character codes from the keyboard. It also receives digital symbol codes from the PPs and converts these to analog symbols to the CRT.

Real-Time Clock

The real-time clock is a 12-bit free-running counter, incrementing at a 1-MHz rate. It is permanently attached to channel 14_8. This channel may be read at any time since it is active and full flags are always set.

Two-Port Multiplexer

The two-port multiplexer provides communication capability between a PP and two attached terminals. It can simultaneously drive the two terminals at different baud rates. One port is reserved for maintenance, and the other port is reserved for future use. The two-port multiplexer is permanently attached to channel 15_8. Each port may drive a separate terminal.

For a dual-IOU option, both ports of the primary IOU two-port multiplexer are connected to the 1900X System Console (CC598-B). Only one port of the second IOU two-port multiplexer is connected to the same 1900X System Console, and the other port is not used.

Maintenance Channel

The maintenance channel (MCH) is used for initialization of the CP and CM maintenance registers and monitoring of error status.

The maintenance channel consists of the maintenance channel interface on channel 17_8; a maintenance access control (MAC) in the CP, CM, and IOU; and a set of interconnecting cables. The maintenance channel of the second IOU in a dual-IOU option has access only to the MAC in the IOU.

Any PP can be programmed to act as the maintenance control unit (MCU). However, hardware dictates PP0 as having special deadstart functions such that PP0 optimally serves as the MCU. In any case, the PP acting as the MCU performs initialization and maintenance functions that include:

- Initializing registers, controls, and memories.
- Monitoring and recording error information.
- Verifying error detection and correction hardware.

The MCU directs these operations by sending function words (instructions) over the maintenance channel to the CP, CM, and IOU. The MCU retains all normal PP capabilities and, except for PP0 deadstart functions, does not gain any special hardware capabilities.

A separate MCU in the second IOU in a dual-IOU option reports its error status to the primary IOU via central memory.

IOU Maintenance Registers

The MAC in the IOU contains several maintenance registers which hold IOU status or error information. Table 11-3 lists the IOU maintenance registers. For detailed descriptions of these registers, refer to IOU Registers in volume 2 of this manual (listed in About This Manual).

Table 11-3. IOU Maintenance Registers

Register Name	Number of Bits	Address	Access Type Copy	Access Type MCH
Element identifier (EID)	32	10	-	R
Environment control (EC)	32	30	-	R/W
Fault status 1 (FS-1)	64	80	-	R/W
Fault status 2 (FS-2)	64	81	-	R/W
Fault status mask (FSM)	64	18	-	R/W
Options installed (OI)	64	12	-	R
OS bounds (OSB)	64	21	-	R/W
Status summary (SS)	6	00	-	R
Test mode (TM)	16	A0	-	R/W

Element Identifier (EID) Register

The 32-bit EID register is a backpanel-wired register identifying each system hardware element. The EID bits are represented as follows:

Bits	Description
32 through 39	Element type
40 through 47	Model number
48 through 63	Serial number (hexadecimal)

Environment Control (EC) Register

The 64-bit EC register controls timing margins, test mode and deadstart, PP memory dumps, reconfiguration, and stop-on-error conditions for the IOU. It also selects PP internal registers for reading. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

Fault Status (FS) Registers

The 64-bit FS registers indicate the presence of uncorrectable faults in the IOU, PP memories, I/O channels, or PP hardware. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

Fault Status Mask Register

This 64-bit register controls IOU fault reporting to the IOU fault status (FS) registers. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

Options Installed (OI) Register

The 64-bit OI register identifies the options installed in the IOU. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

OS Bounds Register

The 64-bit operating system (OS) bounds register divides the CM into an upper and a lower region for system protection. The OS bounds register contains a bit for each PP which indicates the region in CM into which the specified PP may initiate exchange operations or writes. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

Status Summary Register

The status summary register indicates errors in the CP, CM and IOU. It also provides information about the PP-halt, error status, and physical environment conditions. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

Test Mode (TM) Register

The 64-bit TM register forces faults in the IOU for testing of the fault sensing logic. Bits 48 through 63 of this register serve a dual role. With the Enable Test Mode Register bit set in the EC register, these bits are used to force test conditions (refer to the Maintenance Register Codes Booklet listed in About This Manual for further information). When the Enable Test Mode Register bit is clear, these read/write bits can be used by software as interlock/flag status bits.

Central Memory Access by PPs

Any PP can access CM. During a write from the IOU to CM, the IOU assembles either four successive 16-bit PP words into one 64-bit CM word (Virtual State) or five successive 12-bit PP words into one 60-bit CM word (CYBER 170 State).

During a CM read, the IOU disassembles either a 64-bit CM word into four 16-bit PP words (Virtual State), or a 60-bit CM word into five 12-bit PP words (CYBER 170 State).

To find the CM address, a PP reads the A register. If bit 17 of the A register is clear, the PP uses the contents of the A register for the CM address. If bit 17 of the A register is set, the PP adds the relocation address from the R register to the A register for the CM address.

A maximum of 30 PPs can simultaneously read CM words, and 30 PPs can write CM words.

Model 990 Operating Instructions 12

Controls and Indicators 12-1
Deadstart Displays/Controls 12-1
Central Memory Controls 12-4

Power-On and Power-Off Procedures 12-6

Operating Procedures 12-6
Control Checks 12-6
Deadstart Sequences 12-6

IOU Reconfiguration 12-7

Model 990 Operating Instructions 12

This chapter describes mainframe controls and indicators and the operating procedures which are hardware dependent. Software-dependent procedures are in system software reference manuals listed under Additional Related Manuals in About This Manual.

Controls and Indicators

This section describes IOU deadstart controls and indicators and CM configuration switches used by the system operator. Other controls used by maintenance personnel are described in the hardware operator's guide and the hardware maintenance manuals of the power distribution and warning system, the cooling system, and the system console listed in the system publication index in About This Manual.

Deadstart Displays/Controls

Pressing the DEADSTART pushbutton on the CC545 system console, or pressing the CTRL-G, CTRL-R, then M key on the CC634-B system console or pressing either CTRL-G or CTRL-F2, then the M key on the CC598-B system console initiates deadstart, and an initial deadstart display appears on the screen of the system console. It is created by an independent microcomputer in the mainframe and does not rely on any program being operational in the PPs. The initial deadstart display is used to select a 16-word deadstart program for PP0 and to initiate the deadstart sequence for PP0. It is also used to reconfigure PPMs and barrels, and to display error status and maintenance information.

The format of the deadstart options display is shown in figure 12-1, and the deadstart display is shown in figure 12-2. Table 12-1 describes the two operator-selectable options and table 12-2 describes the operator entries and functions for the deadstart display. Other deadstart displays are available for maintenance use. Refer to the CYBER Initialization Package (CIP) Reference Manual listed in About This Manual for additional information.

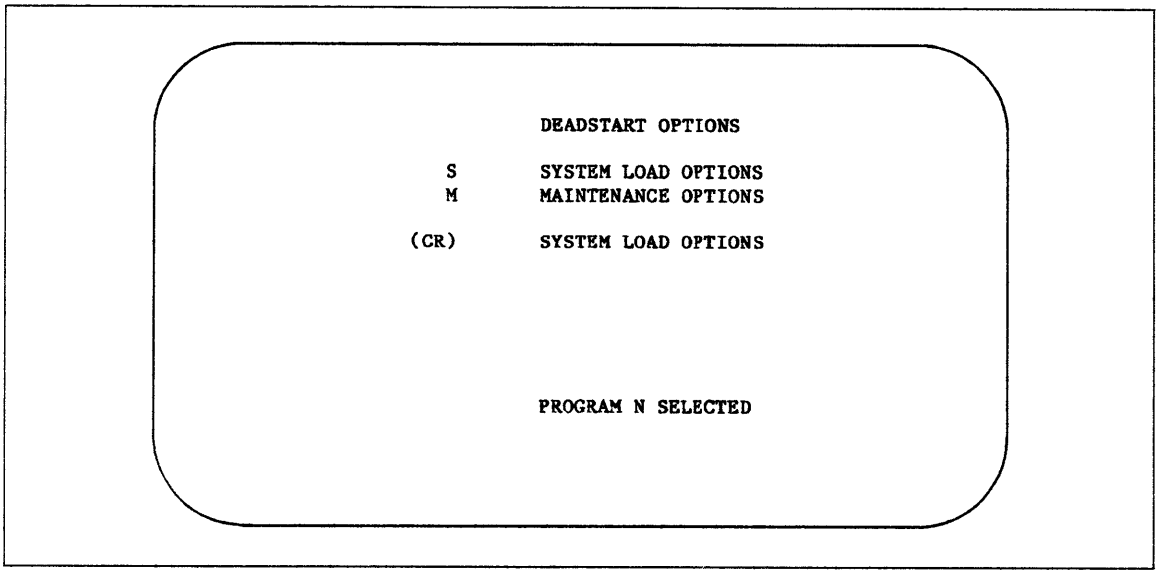


Figure 12-1. Deadstart Options Display

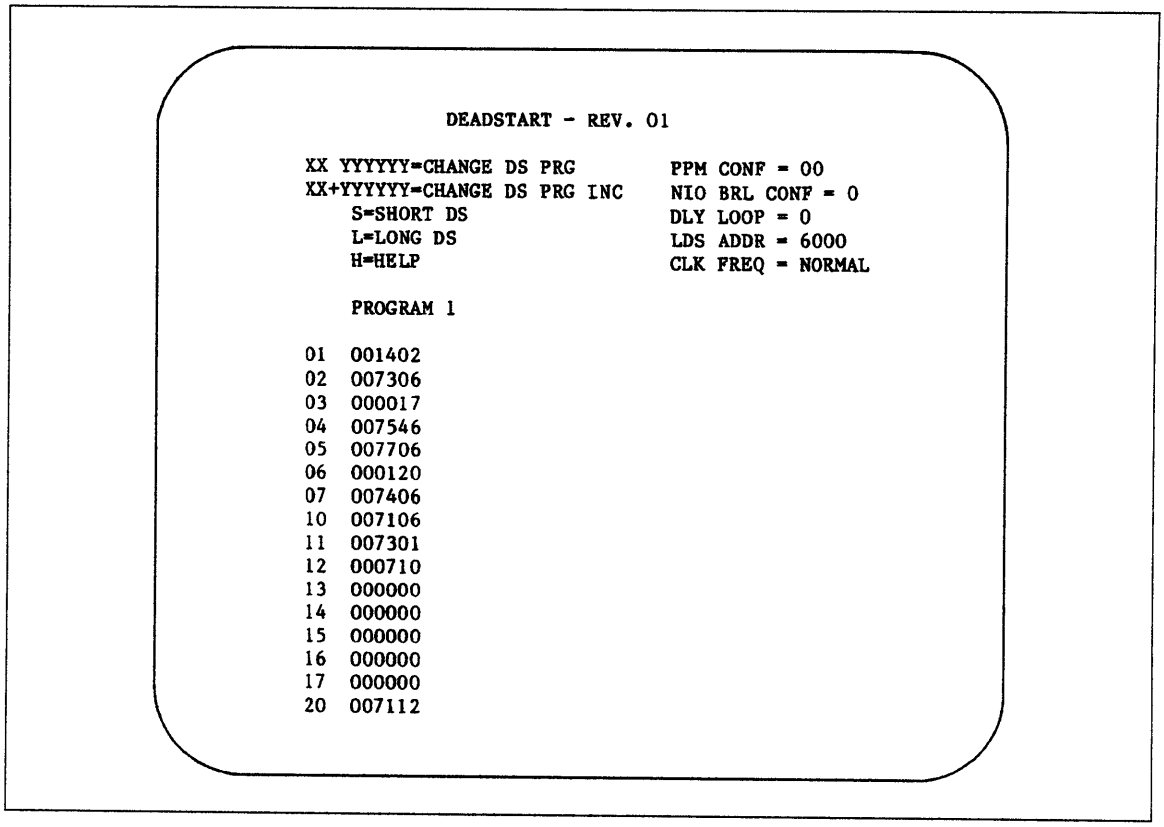


Figure 12-2. Initial Deadstart Display

Table 12-1. Deadstart Options Display

Option	Description
S	Selects a short deadstart sequence using the deadstart program identified at the bottom of the display. Upon completion of the deadstart sequence, a display for loading system software appears.
M	Causes the deadstart display to appear on the screen.

Table 12-2. Deadstart Display Operator Entries and Functions

Operator Entry	Function
xx yyyyyy	Enters a single word in the deadstart program at xx to a new value yyyyyy (octal).
xx+yyyyyy	Changes words in the deadstart program in sequence starting at xx.
S	Selects a short deadstart sequence.
L	Selects a long deadstart sequence.
H	Brings up a display that lists and explains all available commands. Refer to the hardware operator's guide for detailed information about these commands.

Central Memory Controls

The CM contains a three-position configuration switch (figure 12-3). The switch is located on a switch box attached to the underside of the top panel over the CMC section.

Each switch, SW2 through SW4, forces one corresponding CM address bit, 39 through 41, either to a zero (switch down) or to a one (switch up). Refer to table 12-3.

In case of CM malfunctions, the remaining good memory can be reconfigured so it is accessible by contiguous addresses from 0 to the maximum remaining address. This is accomplished by setting configuration switches (figure 12-3) as listed in table 12-3. Refer to the hardware operator's guide listed in the system publication index for further information.

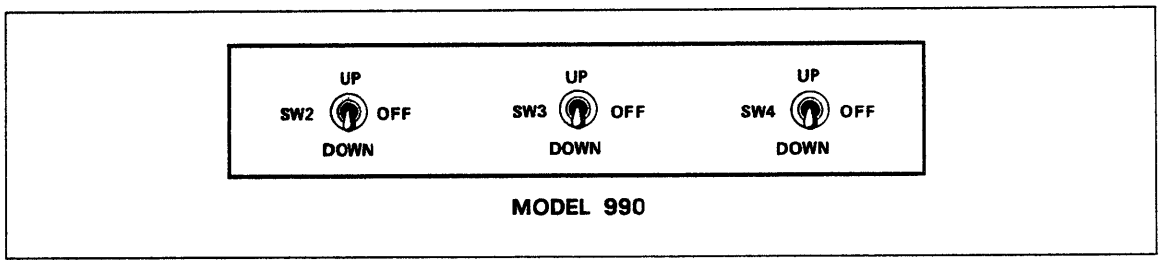


Figure 12-3. CM Configuration Switches

Table 12-3. Central Memory Reconfiguration

Original CM		Reconfigured CM						
			Location of Failing CM[1]			Reconfiguration Setting[2]		
Words (Size)	Address Range	Words (Size)	RMA Bit 39	RMA Bit 40	RMA Bit 41	SW2	SW3	SW4
1049K (8 MB)	0-3 777 777	524K (4 MB)	X	X	0	-	-	U
		524K (4 MB)	X	X	1	-	-	D
2097K (16 MB)	0-7 777 777	1049K (8 MB)	X	0	X	-	U	-
		1049K (8 MB)	X	1	X	-	D	-
3146K (24 MB)	0-13 777 777	1049K (8 MB)	0	0	X	-	U	-
		1049K (8 MB)	0	1	X	-	D	-
		2097K (16 MB)	1	0	X	D	-	-
4195K (32 MB)	0-17 777 777	2097K (16 MB)	0	X	X	U	-	-
		2097K (16 MB)	1	X	X	D	-	-

1. CM remaining can be further reconfigured by setting additional configuration switches.

2. U equals up, D equals down, and dash (-) equals center position.

Power-On and Power-Off Procedures

In case of an emergency, use the system EMERGENCY OFF switch. The power-on and power-off procedures are described in the hardware operator's guide listed in the system publication index.

CAUTION

Inproper application or removal of power may damage system circuits and/or air conditioning system. Power must be turned on/off by designated personnel only, except for the system EMERGENCY OFF switch. Use only for extreme emergency, not for normal shutdown.

Operating Procedures

Refer to the hardware operator's guide listed in the system publication index. The system is initialized by setting its deadstart display control parameters, and then by running either a long or short deadstart sequence (defined later in this section). After initialization, the keyboard is used to instruct the system further, under program control.

Control Checks

Before activating a long or short deadstart sequence, check the deadstart display parameters against their intended use. The normal settings of these parameters are as follows:

Parameter	Value
PPM CONF	00
BRL CONF	0
LDS ADDR	6000
Error messages	None

Deadstart Sequences

In response to a keyboard command (L or S) to the deadstart display, the IOU performs a deadstart sequence. Depending on the command (L or S), either the long or the short deadstart sequence is performed. The short deadstart sequence is used when hardware integrity verification is not required. The long deadstart sequence performs all the tasks performed by the short deadstart sequence and some additional tasks. The main additional task is the running of a diagnostic program, from a read-only memory (ROM) in the IOU, on logical PP0. The diagnostic program takes approximately 15 seconds to run.

Both deadstart sequences begin with a master clear which sets up all PPs, except logical PP0, for a 4096-word block input starting at PP location 0. The input into each PP is from the channel with the same number as the logical number of the PP concerned. The master clear also resets all external devices and sets maintenance channel connect code bit 52. The individual registers are set as follows.

Register	Initialization	Description
K	007100_8	Instruction display
P	007777_8	Causes block input to start from location 0
A	$10{,}000_8$	Count of 4096 words
Q	0, 1, 2...	I/O channel numbers (PP0: 0, PP1: 1, and so on)

All registers in both barrels are set to these values, except the registers of PP0.

If the long deadstart sequence is being performed, hardware clears location 7777_8 in all PP memories and sets the P register of PP0 to the value indicated by the parameter LDS ADDR = XXXX (normally 6000_8). PP0 starts performing a test program from a read-only memory in IOU. Hardware errors cause the LDS program to hang before completion. In the absence of errors, execution proceeds until the test program reaches location 7776_8. When this happens, the unique part of the long deadstart sequence ends with a master clear.

Next both deadstart sequences clear PP0 location 0, write the deadstart program on the display into PP0 memory locations 1 to 20_8, and clears PP0 location 21_8. PP0 then starts executing the program entered from the deadstart display (which is normally a bootstrap program to input more data from an assigned external device).

The short deadstart sequence does not disturb PP memory other than PP0 locations 0 to 21_8. Both deadstart sequences leave all PPs, except PP0, waiting for a block input or for action through the maintenance channel. After the block input is complete, each PP starts executing the program entered from whatever address was entered into location 0 of that PP.

IOU Reconfiguration

NOTE

Only PPs can be reconfigured in an AT511-A/AT512-A IOU. Ignore all mention of the RB X command and the BRL CONF parameter in the following discussion if the IOU to be reconfigured has this equipment number(s). In tables 12-4 and 12-5, only the RB=0 examples are applicable.

The logical PP numbers and hardware are assigned to physical PPs circularly from the settings of IOU deadstart display PPM CONF and BRL CONF parameters, specifying which physical barrel and PPM is PP0. Maximum values for these parameters depend on the number of PPs installed. Illegal values entered in RB X and RP XX commands are rejected by the deadstart display and cause error messages to appear on the screen (refer to the hardware operator's guide). Reconfiguration is discussed in detail in the hardware operator's guide; allowable values for the PPM CONF and BRL CONF parameters and reconfiguration examples are shown in tables 12-4 and 12-5.

Table 12-4. PP and Barrel Reconfiguration Example, RP=0

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=0					
10	00	00	05		
	01	01	06		
	02	02	07	X	X
	03	03	10		
	04	04	11		
15	00	00	05	20	
	01	01	06	21	
	02	02	07	22	X
	03	03	10	23	
	04	04	11	24	
20	00	00	05	20	25
	01	01	06	21	26
	02	02	07	22	27
	03	03	10	23	30
	04	04	11	24	31

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 12-4. PP and Barrel Reconfiguration Example, RP=0 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=1					
10	00	05	00		
	01	06	01		
	02	07	02	X	X
	03	10	03		
	04	11	04		
15	00	20	00	05	
	01	21	01	06	
	02	22	02	07	X
	03	23	03	10	
	04	24	04	11	
20	00	25	00	05	20
	01	26	01	06	21
	02	27	02	07	22
	03	30	03	10	23
	04	31	04	11	24

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 12-4. PP and Barrel Reconfiguration Example, RP=0 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=2					
10	00				
	01				
	02	X	X	X	X
	03				
	04				
15	00	05	20	00	
	01	06	21	01	
	02	07	22	02	X
	03	10	23	03	
	04	11	24	04	
20	00	20	25	00	050
	01	21	26	01	061
	02	22	27	02	072
	03	23	30	03	103
	04	24	31	04	114

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 12-4. PP and Barrel Reconfiguration Example, RP=0 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=3					
10	00				
	01				
	02	X	X	X	X
	03				
	04				
15	00				
	01				
	02	X	X	X	X
	03				
	04				
20	00	05	20	25	00
	01	06	21	26	01
	02	07	22	27	02
	03	10	23	30	03
	04	11	24	31	04

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

Table 12-5. PP and Barrel Reconfiguration Example, RP=2

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=0					
10	00	03	10		
	01	04	11		
	02	00	05	X	X
	03	01	06		
	04	02	07		
15	00	03	10	23	
	01	04	11	24	
	02	00	05	20	X
	03	01	06	21	
	04	02	07	22	
20	00	03	10	23	30
	01	04	11	24	31
	02	00	05	20	25
	03	01	06	21	26
	04	02	07	22	27

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 12-5. PP and Barrel Reconfiguration Example, RP=2 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=1					
10	00	10	03		
	01	11	04		
	02	05	00	X	X
	03	06	01		
	04	07	02		
15	00	23	03	10	
	01	24	04	11	
	02	20	00	05	X
	03	21	01	06	
	04	22	02	07	
20	00	30	03	10	23
	01	31	04	11	24
	02	25	00	05	20
	03	26	01	06	21
	04	27	02	07	22

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 12-5. PP and Barrel Reconfiguration Example, RP=2 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=2					
10	00				
	01				
	02	X	X	X	X
	03				
	04				
15	00	10	23	03	
	01	11	24	04	
	02	05	20	00	X
	03	06	21	01	
	04	07	22	02	
20	00	23	30	03	10
	01	24	31	04	11
	02	20	25	00	05
	03	21	26	01	06
	04	22	27	02	07

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 12-5. PP and Barrel Reconfiguration Example, RP=2 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=3					
10	00				
	01				
	02	X	X	X	X
	03				
	04				
15	00				
	01				
	02	X	X	X	X
	03				
	04				
20	00	10	23	30	03
	01	11	24	31	04
	02	05	20	25	00
	03	06	21	26	01
	04	07	22	27	02

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

CYBER 990E, 995E, and 994 System Description 13

Physical Characteristics 13-1

Functional Characteristics 13-4
- Central Processor 13-5
- Central Memory (CYBER 990E and 995E) 13-6
- Central Memory (CYBER 994) 13-6.1
- Input/Output Unit 13-7

Major System Component Descriptions 13-8
- Central Processor 13-8
  - Instruction Section 13-8
  - Registers 13-9
  - Execution Section 13-9
  - Cache Memory 13-9
  - Addressing Section 13-9
  - Central Memory Control 13-9
- Central Memory 13-10
- Input/Output Unit 13-11
- System Console 13-12

CYBER 990E, 995E, and 994 System Description 13

This chapter describes the physical and functional characteristics and major system components.

These high-speed computer systems are used for both business and scientific applications. The system includes the following components.

- Central processor (CP)
- Central memory (CM)
- Input/output unit (IOU)

Physical Characteristics

The mainframe configuration (figure 13-1) includes interconnected CP, CM, and IOU cabinet sections that compose the system cabinet. (The system console is also required for system operation.) Each cabinet section contains a logic chassis with plug-in circuit boards. The CP consists of 10 sections, plus a single section for central memory control (CMC). With the CYBER 995E dual-CP, an additional identical cabinet contains the second CP. If a second CP is installed in a CYBER 990E, it then becomes a CYBER 995E. (The second CP does not support CYBER 170 State operation.) If a second CP is installed on a CYBER 994, it is defined as a CYBER 994 with a dual-CP. A CYBER 995E always contains two CPs (dual-CP).

The basic metal-oxide semiconductor (MOS) CM consists of one section with four distributors. The single- and dual-CP systems both support an additional CM section as an option (CYBER 990E and 995E only). The IOU consists of two sections (CIO section optional on CYBER 994) with initialization and maintenance controls and displays. A second IOU is installed in a separate cabinet for a dual-IOU option. This secondary IOU provides the same number and types of channels as the primary IOU.

Computer System	No. of CPs	Type of CM
CYBER 990E	1	MOS (64K circuits)
CYBER 995E	2	MOS (64K circuits)
CYBER 994	1 or 2	MOS (256K circuits)

The CP, CM, and IOU sections each contain an ac/dc control section with voltage margin testing facilities and dc power supplies. A standalone cooling unit provides cooling for the CP, CM, and IOU logic chassis, except IOU chassis for CYBER 994 is air-cooled.

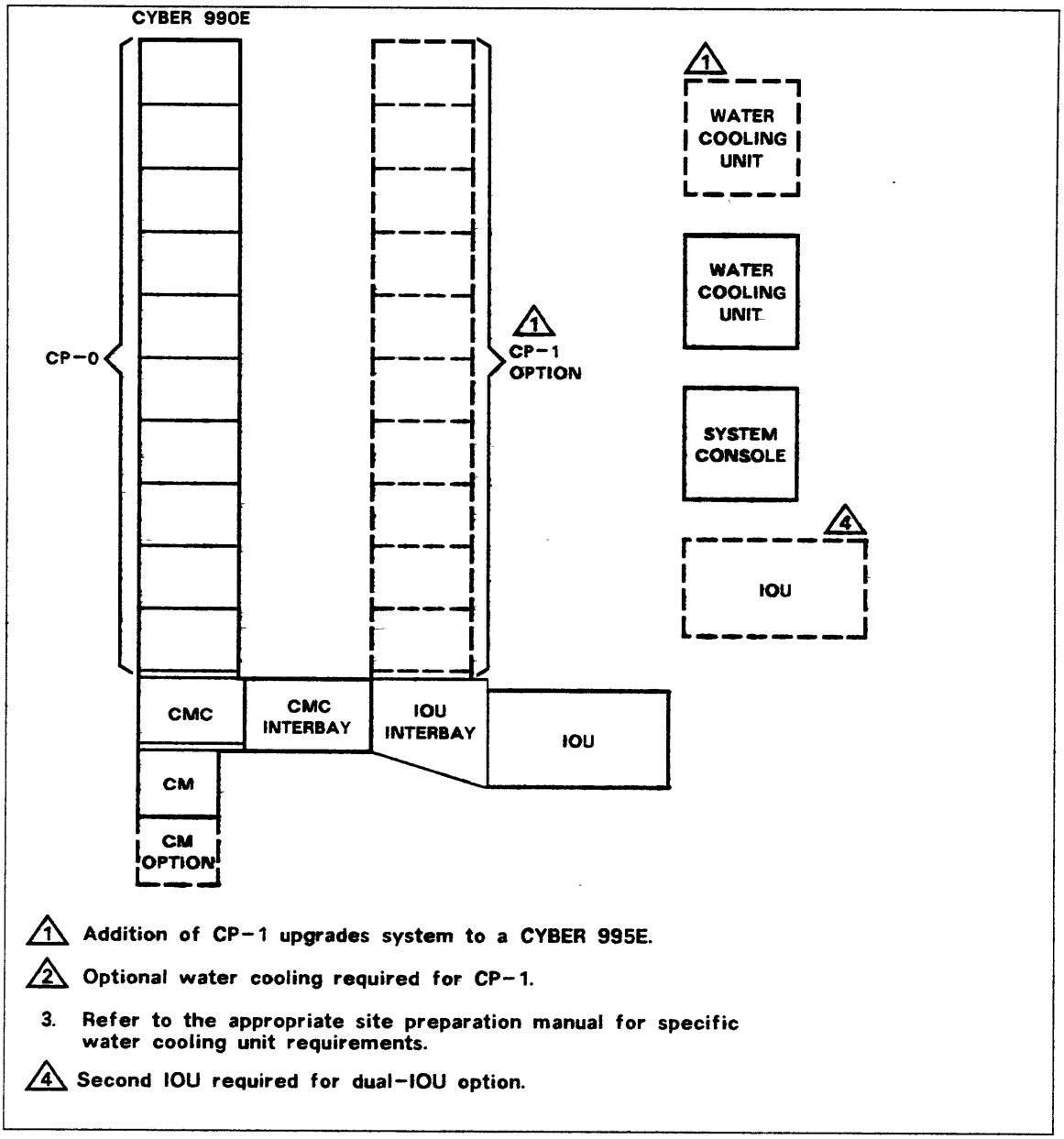


Figure 13-1. System Configuration

(Continued)

(Continued)

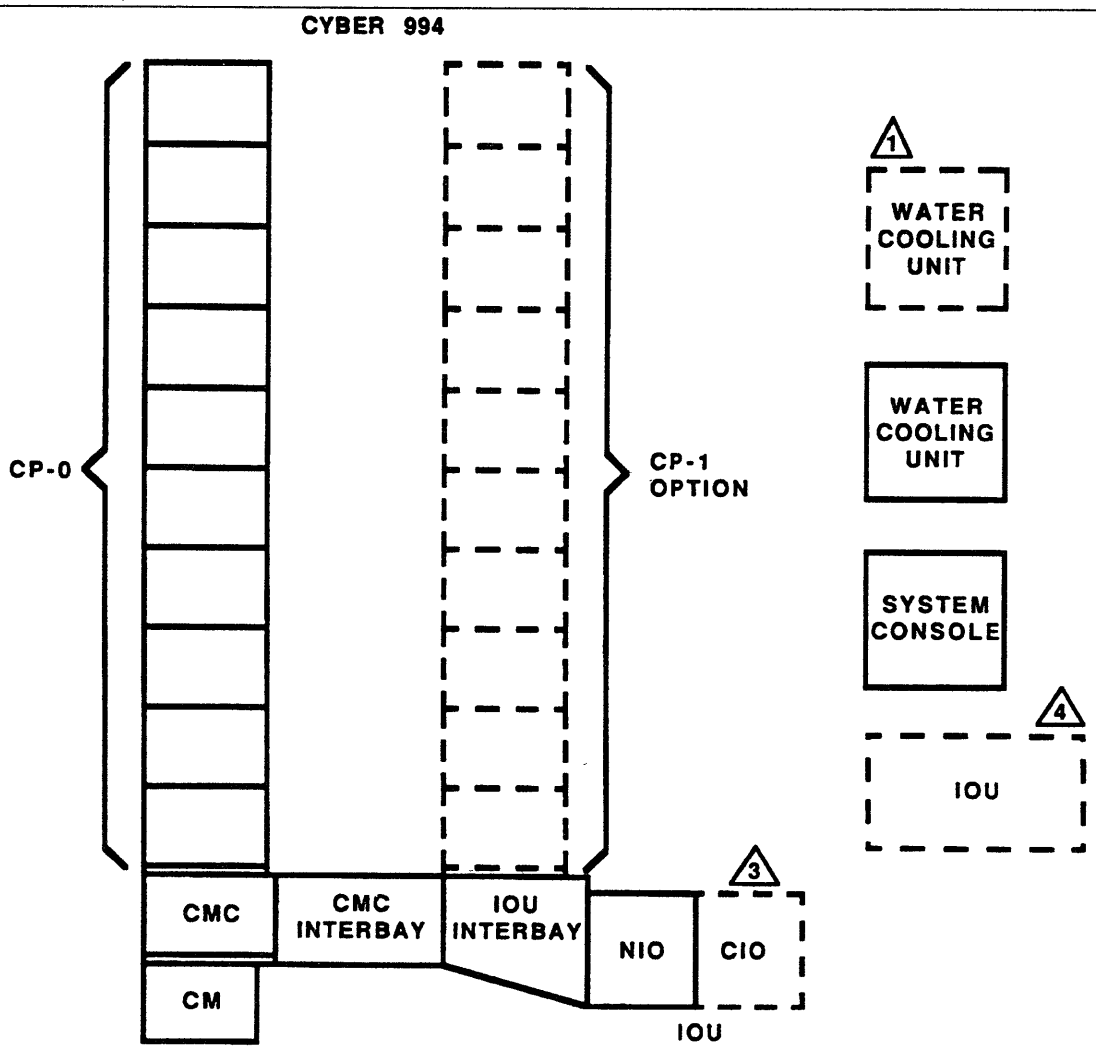
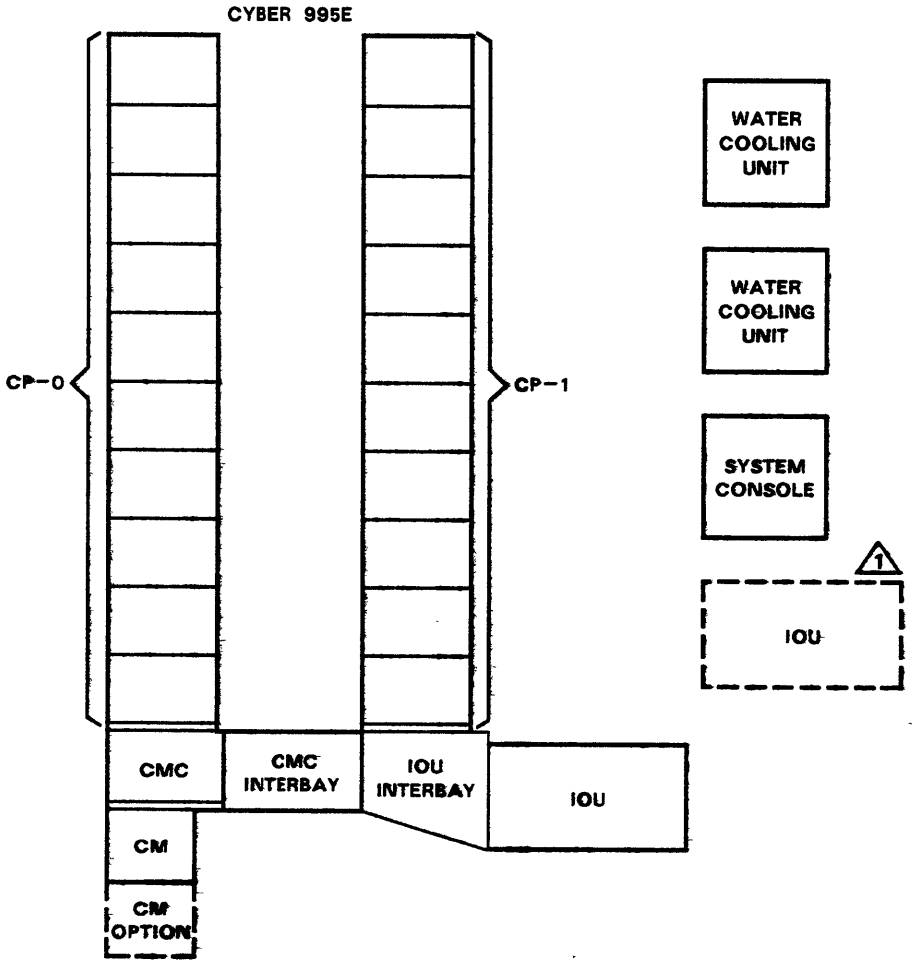


Figure 13-1. System Configuration

(Continued)

(Continued)



1 Second IOU required for dual-IOU option.

2. Refer to the appropriate site preparation manual for specific water cooling unit requirements.

Figure 13-1. System Configuration

Functional Characteristics

Emitter-coupled logic (ECL) and large-scale integration (LSI) logic is used to achieve high computation speeds. High speed is also the objective of the CP design, which is based on the capability to execute many operations concurrently.

The CP supports two states of operation.

Virtual State	Operates with virtual-memory byte addressing, using the Virtual State instruction set and data formats. Virtual State is the native operating state of the CP.
CYBER 170 State	Operates with real-memory word addressing, using the CYBER 170 State CP instruction set and data formats.

The Virtual State and CYBER 170 State environments may be present at the same time with the CP executing in either environment.

- NOS/VE is the operating system of Virtual State.
- NOS is the operating system of CYBER 170 State.

The MOS memory is divided into eight independent banks. System input/output speeds are determined by the capabilities of existing external devices.

The CYBER 990E contains a single central processor (CP-0) capable of dual-state operations. The dual states are: CYBER 170 State and Virtual State.

The CYBER 995E contains dual central processors (CP-0 and CP-1). CP-0 is capable of dual-state operation. CP-1 is capable of only Virtual State operation.

The CYBER 994 can contain up to two central processors (CP-0 and CP-1). CP-0 is capable of dual-state operation. Optional CP-1 is capable of only Virtual State operation.

Central Processor

The CP has the following characteristics:

- 60-bit internal word.
- Eight 60-bit operand (X) registers.
- Eight 18-bit address (A) registers.
- Eight 18-bit index (B) registers.
- Two registers that isolate each user's extended memory space (RAC, FLC).
- Two registers that isolate each user's extended memory space (RAE, FLE).
- Register exchange instructions (exchange jumps) for interrupting programs.
- Floating-point (FP) arithmetic (10-bit exponent plus sign bit, 48-bit coefficient plus sign bit). Some FP instructions use 96-bit (double precision) coefficients.
- Integer arithmetic (60/18-bit operands).
- Character string compare/move facilities (6-bit characters).
- Packed instructions (15/30/60-bit instructions in 60-bit words).
- Synchronous internal logic.
- 16-nanosecond clock period.
- 4096-word cache buffer memory.
- Instruction and branch instruction lookahead.
- Microcode control.
- Parity checking of selected data and address paths.
- Maintenance channel to IOU.

Central Memory (CYBER 990E and 995E)

The CM has the following characteristics:

- Each column (cabinet) contains four distributors and contains a minimum of 2097K words (16 MB), MOS memory; options available to 16M words (128 MB) in 2097K word increments.
- Each distributor has a 72-bit data word, consisting of 64 data bits and 8 single-error correction/double-error detection (SECDED) bits. (There are an additional 8 unused bits.)
- Organization of eight independent banks.
- Memory ports located in the CMC cabinet.
- 16-nanosecond clock period.
- Maximum data transfer rate of one word every 16 nanoseconds from each of the four CMC distributors.
- 113-nanosecond read access time.
- 96-nanosecond read/write cycle time.
- 192-nanosecond partial-write cycle time.
- Read and write data queuing capability in CMC.
- Parity checking of address paths.
- Bounds register to limit write access.
- Unified-extended memory (UEM), an extended memory within CM.

Central Memory (CYBER 994)

The CM has the following characteristics:

- 8,388K (8M) words (64 MB), MOS memory; options available to 32M words (256 MB) in 8,388K (8M) word increments.
- Each distributor has a 72-bit data word, consisting of 64 data bits and 8 SECDED bits. (There are an additional 8 unused bits.)
- Organization of eight independent banks.
- Memory ports located in the CMC cabinet.
- 16-nanosecond clock period.
- Maximum data transfer rate of one word every 16 nanoseconds from each of the four CMC distributors.
- 113-nanosecond read access time.
- 96-nanosecond read/write cycle time.
- 192-nanosecond partial-write cycle time.
- Read and write data queuing capability in CMC.
- Parity checking of address paths.
- Bounds register to limit write access.
- Unified-extended memory (UEM), an extended memory within CM.

Input/Output Unit

The IOU has the following characteristics:

- A nonconcurrent input/output (NIO) subsystem consisting of 20 CYBER 170 compatible peripheral processors (PPs). Each PP has an 8K x 16-bit word independent memory (PPM) degradeable to 4K x 16 bits (refer to the CYBER Initialization Package (CIP) Reference Manual listed under Additional Related Manuals in About This Manual for information on degrading PPMs).
- A concurrent input/output (CIO) subsystem consisting of five or ten PPs (CYBER 990E and 995E) and none, five, or ten PPs (CYBER 994). Each PP has an 8K x 16-bit PPM. In addition, there is an optional direct-memory access (DMA)-enhanced intelligent standard interface (ISI) channel adapter, an optional intelligent peripheral interface (IPI) channel adapter, and an optional CYBER 170 DMA channel adapter. These adapters support DMA transfer between CM and an external device as well as standard I/O data transfer. An adapter can be installed in any one of 10 channel locations in the CIO cabinet. There are two types of CYBER 170 DMA transfers, fast and normal. Fast transfers are used with the Extended Semiconductor Memory-II (ESM-II), and normal transfers are used with other CYBER 170 external devices.
- Execution of 12- or 16-bit PP code.
- Port to CM.
- Twenty-four CYBER 170 compatible I/O channels available with a maximum data transfer rate of 3 megabytes/second.
- Interface to real-time clock, display controller, and two-port multiplexer.
- Bounds register controlling write access to CM.
- SECDED data verification on all PP memories.
- Parity checking on all major data and address paths.
- Maintenance channels giving PPs access to CP, CM, and IOU registers to perform system initialization and maintenance functions. The maintenance channel of second IOU in a dual-IOU option has access to only IOU registers.
- Operating speed of 250 nanoseconds and a minor cycle of 50 nanoseconds.

Major System Component Descriptions

The following are the major system components:

- Central processor (CP)
- Central memory (CM)
- Input/output unit (IOU)
- System console

Brief descriptions of these components are provided in the remainder of this chapter. The system block diagram is shown on figure 13-2.

Central Processor

The CP hardware (figure 13-2) consists of the following:

- Instruction section
- Registers
- Execution section
- Cache memory
- Addressing section
- Central memory control

Central Processor 0 (CP-0) is capable of dual-state operation: Virtual State and CYBER 170 State. If a second central processor is added (CP-1), it is capable of only Virtual State operation.

The CP is isolated from the IOU and is thus able to carry on computation or character manipulation unencumbered by I/O requirements.

Instruction Section

The instruction section directs the arithmetic and manipulative functions for instruction execution. The instruction section prefetches instruction words from memory and disassembles them into instructions.

Registers

Operating registers reduce storage accesses for operands used during the execution of an instruction. These registers are:

- Eight 60-bit X registers (X0 through X7) which hold operands used for computation.
- Eight 18-bit A registers (A0 through A7) which use A0 primarily for indexing and A1 through A7 for CM operand addressing.
- Eight 18-bit B registers (B0 through B7) which are primarily indexing registers to control program execution. The B0 register always contains all zeros.

Eight support registers support the operating registers during program execution. These registers are:

- 18-bit program address (P) register.
- 21-bit reference address for CM (RAC) register. This is a program's lower bound.
- 21-bit field length for CM (FLC) register. This is a program's upper bound.
- 6-bit exit mode (EM) register.
- 6-bit flag register.
- 21-bit reference address for UEM (RAE) register.
- 24-bit field length for UEM (FLE) register.
- 18-bit monitor address (MA) register.

The registers store data and control information, present operands to the execution section, and store results.

The operating and support registers reside in the register unit and process state registers section.

Execution Section

The execution section combines the operands to achieve the result.

Cache Memory

The cache memory consists of 4096 words. The memory addressing sections determine whether a requested word is in the cache memory. If it is not, they read four consecutive words from central memory into the cache memory.

Addressing Section

The addressing section checks memory addresses against the CP registers RAC, FLC, RAE, and FLE to ensure isolation of user memory space.

Central Memory Control

CMC is integrated within the CP and controls the flow of data between CM and requesting system components.

Central Memory

The CM (figure 13-2) consists of the following:

- Eight memory banks
- Memory ports

The CM is a static MOS memory organized into eight independent banks. The distributor is located in the CMC and is used for bank phasing. The CYBER 990E and 995E use 64K chips, and CYBER 994 uses 256K chips.

A portion of CM can be reserved for use as extended memory. It is UEM and is referenced by the RAE and FLE registers. The UEM operates in 24-bit standard addressing mode. All memory ports have queuing buffers. The memory ports are located in the central processor cabinet.

The following CM option used on CYBER 990E and 995E allows selection of seven memory size increments.

Option	Model	Description
18850	32	Adds 2M 64K-words to increase size from 2M to 4M words (16 MB to 32 MB)
	48	Adds 2M 64K-words to increase size from 4M to 6M words (32 MB to 48 MB)
	64	Adds 2M 64K-words to increase size from 6M to 8M words (48 MB to 64 MB)
	80	Adds 2M 64K-words to increase size from 8M to 10M words (64 MB to 80 MB). Second cabinet required.
	96	Adds 2M 64K-words to increase size from 10M to 12M words (80 MB to 96 MB)
	112	Adds 2M 64K-words to increase size from 12M to 14M words (96 MB to 112 MB)
	128	Adds 2M 64K-words to increase size from 14M to 16M words (112 MB to 128 MB)

The following option is available to replace all existing 64K chips with 256K chips in a CYBER 990E or 995E.

Option	Model	Description
19712	64	Adds 8M 64K-words (64 MB)

The following option is available to upgrade a memory using 256K chips in a CYBER 990E, 995E, or 994.

Option	Model	Description
19701	128	Adds 8M 256K-words to increase size from 8M to 16M words (64 MB to 128 MB).
	192	Adds 8M 256K-words to increase size from 16M to 24M words (128 MB to 192 MB).
	256	Adds 8M 256K-words to increase size from 24M to 32M words (192 MB to 256 MB).

Input/Output Unit

The IOU (figure 13-2) consists of:

- Twenty logically independent NIO PPs. Options are available to increase the total to 25 or 30 PPs.
- None (cabinet is an option on CYBER 994), five, or ten optional logically independent CIO PPs and DMA channel adapters. Internal interface exists to 24 I/O channels. Options are available to increase the total to 34 channels.
- External interfaces to I/O channels:
  - 11 or 23 CYBER 170 channel interfaces.
  - Display controller interface (CYBER 170 channel 10_8).
  - Real-time clock interface (channel 14_8).
  - Two-port multiplexer interface (channel 15_8).
  - Maintenance channel interface (channel 17_8).
- Interface to CM.
- Bounds register to limit writes and exchanges to CM.

  The PPs are organized in groups of five, called barrels. The PPs in a barrel time-share comon hardware. Each PP has its own 4K or 8K independent memory and communicates with all I/O channels and with CM.

System Console

The system console, required for system operation, provides a visual, alphanumeric readout for the computer. The receipt of symbol and position information from the computer enables displaying program information on a video display terminal (VDT). The station also contains an alphanumeric keyboard which enables an operator to send data to the computer. The keyboard and CRT combination permits the computer operator to monitor and control system operation. Except for programming information in chapter 17, refer to the system console manual listed in About This Manual for further information.

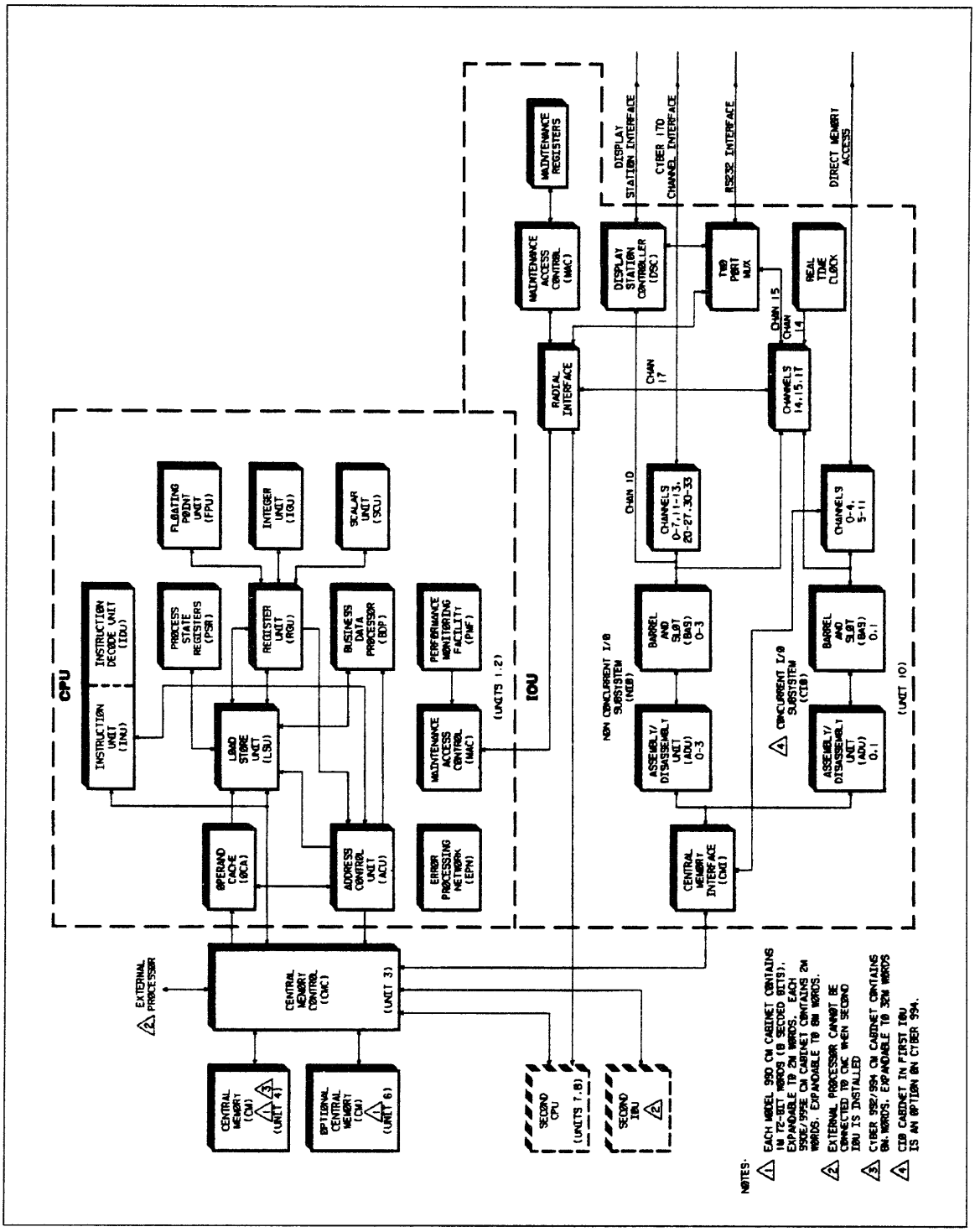


Figure 13-2. Computer System Block Diagram

CYBER 990E, 995E, and 994 Functional Descriptions 14

Central Processor 14-1
Instruction Section 14-1
Instruction Lookahead 14-1
Maintenance Access Control 14-1
Instruction Control Sequences 14-2
Boolean Sequence 14-2
Shift Sequence 14-3
Floating-Add Sequence 14-4
Floating-Multiply and Floating-Divide Sequence 14-4
Increment Sequence 14-4
Compare/Move Sequence 14-6
CYBER 170 Exchange Sequence 14-7
Block Copy Sequence 14-7
Direct Read/Write Sequence 14-7
Normal Jump Sequence 14-8
Return Jump Sequence 14-8
Registers 14-9
Operating Registers 14-10
X Registers 14-10
A Registers 14-10
B Registers 14-10
Support Registers 14-11
P Register 14-11
RAC Register 14-11
FLC Register 14-11
EM Register 14-12
Flag Register 14-12
RAE Register 14-13
FLE Register 14-13
MA Register 14-13
Execution Section 14-14
Cache Memory 14-14
Addressing Section 14-14
Central Memory Control 14-14

Central Memory 14-15
Address Format (CYBER 990E and 995E) 14-15
Address Format (CYBER 994) 14-16
CM Access and Cycle Times 14-16.1
CM Ports and Priorities 14-16.1
SECDED 14-17
CM Layout 14-22
CM Bounds Register 14-22
Central Memory Reconfiguration 14-22

Input/Output Unit 14-23
Peripheral Processor 14-24
Deadstart 14-26
Barrel and Slot 14-26
PP Registers 14-28
R Register 14-28
A Register 14-28

P Register 14-29
Q Register 14-29
K Register 14-29
PP Numbering 14-30
PP Memory 14-30
I/O Channels 14-31
Display Station Controller 14-31
Real-Time Clock 14-31
Two-Port Multiplexer 14-32
Maintenance Channel 14-32
IOU Maintenance Registers 14-33
Element Identifier (EID) Register 14-34
Environment Control (EC) Register 14-34
Fault Status (FS) Register 14-34
Fault Status Mask Register 14-34
Options Installed (OI) Register 14-34
OS Bounds Register 14-34
Status Summary Register 14-35
Test Mode (TM) Register 14-35
Central Memory Access by PPs 14-35

CYBER 990E, 995E, and 994 Functional Descriptions 14

This chapter provides functional descriptions of the central processor (CP), central memory (CM), and input/output unit (IOU) as shown in the system block diagram in chapter 13. Functional descriptions for the system display station and the cooling system are in their respective manuals listed in the system publication index in About This Manual.

Central Processor

The CP consists of the instruction section, registers, execution section, cache memory, and central memory control.

Instruction Section

The instruction section consists of logic for instruction control.

Instruction Lookahead

The instruction lookahead hardware (ILH) speeds up instruction processing by stacking prefetched instructions to make them immediately available for execution. It also accurately predicts program branching based on the recent history of each conditional branch.

To maintain a continuous flow of instructions, the instruction section prefetches instruction words ahead of the instruction being read and stores them in the 64-word instruction buffer stack (IBS). This high-speed buffer is set in the instruction stream between CM and the CP execution section. When an instruction is requested for execution, IBS checks whether that instruction is in the stack. If in the stack, the instruction proceeds to instruction decode and initiation. If not in the stack, the instruction is fetched from CM and placed in the IBS. A least-recently-used replacement algorithm determines the new instruction's position in the IBS.

If the instruction issued is a conditional branch, the instruction section predicts the branch taken or not taken based on the recent history of the branch. It then prefetches instructions along that path before the branch outcome is known.

The actual result of the branch determines whether the conditionally issued instructions may execute to completion. If a branch prediction is correct, the instruction section enables the modifying of registers and CM from issued instructions. If a branch prediction is incorrect, hardware purges the issued instructions along the incorrect branch and issues instructions along the correct branch. In this case, registers and CM appear as if no instructions were issued after the incorrectly predicted branch. As a precaution to incorrect branch prediction, the instruction section will not issue subsequent branch instructions until the prior branch instruction is resolved.

Maintenance Access Control

The maintenance access control performs initialization and maintenance operations in the CP.

Instruction Control Sequences

The instruction control section performs instruction translation and control sequences. Each control sequence obtains the necessary instruction operands from the operating registers and provides the control signals for execution. Instructions read from CM are 60-bit instruction words that are in four 15-bit groups, two 30-bit groups, or a combination of 15-bit and 30-bit groups. The 15-bit groups are termed parcels with the first parcel (parcel 0) being the highest-order 15 bits of a 60-bit CM word. Second, third, and fourth parcels (parcels 1, 2, and 3) follow in order. The 30-bit groups contain two 15-bit parcels.

The instruction control sequences control the execution of one or more instructions of a common type. These sequences and associated instructions are briefly described in this section. For further information, refer to CP Instruction Descriptions in chapter 16.

Boolean Sequence

The Boolean sequence controls instructions that require bit-by-bit data manipulation. This includes both the logical and transmissive operations. The instructions requiring logical operations are:

11	Logical product (Xj) and (Xk) to Xi	BXi Xj * Xk
12	Logical sum of (Xj) and (Xk) to Xi	BXi Xj + Xk
13	Logical difference of (Xj) and (Xk) to Xi	BXi Xj - Xk
15	Logical product of (Xj) with complement of (Xk) to Xi	BXi -Xk * Xj
	Logical sum of (Xj) with complement of (Xk) to Xi	BXi -Xk + Xj
17	Logical difference of (Xj) with complement of (Xk) to Xi	BXi -Xk - Xj

The instructions requiring transmissive operations are:

10	Transmit (Xj) to Xi	BXi Xj
11	Transmit complement of (Xk) to Xi	BXi -Xk

Shift Sequence

The shift sequence controls instructions that require shifting the 60-bit field of data within the operand word. The shift instructions are:

20	Left shift (Xi) by jk	LXi jk
21	Right shift (Xi) by jk	AXi jk
22	Left shift (Xk) nominally (Bj) places to Xi	LXi Bj, Xk
23	Right shift (Xk) nominally (Bj) places to Xi	AXi Bj, Xk
43	Form mask of jk bits to Xi	MXi jk

The shift sequence also controls the pack and unpack instructions. In the packed floating format, the coefficient is contained in the lower 48 bits. The sign and biased exponents are contained in the upper 12 bits. The unpack instruction obtains the packed word from the Xk register, delivers the coefficient to the Xi register, and delivers the exponent to the Bj register. The unpack and pack instructions are:

26	Unpack (Xk) to Xi and Bj	UXi Bj, Xk
27	Pack (Xk) and (Bj) to Xi	PXi Bj, Xk

The shift sequence also controls the normalize operations. The coefficient portion of the operand is repositioned, and the exponent is adjusted so that the most significant bit of the coefficient is in the highest-order bit position of the coefficient, and the exponent is decreased by the number of bit positions shifted. The normalize instructions are:

24	Normalize (Xk) to Xi and Bj	NXi Bj, Xk
25	Round normalize (Xk) to Xi and Bj	ZXi Bj, Xk

Floating-Add Sequence

The floating-add sequence controls the operations necessary to form the 48-bit floating sum with a 12-bit exponent of the floating-point sum or difference of two floating-point operands. The floating-add instructions are:

30	Floating sum of (Xj) and (Xk) to Xi	FXi Xj + Xk
31	Floating difference of (Xj) and (Xk) to Xi	FXi Xj - Xk
32	Floating double-precision sum of (Xj) and (Xk) to Xi	DXi Xj + Xk
33	Floating double-precision difference of (Xj) and (Xk) to Xi	DXi Xj - Xk
34	Round floating sum of (Xj) and (Xk) to Xi	RXi Xj + Xk
35	Round floating difference of (Xj) and (Xk) to Xi	RXi Xj - Xk

Floating-Multiply and Floating-Divide Sequence

The floating-multiply and floating-divide sequence controls the operation of floating-multiply, floating-divide, and population-count instructions.

The multiply instructions are:

40	Floating product of (Xj) and (Xk) to Xi	FXi Xj * Xk
41	Round floating product of (Xj) and (Xk) to Xi	RXi Xj * Xk
42	Floating double-precision product of (Xj) and (Xk) to Xi	DXi Xj * Xk

The divide instructions are:

44	Floating divide (Xj) by (Xk) to Xi	FXi Xj/Xk
45	Round floating divide (Xj) by (Xk) to Xi	RXi Xj/Xk

The population-count instruction counts the number of one bits in a 60-bit operand. The instruction is:

47	Population count of (Xk) to Xi	CXi Xk

Increment Sequence

The increment sequence controls the ones complement addition and subtraction of 18-bit fixed-point operands for increment instructions 50 through 77. The sequence also controls the 60-bit ones complement sum and difference values for long-add instructions 36 and 37.

The increment instructions are:

50	Set Ai to (Aj) + K	SAi Aj + K
51	Set Ai to (Bj) + K	SAi Bj + K
52	Set Ai to (Xj) + K	SAi Xj + K
53	Set Ai to (Xj) + (Bk)	SAi Xj + Bk
54	Set Ai to (Aj) + (Bk)	SAi Aj + Bk
55	Set Ai to (Aj) - (Bk)	SAi Aj - Bk
56	Set Ai to (Bj) + (Bk)	SAi Bj + Bk
57	Set Ai to (Bj) - (Bk)	SAi Bj - Bk
60	Set Bi to (Aj) + K	SBi Aj + K
61	Set Bi to (Bj) + K	SBi Bj + K
62	Set Bi to (Xj) + K	SBi Xj + K
63	Set Bi to (Xj) + (Bk)	SBi Xj + Bk
64	Set Bi to (Aj) + (Bk)	SBi Aj + Bk
65	Set Bi to (Aj) - (Bk)	SBi Aj - Bk
66	Set Bi to (Bj) + (Bk)	SBi Bj + Bk
67	Set Bi to (Bj) - (Bk)	SBi Bj - Bk
70	Set Xi to (Aj) + K	SXi Aj + K
71	Set Xi to (Bj) + K	SXi Bj + K
72	Set Xi to (Xj) + K	SXi Xj + K
73	Set Xi to (Xj) + (Bk)	SXi Xj + Bk
74	Set Xi to (Aj) + (Bk)	SXi Aj + Bk
75	Set Xi to (Aj) - (Bk)	SXi Aj - Bk
76	Set Xi to (Bj) + (Bk)	SXi Bj + Bk
77	Set Xi to (Bj) - (Bk)	SXi Bj - Bk

The long-add instructions are:

36	Integer sum of (Xj) and (Xk) to Xi	IXi Xj + Xk
37	Integer difference of (Xj) and (Xk) to Xi	IXi Xj - Xk

Compare/Move Sequence

The compare/move sequence controls data manipulation on a character basis. The compare/move instructions (also referred to as CMU instructions) are 60-bit instructions that use six support registers for source and result field CM addresses and character position offsets. The support registers load from the 60-bit instruction word. The compare/move instructions are:

464	Move indirect (Bj) + K	IM Bj + K
465	Move direct	DM
466	Compare collated	CC
467	Compare uncollated	CU

The support registers are:

- An 18-bit K1 register that specifies which relative CM address word contains the first character of the source data field.
- An 18-bit K2 register that specifies which relative CM address word contains the first character of the result field.
- A 4-bit C1 register that specifies the character position or offset of the first CM word of the source field.
- A 4-bit C2 register that specifies the character position or offset of the first CM word of the result field.
- Two 16-bit L registers (LA and LC) that specify the number of characters in the data field. The LA register is associated with K1, and the LC register is associated with K2. Instruction 464 uses 14 register bits. Instructions 465, 466, and 467 use only the lower eight register bits.

NOTE

CMU instructions are provided for compatibility with previous systems. For better performance, recompile jobs to avoid use of CMU instructions.

CYBER 170 Exchange Sequence

A CYBER 170 exchange sequence is the method used to swap jobs in and out of execution. When a CYBER 170 exchange jump instruction occurs, the CYBER 170 exchange sequence writes the contents of the current job's CP registers (described later in this chapter) into an area of central memory called a CYBER 170 exchange package. A CYBER 170 exchange package is associated with each job. It contains sufficient information to restart a job if the job is interrupted during execution and swapped out by a CYBER 170 exchange jump. To complete the sequence, CP registers for another job are read from its CYBER 170 exchange package and that job begins or resumes execution. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

Block Copy Sequence

The block copy sequence controls the transfer of data between CM and UEM. The number of words to be transferred is determined by the addition of K to the contents of Bj. The starting address for CM is formed by adding either the A0 register or certain bits of the X0 register to the RAC reference address. The starting address for UEM is formed by adding certain bits of the X0 register to the RAE reference address. The block copy instructions are:

011	Block copy Bj + K words from UEM to CM	RE Bj + K
012	Block copy Bj + K words from CM to UEM	WE Bj + K

Direct Read/Write Sequence

Instructions 014 and 015 perform single-word, direct read and write operations for UEM; and instructions 660 and 670 perform single-word, direct read and write operations for central memory.

014	Read one word from UEM at (Xk + RAE) into Xj	RXj Xk
015	Write one word from Xj to UEM at (Xk + RAE)	WXj Xk
660	Read central memory at (Xk) to Xj	CRXj Xk
670	Write Xj into central memory at (Xk)	CWXj Xk

Normal Jump Sequence

The normal jump sequence controls the execution of branch instructions 02 through 07. The 02 instruction performs an unconditional jump to the Bi register address plus K. The branch address is K with i equals zero. The 02 instruction is:

02	Jump to (Bi) + K	JP Bi + K

The conditional jump instructions 03 through 07 branch to address K if the jump condition is met.

These instructions are:

030	Branch to K if (Xj) = 0	ZR Xj, K
031	Branch to K if (Xj) = 0	NZ Xj, K
032	Branch to K if (Xj) is positive	PL Xj, K
033	Branch to K if (Xj) is negative	NG Xj, K
034	Branch to K if (Xj) is in range	IR Xj, K
035	Branch to K if (Xj) is out of range	OR Xj, K
036	Branch to K if (Xj) is definite	DF Xj, K
037	Branch to K if (Xj) is indefinite	ID Xj, K
04	Branch to K if (Bi) = (Bj)	EQ Bi, Bj, K
05	Branch to K if (Bi) ≠ (Bj)	NE Bi, Bj, K
06	Branch to K if (Bi) ≥ (Bj)	GE Bi, Bj, K
07	Branch to K if (Bi) < (Bj)	LT Bi, Bj, K

Return Jump Sequence

The return jump sequence controls the execution of three instructions.

00	Error exit to MA or program stop	PS
010	Return jump to K	RJ K
013	Central exchange jump to (Bj) + K or monitor exchange jump to MA	XJ Bj + K

Registers

The CP contains the operating and support registers described in the following paragraphs. These registers are located in the register unit and process state registers sections (refer to figure 13-3).

The contents of these registers can be written into memory and reloaded from memory as a CYBER 170 exchange package by a single CP instruction (CYBER 170 exchange jump). Figure 14-1 shows the CYBER 170 exchange package.

The time a CYBER 170 exchange package resides in CP hardware is called an execution interval. During this interval, the contents of X, A, B, and P registers can be changed by CP instructions. The contents of other support registers change only as a result of a CYBER 170 exchange jump. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

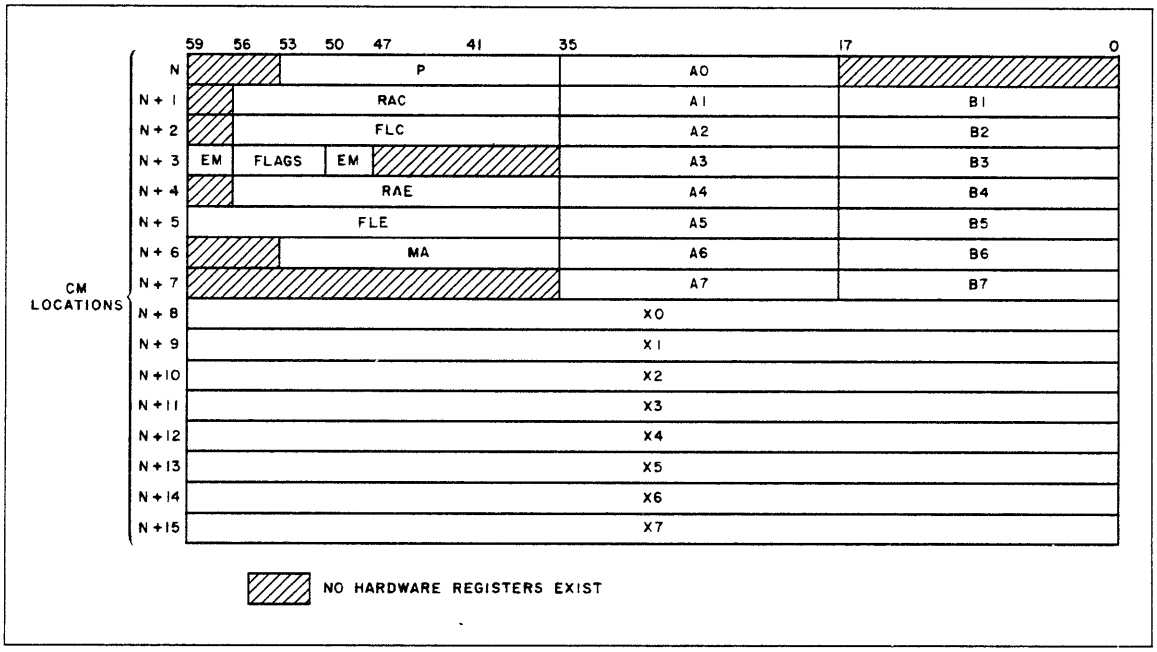


Figure 14-1. CYBER 170 Exchange Package

Operating Registers

The operating registers consist of operand (X), address (A), and index (B) registers. These registers minimize memory references for arithmetic operands and results.

X Registers

The CP contains eight 60-bit X registers, X0 through X7. The X0 register is used in the compare instructions to indicate if two fields of characters are equal. Also, the X0 register provides the relative UEM starting address in a block copy operation.

The X1 through X7 registers are primarily data handling registers for computation. X1 through X5 are used to input data from CM, and X6 and X7 are used to transmit data to CM.

Operands and results transfer between CM and the X registers as a result of placing CM addresses into corresponding A registers.

A Registers

The CP contains eight 18-bit A registers, A0 through A7. The A0 register serves as an intermediate register for the user's discretion. The A0 register is used in the compare collate instruction for the collate table address. Also, the A0 register provides the relative CM starting address in a block copy operation.

The A1 through A7 registers are essentially CM operand address registers associated one-for-one with the X registers. Placing a quantity into an address register (A1 through A5) causes a CM read reference to that address and transmits the CM word to the corresponding X register (X1 through X5). Similarly, placing a quantity into the A6 or A7 register causes the word in the corresponding X6 or X7 register to be written into that relative address of CM.

B Registers

The CP contains eight 18-bit B registers, B0 through B7. These registers are primarily indexing registers to control program execution. Program loop counts may also be incremented or decremented in these registers.

Program addresses may be modified on the way to an A register by adding or subtracting B register quantities. The B registers also hold shift counts for the nominal Bj shifts, the resultant exponent for the unpack, the operand exponent for the pack, and the resultant shift count from a normalize. The B0 register always contains positive zero which can be used as an operand. This register cannot hold results from instructions.

Support Registers

Eight support registers assist the operating registers during the execution of programs. The contents of the support registers are stored in CM, and their new contents are loaded from CM during a CYBER 170 exchange sequence. With the exception of the P register, the contents of the support registers cannot be altered during the execution interval of a CYBER 170 exchange package. When the execution interval completes, the data in the support registers is sent back to CM through a CYBER 170 exchange jump.

P Register

The 18-bit program address (P) register loads from CM during the first word of a CYBER 170 exchange sequence and contains the current program execution address. The register serves as a program address counter and holds the relative CM address for each program step.

RAC Register

The 21-bit CM reference address (RAC) register loads from CM during the second word of a CYBER 170 exchange sequence. An absolute CM address forms by adding RAC to a relative address determined by the instruction. The content of the P register is added to RAC to form the program address in CM. A P-equal-to-zero condition specifies relative address 0 and, therefore, (RAC). This CM location is reserved for recording error exit conditions and should not be used to store data or instructions.

FLC Register

The 21-bit CM field length (FLC) register loads from CM during the third word of a CYBER 170 exchange sequence. The FLC register defines the size of the field of the program in execution. Relative CM addresses are compared with FLC to check that the program is not going out of its allocated memory range.

EM Register

The 6-bit exit mode (EM) register loads from CM during the fourth word of a CYBER 170 exchange sequence. The EM register holds six exit mode selection bits that control individual error conditions for a program. Selected EM register bits cause the CP to error exit when the corresponding conditions occur. Any or all of the six bits can be set at one time. Clear EM register bits allow the CP to continue, without error processing, when most of the corresponding conditions occur. Refer to the error exit tables under Error Response in chapter 17 for specific cases. The exit mode selection bits appear in the exchange package as bits 48 through 50 and 57 through 59. The bits and their corresponding conditions are:

Mode Selection Bit	Significance
48	Address out of range
49	Infinite operand
50	Indefinite operand
57	Hardware error
58	Hardware error
59	Hardware error

Flag Register

The 6-bit flag register loads from CM during the fourth word of a CYBER 170 exchange sequence. The flag register holds six bits that function as control flags.

Bit	Condition
51	Hardware error bit.
52	Instruction stack (lookahead) purge flag. If set, extended purging of instruction lookahead registers is enabled. For further information, refer to Instruction Lookahead Purge Control under CP Programming in chapter 17.
53	CMU interrupted flag. If set, one of instructions 464 through 467 has been interrupted. The information necessary to resume operation has been saved.
54	Block copy flag. If set, block copy instructions (011, 012) use bits 30 through 50 of X0 rather than A0 to determine the CM address. For further information, refer to the descriptions of the block copy instructions in chapter 16.
55	Expanded addressing select flag. If set, UEM is operating in expanded addressing mode; if clear, UEM is operating in 24-bit standard addressing mode. For further information, refer to Addressing Modes under Memory Programming in chapter 17.
56	UEM enable flag. If set, UEM is available. This flag must be set to allow 011, 012, 014, and 015 instructions to access UEM.

RAE Register

The 21-bit UEM reference address (RAE) register loads from CM during the fifth word of a CYBER 170 exchange sequence. The lower 6 bits of this register are always zero. An absolute UEM address forms by adding RAE to the relative address which is determined by the instruction.

FLE Register

The 24-bit UEM field length (FLE) register loads from CM during the sixth word of a CYBER 170 exchange sequence. The lower 6 bits of this register are always zero. The FLE register defines the size of the field in UEM for the program in execution. Relative UEM addresses are compared with FLE.

MA Register

The 18-bit monitor address (MA) register loads from CM during the seventh word of a CYBER 170 exchange sequence. The MA register contains the absolute starting address of an exchange package which is used when executing a central exchange jump (013) instruction with the CYBER 170 monitor flag clear, or when honoring a monitor exchange jump to MA (262x) instruction with the CYBER 170 monitor flag clear. For further information, refer to CYBER 170 Exchange Jump in chapter 17.

Execution Section

The execution section combines the operands into results, providing additional sequencing control where necessary.

Cache Memory

Cache memory is a high-speed buffer memory which is transparent to the user. It reduces effective CM access time by eliminating unnecessary CM references. When the CP first reads CM, a block of four words from CM (containing the requested word) is read rapidly into cache memory.

These words include all data except instructions. On subsequent reading of any of these words, CM need not be accessed when these words are in cache memory. Often this is the case because the same data is read more than once, or because a loop of instructions is repeatedly executed. Cache memory is 4096 words.

Addressing Section

An address adder calculates memory addresses for data and unconditional jump instructions.

Memory management hardware verifies that memory addresses are to access permitted memory areas. If this is the case, this hardware accesses cache memory and, if necessary, central memory.

Central Memory Control

Central memory control (CMC) provides an interface to CM for the CP and IOU. It is physically located in the CP cabinet. CMC includes:

- Ports and distributor
- SECDED logic
- Partial-write logic
- Memory control logic
- Maintenance registers

Central Memory

The CM performs the following functions.

- In a CM consisting of 64K chips, each memory cabinet contains eight banks, storing 2M to 8M 64-bit words (the leftmost 4 bits are undefined) and an 8-bit SECDED code. A maximum of two memory cabinets allows a total of 16M 64-bit words.
- In a CM consisting of 256K chips, a single memory cabinet houses eight independent banks, storing from 8M to 32M 64-bit words (the leftmost 2 bits are undefined) and an 8-bit SECDED code.
- The two ports make CM accessible to the CP and every PP.
- A bounds register limits access to CM from either or both ports.
- The SECDED generators generate the SECDED code bits stored with each word. SECDED checks circuits, corrects single-bit errors, and detects double-bit errors.
- The maintenance channel interface gives a PP in the IOU access to the CM maintenance registers for system initialization, corrective action, error reporting and diagnostics, and for setting the port bounds register.

Address Format (CYBER 990E and 995E)

Figure 14-2 illustrates the address format used between CMC and CM in a CYBER 990E or 995E.

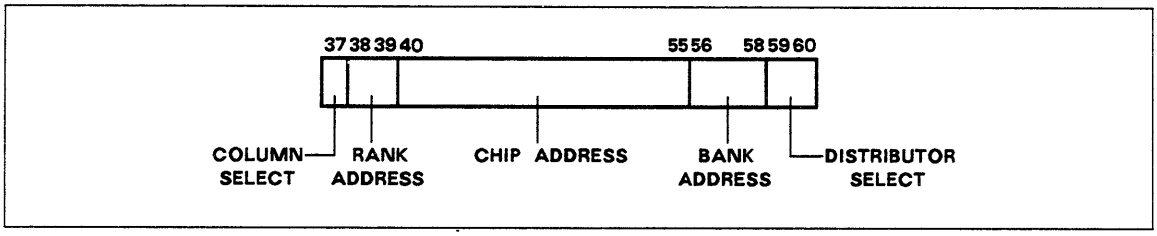


Figure 14-2. Address Format

The following list defines the address fields for figure 14-2.

- Column Select specifies one of two columns.
- Rank Address specifies one of four ministorage board ranks. Each of the four ranks contains 16 ministorage boards.
- Chip Address defines the address of one word in the 64K MOS memory chips for the selected bank.
- Bank Address specifies one of eight banks. (Each bank requires two ministorage boards on a storage board assembly.)
- Distributor Select specifies one of four distributors in a column.

Address Format (CYBER 994)

Figure 14-3 illustrates the address format for the CYBER 994 sent from the CMC to CM.

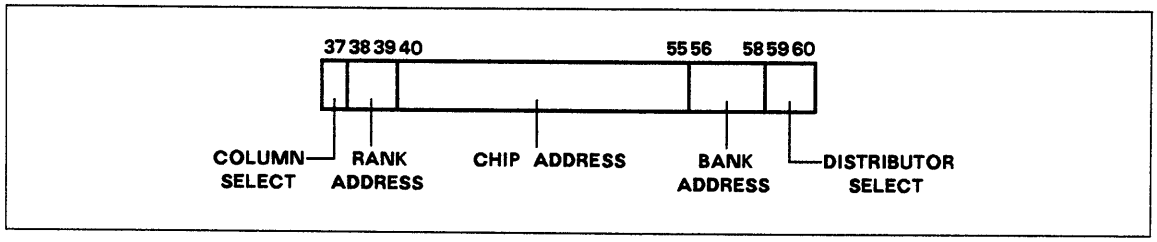


Figure 14-3. Address Format (CYBER 994)

The following list defines the address fields for figure 14-3.

- Column Select specifies one of two columns, but column 1 is not physically present.
- Rank Address specifies one of four ministorage board ranks. Each of the four ranks contains 16 ministorage boards.
- Chip Address defines the address of one word in the 256K MOS memory chips for the selected bank.
- Bank Address specifies one of eight banks. (Each bank requires two ministorage boards on a storage board assembly.)
- Distributor Select specifies one of four distributors in a column.

CM Access and Cycle Times

The CM read/write cycle time is 96 nanoseconds (6 clock periods). The read access time is 128 nanoseconds (8 clock periods). The maximum data transfer rate is one word every 16 nanoseconds (1 clock period) from each of the four distributors.

The CM bounds register, located in CMC, limits CM write access for the selected ports to an area between two addresses specified in this register. The CM bounds register is set through the maintenance channel (refer to Maintenance Channel Programming in chapter 17).

CM Ports and Priorities

A priority network resolves access conflicts on a rotating basis, preventing long-term lockout of any port. In case of simultaneous requests, the CP has priority.

Refresh requests have priority over port requests. Refer to table 14-1 for maximum request lockout time in bank cycles.

Table 14-1. Port Priority

Port	Read or Write Requests
Refresh	1
Port 0	4
Port 1	5

Note:

One bank cycle equals 6 clock periods or 96 nanoseconds.

SECDED

The SECDED logic corrects single-bit errors during a CM read, permitting unimpeded computer operation. The SECDED logic prepares for the error correction by generating error correction code (ECC) bits for each data word, and by storing these ECC bits in CM with the data word during the CM write.

Table 14-2 lists the hexadecimal codes for all the combinations of syndrome bits with the number of the data bit assigned each code or a note categorizing the code. Then, during a CM read, CM performs the following SECDED sequence.

1. Read one CM word and generate new ECC bits for data portion of CM word.
2. Compare new ECC bits with CM word ECC bits.
3. If old and new ECC bits match, no error exists. Send data to requesting unit.
4. If bits do not match, generate syndrome bits from result of ECC compare.
5. Decode syndrome bits to determine if single or multiple bit failure.
6. If single bit failure, correct by inverting failing bit in data word. Send corrected word to requesting unit.
7. If multiple bit or other uncorrectable error, send uncorrectable error response code to CP or IOU. A PP in the IOU may then analyze the syndrome bits using the maintenance channel.

Table 14-2. SECDED Syndrome Codes/Corrected Bits

Code	Bit	Code	Bit	Code	Bit	Code	Bit
00	6	10	67[2]	20	66[2]	30	2/3[5]
01	71[2]	11	3	21	3	31	4
02	70[2]	12	3	22	3	32	4
03	6/7[5]	13	4	23	4	33	3
04	69[2]	14	3	24	3	34	4
05	3	15	4	25	4	35	3
06	3	16	4	26	4	36	3
07	24[1]	17	5	27	5	37	28[1]
08	68[2]	18	3	28	3	38	4
09	3	19	4	29	4	39	3
0A	3	1A	4	2A	4	3A	3
0B	16[1]	1B	5	2B	5	3B	20[1]
0C	4/5[5]	1C	4	2C	4	3C	3
0D	8[1]	1D	5	2D	5	3D	12[1]
0E	0[1]	1E	5	2E	5	3E	4[1]
0F	3	1F	4	2F	4	3F	3

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes indicated.

6. No error detected.

(Continued)

Table 14-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
40	65[2]	50	3	60	3	70	56[1]
41	3	51	4	61	4	71	5
42	3	52	4	62	4	72	5
43	4	53	3	63	3	73	60[1]
44	3	54	4	64	4	74	5
45	4	55	3	65	3	75	58[1]
46	4	56	3	66	3	76	62[1]
47	5	57	26[1]	67	30[1]	77	5
48	3	58	4	68	4	78	5
49	4	59	3	69	3	79	57[1]
4A	4	5A	3	6A	3	7A	61[1]
4B	5	5B	18[1]	6B	22[1]	7B	5
4C	4	5C	3	6C	3	7C	59[1]
4D	10[5]	5D	10[1]	6D	14[1]	7D	5
4E	5	5E	2[1]	6E	6[1]	7E	5
4F	4	5F	3	6F	3	7F	63[1]

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes indicated.

(Continued)

Table 14-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
80	64[2]	90	3	A0	3	B0	48[1]
81	3	91	4	A1	4	B1	5
82	3	92	4	A2	4	B2	5
83	4	93	3	A3	3	B3	52[1]
84	3	94	4	A4	4	B4	5
85	4	95	3	A5	3	B5	50[1]
86	4	96	3	A6	3	B6	54[1]
87	5	97	25[1]	A7	29[1]	B7	5
88	3	98	4	A8	4	B8	5
89	4	99	3	A9	3	B9	49[1]
8A	4	9A	3	AA	3	BA	53[1]
8B	5	9B	17[1]	AB	21[1]	BB	5
8C	4	9C	3	AC	3	BC	51[1]
8D	5	9D	9[1]	AD	13[1]	BD	5
8E	5	9E	1[1]	AE	5[1]	BE	5
8F	4	9F	3	AF	3	BF	55[1]

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes indicated.

(Continued)

Table 14-2. SECDED Syndrome Codes/Corrected Bits *(Continued)*

Code	Bit	Code	Bit	Code	Bit	Code	Bit
C0	0/1[5]	D0	40[1]	E0	32[1]	F0	3
C1	4	D1	5	E1	5	F1	4
C2	4	D2	5	E2	5	F2	4
C3	3	D3	44[1]	E3	36[1]	F3	3
C4	4	D4	5	E4	5	F4	4
C5	3	D5	42[1]	E5	34[1]	F5	3
C6	3	D6	46[1]	E6	38[1]	F6	3
C7	27[1]	D7	5	E7	5	F7	31[1]
C8	4	D8	5	E8	5	F8	4
C9	3	D9	41[1]	E9	33[1]	F9	3
CA	3	DA	45[1]	EA	37[1]	FA	3
CB	19[1]	DB	5	EB	5	FB	23[1]
CC	3	DC	43[1]	EC	35[1]	FC	3
CD	11[1]	DD	5	ED	5	FD	15[1]
CE	3[1]	DE	5	EE	5	FE	7[1]
CF	3	DF	47[1]	EF	39[1]	FF	3

1. Corrected single-bit error.

2. Syndrome code bit failed (single code bit set).

3. Double error or multiple error (even number of code bits set).

4. Multiple error reported as a single error.

5. Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes indicated.

CM Layout

Central memory contains an area that is reserved for special software called Virtual State software. Along with the hardware and microcode, this software handles the operations of Virtual State as described in chapter 17. Virtual State software is located at the higher end of memory. The remaining memory is available to the CYBER 170 State and may be allocated as central memory (accessible via RAC and FLC) or as UEM (accessible via RAE and FLE and the 011, 012, 014, and 015 instructions). Refer to figure 14-4.

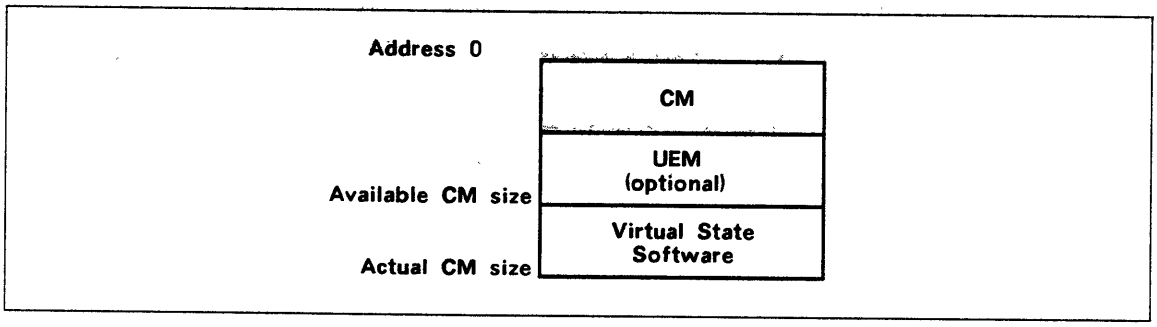


Figure 14-4. CM Layout

CM Bounds Register

The CM bounds register limits the write access to CM from specified ports. The ports are limited to the area between an upper and lower bound as specified in the CM bounds register. Bits in byte 0 specify the port(s) from which the write access is limited. The CM bounds register is set through the maintenance channel. For further information, refer to Maintenance Channel Programming in chapter 17.

Central Memory Reconfiguration

Central memory reconfiguration is a manually performed function that permits the computer operator to restructure the CM addresses so that a failing part of CM can be quickly locked out to provide a continuous block of usable CM. CM reconfiguration is accomplished by setting the switches on the memory unit to manipulate the upper address bits. The CYBER 994 cannot be reconfigured.

When a configuration switch is set forcing a CM address bit to a zero/one, the address range corresponding to the original installed memory accesses some parts of the reconfigured memory more than once. Addresses up to the rightmost forced bit, and half the addresses using the rightmost forced bit, cover a contiguous address space from location 0, which is the reconfigured memory. For further information, refer to chapter 15.

Input/Output Unit

The IOU consists of the NIO subsystem and the CIO subsystem. (CIO subsystem is an option on CYBER 994). The NIO consists of 20 PPs, each having 8K x 16-bit words of memory and a repertoire of 122 instructions. All 20 PPs share access to 24 I/O channels. The PPs are partitioned into four barrels of five PPs.

The CIO consists of one or two barrels each containing five PPs per barrel. Each barrel of the CIO has a group of five direct-memory access (DMA) I/O channels. These channel groups are dedicated to a barrel and no inter-barrel communication is possible on DMA channels.

An optional DMA-enhanced intelligent standard interface (ISI) channel adapter, intelligent peripheral interface (IPI) channel adapter, or CYBER 170 channel adapter can be installed in any one of 10 channel locations in the CIO cabinet. The adapter transfers data between the ISI, IPI, or CYBER 170 channel and PP memory using standard I/O intructions. It also supports DMA transfer in which data flows directly between the CM and an external device without going through the PP.

There are two types of CYBER 170 DMA transfers, fast and normal. Fast transfers are used with the Extended Semiconductor Memory-II (ESM-II), and normal transfers are used with other CYBER 170 external devices.

This data flow is called DMA since it accesses memory directly. DMA allows a higher I/O bandwidth and also allows the PP execution unit to operate independently from the I/O channel during data transfers (concurrently). This concurrency gives the PP more instruction cycle time to process I/O requests from the CP.

The IOU performs the functions necessary to locate, select, and initialize the external devices connected to the system and controls the transfer of data between a selected device and CM. The IOU also performs system maintenance functions.

The IOU contains the following functional areas:

- Peripheral processors
- I/O channels
- Display station controller
- Real-time clock
- Two-port multiplexer
- Maintenance channel
- CM access

Peripheral Processor

The basic IOU contains 20 NIO PPs and none (CIO cabinet is an option on CYBER 994), five, or ten CIO PPs. Either cabinet configuration can be expanded with five or ten CIO PPs. Each PP is a logically independent computer with its own memory.

Each 5-PP group is organized into a multiplexing system which allows the PPs to share common hardware for arithmetic, logical, and I/O operations without losing independence.

This multiplexing system comprises five ranks of registers termed a barrel. Each rank contains information related to the instruction being executed by one PP.

Within the NIO subsystem:

- Any PP in a barrel may communicate with any other PP over any of the 24 I/O channels.
- The channels are numbered from 0 through 11_8 and from 20_8 through 31_8.
- Inter-PP communication is always on a 16-bit boundary.
- The NIO subsystem supports CYBER 170 peripheral equipment (12 bits) using CYBER 170 channel modules.
- Special I/O instructions are provided to convert 12-bit channel words to 16-bit PP words and vice versa.

Within the CIO subsystem:

- PPs within a barrel may communicate with any other PP in the same barrel on any of the five dedicated I/O channels.
- Communication between barrel or with the NIO subsystem must be done via channels 15_8 or 17_8.
- Inter-PP communication is always on a 16-bit boundary. The CIO channels are 0 through 11_8.

Each PP can communicate with:

- Other PPs over the I/O channels (subject to restrictions stated above).
- The CP via CM read and write operations.
- The CP (in CYBER 170 State operation) by issuing a CYBER 170 State exchange request to a specific CYBER 170 State exchange package associated with the issuing PP.

Each PP can also cause an interrupt condition with the CP operating in either Virtual State or CYBER 170 State.

Each PP executes programs alone or with other PPs to control data transfers between external devices and CM. These programs, called I/O drivers, comprised IOU instructions combined to interact with operating system requests issued through CM.

The I/O drivers translate generalized operating system requests into control functions for accessing the external devices and may also perform device scheduling and optimization.

The I/O drivers use PP memory as a buffer for the data transfer between external devices and CM to isolate IOU data transfers from variations in CM transfer rate.

The optional DMA-enhanced ISI, IPI, and CYBER 170 channels can access CM directly and do not require buffer memories.

Deadstart

A deadstart sequence allows the IOU to initialize itself. A deadstart sequence is initiated, depending on the terminal, as follows:

- CC545 display station console -- press the DEADSTART switch and either the S key or CR.
- CC634-B display terminal -- press CTRL-G, CTRL-R, and either the S key or CR.
- CC598-B system console -- press either CTRL-G or CTRL-F2 and either the S key or CR.

For further information regarding the deadstart sequence and options, refer to CYBER Initialization Package (CIP) Reference Manual listed in About This Manual.

Barrel and Slot

The barrel consists of the A, K, P, Q, and R registers, each one of which has five ranks numbered 0 through 4 (figure 14-5).

Information in these registers is transferred from one rank to the next at a uniform 20-MHz rate, providing a multiplexed system of five PPs, each operating at a 4-MHz rate. The registers are stationary while the PPs rotate. For example, rank 4 registers contain PP0, PP1, PP2, PP3, and PP4 in succession, each consuming 50 nanoseconds of the total cycle time of 250 nanoseconds.

Since PP memories operate at a slower rate, independent memory with its own address and data registers is provided for each PP.

Each time data enters the slot, a portion of the instruction for that data is executed. The slot performs tasks such as arithmetic and logic operations and program address manipulation. Complete execution of an instruction may require the A, K, P, Q, and R register quantities to go more than one trip around the barrel and through the slot.

The PPM may be referenced once each time the PP passes around the barrel and through the slot. During its slot time, the PP may also communicate with CM or with any of the I/O channels.

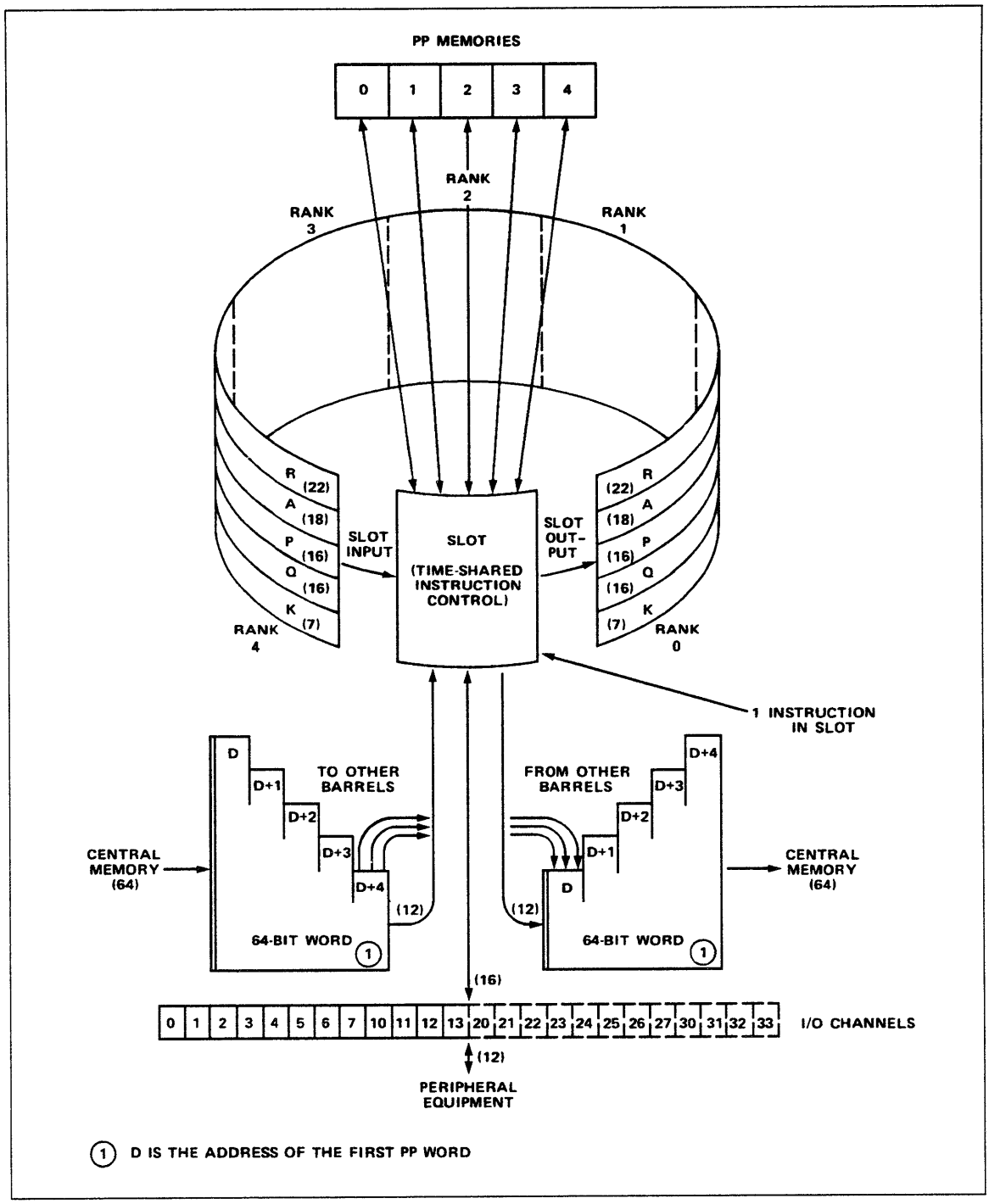


Figure 14-5. Barrel and Slot

PP Registers

The PP registers, which are discussed in the following paragraphs, are:

- R register
- A register
- P register
- Q register
- K register

R Register

The 22-bit R register, in conjunction with the A register, forms an absolute CM address for CM read/write instruction (refer to Central Memory Access by PPs later in this chapter).

A Register

The 18-bit A register contains one of two operands for arithmetic and logic operations. The content of A may be:

- Arithmetic operand
- A CM address or part of a CM address
- An I/O function
- An I/O data word
- Word count for a block I/O or CM transfer

Various instructions operate on 6, 12, 16, or 18 bits of the A register. Calculation results are always placed in the A register, although some instructions also write the result into PP memory.

When the A register provides the CM address, parity is generated with the address for transmission to memory control. When the A register provides data or function words for I/O activities, channel parity is always generated on 16 bits of A.

At deadstart, the A register is set to 10000_8 for the standard 20 PPs and to 20000_8 for the optional 10 PPs.

P Register

The P register operates in two different modes. In 4K mode, P is a 12-bit register; and in 8K mode, P is a 16-bit register. In 8K mode, the PP memory uses only the least significant 13 bits.

The P register is the PP program address counter. Also, during block I/O and CM transfers, the P register temporarily contains the PP memory address of the data transfer. At deadstart, the P register is set to 7777_8 for the NIO PPs and is set to 1_8 for the CIO PPs.

Q Register

The Q register operates in two different modes. In 4K mode, Q is a 12-bit register; and in 8K mode, Q is a 16-bit register. In 8K mode, the PP memory uses only the least significant 13 bits.

The Q register may hold the following data:

- Operand address for direct and indirect addressing.
- Peripheral address of data used during single-word CM read/write instructions.
- Shift count.
- Word count for CM block transfers.
- Upper six bits during constant mode PP instructions.
- Target address for relative jump.
- Channel number for all I/O and channel instructions.

At deadstart, each rank of the Q register is set to a corresponding PP number. Rank 0 is set to PP0, rank 2 is set to PP2, and so on.

K Register

The 7-bit K register is visible to the programmer through the maintenance channel and the IOU deadstart display. This register holds the operation code field of an instruction for display and is used for maintenance purposes. When a PP is halted (idled), this register contains all ones.

PP Numbering

PPs are numbered in octal as follows:

Barrel	PPs
0	N00 through N04
1	N05 through N11
2	N20 through N24
3	N25 through N31
0	C00 through C04
1	C05 through C11

The deadstart sequence decodes a program stored in the IOU microprocessor RAM to determine PP numbering within a barrel. The sequence:

- Assigns barrel numbers according to the program.
- Loads a zero into the Q register in barrel 0 during the first minor cycle after deadstart.

This defines all the data in that rank of the barrel as belonging to PP0 and, since Q is the channel selector, assigns PP0 to channel 0.

During the next minor cycle, Q loads with a one. This defines PP1 and assigns it to channel 1.

This process occurs in parallel in all barrels until the IOU assigns each rank of the barrel with a PP number and a channel number. Reassignment can only be done at deadstart. For further information on PP reassignments, refer to the CIP User's Handbook listed under Additional Related Manuals in About This Manual.

PP Memory

Each NIO PP has an independent 8K-word memory degradeable to 4K; each word contains 16 data bits and 6 bits of SECDED code. Each CIO PP has an independent 8K-word memory; each word contains 16 data bits and six bits of SECDED code.

PP0 reads the deadstart program from the microprocessor RAM during the deadstart operation. Therefore, PP memory 0 must be operational. A PP memory reconfiguration feature allows the user to restore IOU operation if the IOU detects a fault in the PP memory normally assigned to PP0.

To reconfigure, the operator assigns a good PP memory to PP0, and the operating system removes the failing PP memory. Computer operation can continue without the failing PP memory, and repairs can be made during scheduled maintenance. The system must be deadstarted to reconfigure PPs.

I/O Channels

The I/O channels are composed of:

- An internal interface that allows common hardware and software to control the external devices, and
- An external interface that allows the IOU to communicate with the external devices using 12-bit data channels. The internal interface can transfer 16-bit data words between two PPs or between a PP and an external device at a maximum rate of 1 word every 250 ns.

This rate can be sustained for the maximum practical channel transfer (4096 words). During transfers between PPs, if the PPs are in the slot at the same time, the transfer rate is 500 ns.

Any PP can access any of the CYBER 170 bidirectional I/O channels. All PPs communicate with external devices through the independent I/O channels. Each channel may be connected to one or more pieces of external equipment, but only one piece of equipment can use a channel at one time. All channels can be active simultaneously. Available channels are listed as follows:

- Twenty-four CYBER 170 compatible I/O channels or 10 CIO channels available with a maximum data transfer rate of 3 megabytes per second.
- An optional DMA-enhanced ISI channel adapter, IPI channel adapter, or CYBER 170 channel adapter that can be installed in any one of 10 channel locations in the CIO cabinet. The adapters transfer data between the ISI, IPI, or CYBER 170 channels and PP memory using standard I/O instructions. They also support DMA transfer in which data goes directly between CM and an external device without going through the PP. There are two types of CYBER 170 DMA transfers, fast and normal. Fast transfers are used with ESM-II, and normal transfers are used with other CYBER 170 external devices.

Display Station Controller

The display station controller (DSC) is the IOU interface between the PPs and the display station servicing both the keyboard and the video display terminal (VDT). The DSC transmits function words and digital symbol size/position data to the display station and receives digital character codes from the keyboard. It also receives digital symbol codes from the PPs and converts these to analog symbols to the VDT.

Real-Time Clock

The real-time clock is a 12-bit free-running counter, incrementing at a 1-MHz rate. It is permanently attached to channel 14_8. This channel may be read at any time since it is active and full flags are always set.

Two-Port Multiplexer

The two-port multiplexer provides communication capability between a PP and two attached terminals. It can simultaneously drive the two terminals at different baud rates. One port is reserved for maintenance, and the other port is reserved for future use. The two-port multiplexer is permanently attached to channel 15_8. Each port may drive a separate terminal.

For a dual-IOU option, both ports of the primary IOU two-port multiplexer are connected to the 1900X System Console (CC598-B). Only one port of the second IOU two-port multiplexer is connected to the same 1900X System Console, and the other port is not used.

Maintenance Channel

The maintenance channel (MCH) is used for initialization of the CP and CM maintenance registers and monitoring of error status.

The maintenance channel consists of the maintenance channel interface on channel 17_8; a maintenance access control (MAC) in the CP, CM, and IOU; and a set of interconnecting cables. The maintenance channel of the second IOU in a dual-IOU option has access only to the MAC in the IOU.

Any PP can be programmed to act as the maintenance control unit (MCU). However, hardware dictates PP0 as having special deadstart functions such that PP0 optimally serves as the MCU. In any case, the PP acting as the MCU performs initialization and maintenance functions that include:

- Initializing registers, controls, and memories.
- Monitoring and recording error information.
- Verifying error detection and correction hardware.

The MCU directs these operations by sending function words (instructions) over the maintenance channel to the CP, CM, and IOU. The MCU retains all normal PP capabilities and, except for PP0 deadstart functions, does not gain any special hardware capabilities.

A separate MCU in the second IOU in a dual-IOU option reports its error status to the primary IOU via CM.

IOU Maintenance Registers

The MAC in the IOU contains several maintenance registers which hold IOU status or error information. Table 14-3 lists the IOU maintenance registers. For detailed descriptions of these registers, refer to the following paragraphs and to IOU Registers in chapter 17 of this manual.

Table 14-3. IOU Maintenance Registers

Register Name	Number of Bits	Address	Access Type Copy	Access Type MCH
Element identifier (EID)	32	10	-	R
Environment control (EC)	32	30	-	R/W
Fault status 1 (FS-1)	64	80	-	R/W
Fault status 2 (FS-2)	64	81	-	R/W
Fault status mask (FSM)	64	18	-	R/W
Options installed (OI)	64	12	-	R
OS bounds (OSB)	64	21	-	R/W
Status summary (SS)	6	00	-	R
Test mode (TM)	16	A0	-	R/W

Element Identifier (EID) Register

The 32-bit EID register is a backpanel-wired register identifying each system hardware element. The EID bits are represented as follows:

Bits	Description
32 through 39	Element type
40 through 47	Model number
48 through 63	Serial number (hexadecimal)

Environment Control (EC) Register

The 64-bit EC register controls timing margins, test mode and deadstart, PP memory dumps, reconfiguration, and stop-on-error conditions for the IOU. It also selects PP internal registers for reading. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

Fault Status (FS) Registers

The 64-bit FS registers indicate the presence of uncorrectable faults in the IOU, PP memories, I/O channels, or PP hardware. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

Fault Status Mask Register

This 64-bit register controls IOU fault reporting to the IOU fault status (FS) registers. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

Options Installed (OI) Register

The 64-bit OI register identifies the options installed in the IOU. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

OS Bounds Register

The 64-bit operating system (OS) bounds register divides the CM into an upper and a lower region for system protection. The OS bounds register contains a bit for each PP which indicates the region in CM into which the specified PP may initiate exchange operations or writes. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

Status Summary Register

The status summary register indicates errors in the CP, CM and IOU. It also provides information about the PP-halt, error status, and physical environment conditions. Refer to the Maintenance Register Codes Booklet listed in About This Manual for further information.

Test Mode (TM) Register

The 64-bit TM register forces faults in the IOU for testing of the fault sensing logic. Bits 48 through 63 of this register serve a dual role. With the Enable Test Mode Register bit set in the EC register, these bits are used to force test conditions (refer to the Maintenance Register Codes Booklet listed in About This Manual for further information). When the Enable Test Mode Register bit is clear, these read/write bits can be used by software as interlock/flag status bits.

Central Memory Access by PPs

Any PP can access CM. During a write from the IOU to CM, the IOU assembles either four successive 16-bit PP words into one 64-bit CM word (Virtual State) or five successive 12-bit PP words into one 60-bit CM word (CYBER 170 State).

During a CM read, the IOU disassembles either a 64-bit CM word into four 16-bit PP words (Virtual State), or a 60-bit CM word into five 12-bit PP words (CYBER 170 State).

To find the CM address, a PP reads the A register. If bit 17 of the A register is clear, the PP uses the contents of the A register for the CM address. If bit 17 of the A register is set, the PP adds the relocation address from the R register to the A register for the CM address.

A maximum of 30 PPs can simultaneously read CM words, and 30 PPs can write CM words.

CYBER 990E, 995E, and 994 Operating Instructions 15

Controls and Indicators 15-1
Deadstart Displays/Controls 15-1
Central Memory Controls (CYBER 990E and 995E) 15-4
Central Memory Controls (CYBER 994) 15-4

Power-On and Power-Off Procedures 15-6

Operating Procedures 15-6
Control Checks 15-6
Deadstart Sequences 15-6

IOU Reconfiguration 15-7

CYBER 990E, 995E, and 994 Operating Instructions 15

This chapter describes mainframe controls and indicators and the operating procedures which are hardware dependent. Software-dependent procedures are in system software reference manuals listed under Additional Related Manuals in About This Manual.

Controls and Indicators

This section describes IOU deadstart controls and indicators and CM configuration switches used by the system operator. Other controls used by maintenance personnel are described in the hardware operator's guide and the hardware maintenance manuals of the power distribution and warning system, the cooling system, and the system console listed in the system publication index in About This Manual.

Deadstart Displays/Controls

Pressing the DEADSTART pushbutton on the CC545 system console, or pressing the CTRL-G, CTRL-R, then M key on the CC634-B system console, or pressing either the CTRL-G or the CTRL-F2, then the M key on the CC598-B system console initiates deadstart, and an initial deadstart display appears on the screen of the system console. It is created by an independent microcomputer in the mainframe and does not rely on any program being operational in the PPs. The initial deadstart display is used to select a 16-word deadstart program for PP0 and to initiate the deadstart sequence for PP0. It is also used to reconfigure PPMs and barrels, and to display error status and maintenance information.

The format of the deadstart options display is shown in figure 15-1 and the deadstart display is shown in figure 15-2. Table 15-2 describes the two operator-selectable options and table 15-2 describes the operator entries and functions for the deadstart display. Other deadstart displays are available for maintenance use. Refer to the CYBER Initialization Package (CIP) Reference Manual for additional information.

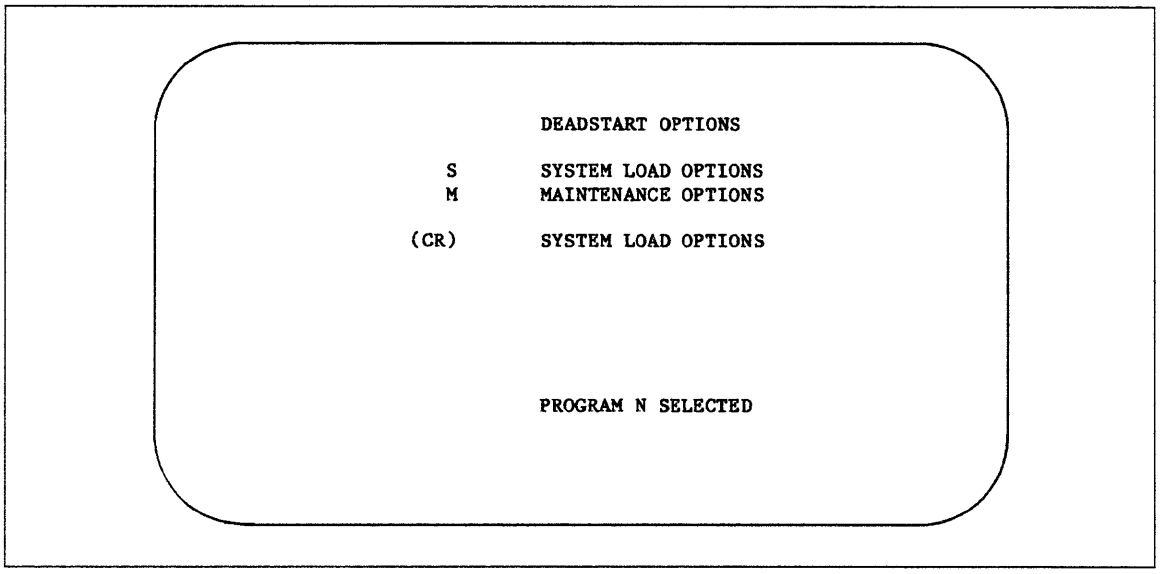


Figure 15-1. Deadstart Options Display

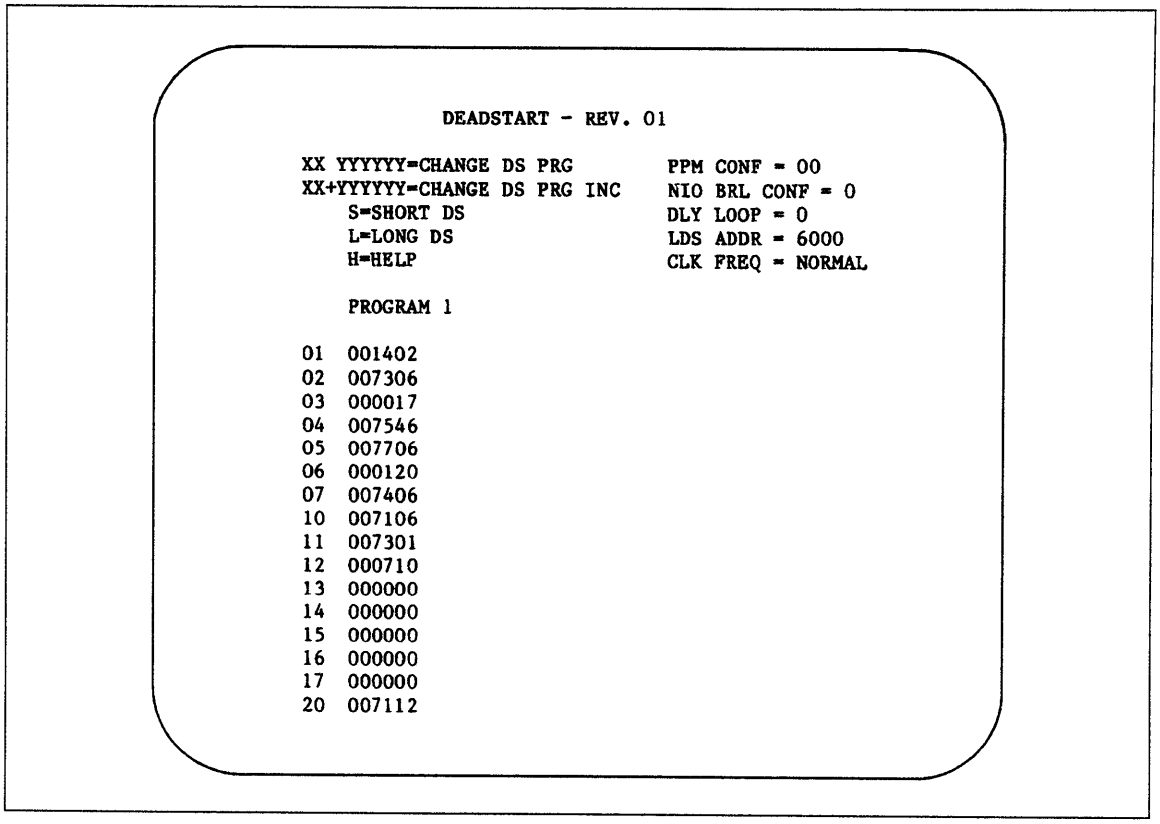


Figure 15-2. Initial Deadstart Display

Table 15-1. Deadstart Options Display

Option	Description
S	Selects a short deadstart sequence using the deadstart program identified at the bottom of the display. Upon completion of the deadstart sequence, a display for loading system software appears.
M	Causes the deadstart display to appear on the screen.

Table 15-2. Deadstart Display Operator Entries and Functions

Operator Entry	Function
xx yyyyyy	Enters a single word in the deadstart program at xx to a new value yyyyyy (octal).
xx+yyyyyy	Changes words in the deadstart program in sequence starting at xx.
S	Selects a short deadstart sequence.
L	Selects a long deadstart sequence.
H	Brings up a display that lists and explains all available commands. Refer to the hardware operator's guide for detailed information about these commands.

Central Memory Controls (CYBER 990E and 995E)

The CM contains a three-position COLUMN DEGRADE switch (figure 15-3). The switch is located on a switch box attached to the underside of the top panel over the CMC section.

The COLUMN DEGRADE switch forces CM address bit 37 either to a zero (switch down) or to a one (switch up). Refer to table 15-3.

In case of CM malfunctions, the remaining good memory can be reconfigured so it is accessible by contiguous addresses from 0 to the maximum remaining address. This is accomplished by setting the COLUMN DEGRADE switch as listed in table 15-3. Refer to the hardware operator's guide listed in the system publication index for further information.

In case of an emergency, use the system EMERGENCY OFF switch. The power-on and power-off procedures are described in the hardware operator's guide listed in the system publication index.

CAUTION

Improper application or removal of power may damage system circuits and/or air conditioning system. Power must be turned on/off by designated personnel only, except for the system EMERGENCY OFF switch. Use only for extreme emergency, not for normal shutdown.

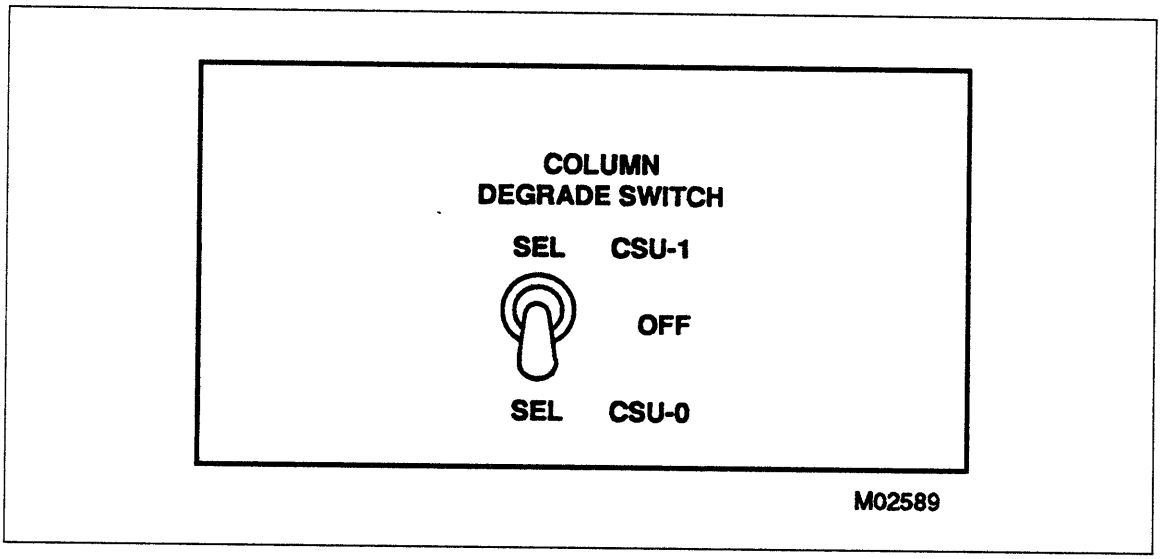


Figure 15-3. CM Configuration Switch

Central Memory Controls (CYBER 994)

The CYBER 994 uses a central memory composed of 256K chips. The 256K-chip memory cannot be reconfigured and, therefore, contains no controls.

Table 15-3. Central Memory Reconfiguration

Original CM		Reconfigured CM		
		Location of Failing CM		
Words (Size)	**Address Range**	**Words (Size)**	**RMA Bit 37**	**COLUMN DEGRADE Switch Setting**
10486K (80 MB)	0-17 777 777	8389K (64 MB)	0	SEL CSU-0
		2097K (16 MB)	1	SEL CSU-1
12583K (96 MB)	0-37 777 777	8389K (64 MB)	0	SEL CSU-0
		4194K (32 MB)	1	SEL CSU-1
14680K (112 MB)	0-77 777 777	8389K (64 MB)	0	SEL CSU-0
		6291K (48 MB)	1	SEL CSU-1
16177K (128 MB)	0-177 777 777	8389K (64 MB)	0	SEL CSU-0
		8389K (64 MB)	1	SEL CSU-1

Power-On and Power-Off Procedures

In case of an emergency, use the system EMERGENCY OFF switch. The power-on and power-off procedures are described in the hardware operator's guide listed in the system publication index.

CAUTION

Inproper application or removal of power may damage system circuits and/or air conditioning system. Power must be turned on/off by designated personnel only, except for the system EMERGENCY OFF switch. Use only for extreme emergency, not for normal shutdown.

Operating Procedures

Refer to the hardware operator's guide listed in the system publication index. The system is initialized by setting its deadstart display control parameters, and then by running either a long or short deadstart sequence (defined later in this section). After initialization, the keyboard is used to instruct the system further, under program control.

Control Checks

Before activating a long or short deadstart sequence, check the deadstart display parameters against their intended use. The normal settings of these parameters are as follows:

Parameter	Value
PPM CONF	00
BRL CONF	0
LDS ADDR	6000
Error messages	None

Deadstart Sequences

In response to a keyboard command (L or S) to the deadstart display, the IOU performs a deadstart sequence. Depending on the command (L or S), either the long or the short deadstart sequence is performed. The short deadstart sequence is used when hardware integrity verification is not required. The long deadstart sequence performs all the tasks performed by the short deadstart sequence and some additional tasks. The main additional task is the running of a diagnostic program, from a read-only memory (ROM) in the IOU, on logical PP0. The diagnostic program takes approximately 15 seconds to run.

Both deadstart sequences begin with a master clear which sets up all PPs, except logical PP0, for a 4096-word block input starting at PP location 0. The input into each PP is from the channel with the same number as the logical number of the PP concerned. The master clear also resets all external devices and sets maintenance channel connect code bit 52. The individual registers are set as follows.

Register	Initialization	Description
K	007100_8	Instruction display
P	007777_8	Causes block input to start from location 0
A	$10{,}000_8$	Count of 4096 words
Q	0, 1, 2...	I/O channel numbers (PP0: 0, PP1: 1, and so on)

All registers in both barrels are set to these values, except the registers of PP0.

If the long deadstart sequence is being performed, hardware clears location 7777_8 in all PP memories and sets the P register of PP0 to the value indicated by the parameter LDS ADDR = XXXX (normally 6000_8). PP0 starts performing a test program from a read-only memory in IOU. Hardware errors cause the LDS program to hang before completion. In the absence of errors, execution proceeds until the test program reaches location 7776_8. When this happens, the unique part of the long deadstart sequence ends with a master clear.

Next both deadstart sequences clear PP0 location 0, write the deadstart program on the display into PP0 memory locations 1 to 20_8, and clears PP0 location 21_8. PP0 then starts executing the program entered from the deadstart display (which is normally a bootstrap program to input more data from an assigned external device).

The short deadstart sequence does not disturb PP memory other than PP0 locations 0 to 21_8. Both deadstart sequences leave all PPs, except PP0, waiting for a block input or for action through the maintenance channel. After the block input is complete, each PP starts executing the program entered from whatever address was entered into location 0 of that PP.

IOU Reconfiguration

NOTE

Only PPs can be reconfigured in an AT511-A/AT512-A IOU. Ignore all mention of the RB X command and the BRL CONF parameter in the following disucssion if the IOU to be reconfigured has this equipment number(s). In tables 15-4 and 15-5, only the RB=0 examples are applicable.

The logical PP numbers and hardware are assigned to physical PPs circularly from the settings of IOU deadstart display PPM CONF and BRL CONF parameters, specifying which physical barrel and PPM is PP0. Maximum values for these parameters depend on the number of PPs installed. Illegal values entered in RB X and RP XX commands are rejected by the deadstart display and cause error messages to appear on the screen (refer to the hardware operator's guide). Reconfiguration is discussed in detail in the hardware operator's guide; allowable values for the PPM CONF and BRL CONF parameters and reconfiguration examples are shown in tables 15-4 and 15-5.

Table 15-4. PP and Barrel Reconfiguration Example, RP=0

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=0					
10	00	00	05		
	01	01	06		
	02	02	07	X	X
	03	03	10		
	04	04	11		
15	00	00	05	20	
	01	01	06	21	
	02	02	07	22	X
	03	03	10	23	
	04	04	11	24	
20	00	00	05	20	25
	01	01	06	21	26
	02	02	07	22	27
	03	03	10	23	30
	04	04	11	24	31

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 15-4. PP and Barrel Reconfiguration Example, RP=0 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=1					
10	00	05	00		
	01	06	01		
	02	07	02	X	X
	03	10	03		
	04	11	04		
15	00	20	00	05	
	01	21	01	06	
	02	22	02	07	X
	03	23	03	10	
	04	24	04	11	
20	00	25	00	05	20
	01	26	01	06	21
	02	27	02	07	22
	03	30	03	10	23
	04	31	04	11	24

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 15-4. PP and Barrel Reconfiguration Example, RP=0 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=2					
10	00				
	01				
	02	X	X	X	X
	03				
	04				
15	00	05	20	00	
	01	06	21	01	
	02	07	22	02	X
	03	10	23	03	
	04	11	24	04	
20	00	20	25	00	050
	01	21	26	01	061
	02	22	27	02	072
	03	23	30	03	103
	04	24	31	04	114

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 15-4. PP and Barrel Reconfiguration Example, RP=0 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=3					
10	00				
	01				
	02	X	X	X	X
	03				
	04				
15	00				
	01				
	02	X	X	X	X
	03				
	04				
20	00	05	20	25	00
	01	06	21	26	01
	02	07	22	27	02
	03	10	23	30	03
	04	11	24	31	04

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

Table 15-5. PP and Barrel Reconfiguration Example, RP=2

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=0					
10	00	03	10		
	01	04	11		
	02	00	05	X	X
	03	01	06		
	04	02	07		
15	00	03	10	23	
	01	04	11	24	
	02	00	05	20	X
	03	01	06	21	
	04	02	07	22	
20	00	03	10	23	30
	01	04	11	24	31
	02	00	05	20	25
	03	01	06	21	26
	04	02	07	22	27

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 15-5. PP and Barrel Reconfiguration Example, RP=2 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=1					
10	00	10	03		
	01	11	04		
	02	05	00	X	X
	03	06	01		
	04	07	02		
15	00	23	03	10	
	01	24	04	11	
	02	20	00	05	X
	03	21	01	06	
	04	22	02	07	
20	00	30	03	10	23
	01	31	04	11	24
	02	25	00	05	20
	03	26	01	06	21
	04	27	02	07	22

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 15-5. PP and Barrel Reconfiguration Example, RP=2 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=2					
10	00				
	01				
	02	X	X	X	X
	03				
	04				
15	00	10	23	03	
	01	11	24	04	
	02	05	20	00	X
	03	06	21	01	
	04	07	22	02	
20	00	23	30	03	10
	01	24	31	04	11
	02	20	25	00	05
	03	21	26	01	06
	04	22	27	02	07

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

(Continued)

Table 15-5. PP and Barrel Reconfiguration Example, RP=2 *(Continued)*

No. of PPs	Physical PPMs in each Barrel	BAR0	BAR1	BAR2	BAR3
Logical PP RB=3					
10	00				
	01				
	02	X	X	X	X
	03				
	04				
15	00				
	01				
	02	X	X	X	X
	03				
	04				
20	00	10	23	30	03
	01	11	24	31	04
	02	05	20	25	00
	03	06	21	26	01
	04	07	22	27	02

Notes:

X: Not applicable; results in message, Error BRL not installed.
RP: PP configuration.
RB: NIO barrel configuration only.
BAR0-3: Physical barrels.

Instruction Descriptions 16

CP Instruction Formats 16-1

Instructions Description Nomenclature 16-3

CP Operating Modes 16-4

CP Instruction Descriptions 16-4
- CP Integer Arithmetic Instructions 16-5
  - Integer Pack/Unpack 16-6
- CP Branch Instructions 16-9
  - Branch 16-9
- CP Block Copy Instructions 16-16
  - Block Copy 16-17
- CP Shift Instructions 16-19
  - Left Shift 16-19
  - Right Shift 16-21
- CP Logical Instructions 16-23
  - Logical Sum 16-24
  - Logical Difference 16-25
  - Logical Product 16-26
- CP Floating-Point Instructions 16-27
  - Floating Sum 16-28
  - Floating Difference 16-31
  - Floating Product 16-34
  - Floating Divide 16-37
- CP Jump Instructions 16-39
  - Jump 16-39
- CP Exchange Instructions 16-41
  - Exchange Jump 16-42
- CP Compare/Move Instructions 16-44
  - Transmit 16-44
  - Compare/Move 16-45
- CP Set Instructions 16-50
  - Set Ai 16-51
  - Set Bi 16-55
  - Set Xi 16-58
  - Read/Write 16-61
- CP Normalize Instructions 16-62
  - Normalize 16-63
  - Round Normalize 16-64
- CP Pass Instructions 16-65
  - Pass 16-65
- CP Illegal Instructions 16-66
  - Error Exit 16-66
  - Illegal Instruction 16-66
  - Illegal Read/Write 16-67
- CP Mask Instruction 16-68
  - Form Mask 16-68
- CP Pop Count Instruction 16-68
  - Population Count 16-68
- CP Read Free-Running Counter Instruction 16-69
  - Read Free-Running Counter 16-69

PP Instruction Descriptions 16-70
PP Instruction Formats 16-70
PP Data Format 16-72
PP Relocation Register Format 16-73
PP Load/Store Instructions 16-74
Load 16-74
Store 16-76
PP Arithmetic Instructions 16-78
Arithmetic Add 16-78
Arithmetic Subtract 16-80
PP Logical Instructions 16-82
Shift 16-82
Selective Clear 16-82
Logical Difference 16-83
Logical Product 16-85
PP Replace Instructions 16-86
Replace Add 16-86
Replace Subtract 16-89
PP Branch Instructions 16-90
Long Jump 16-90
Return Jump 16-91
Unconditional Jump 16-91
Zero/Nonzero Jump 16-92
Plus/Minus Jump 16-93
Jump to m 16-94
PP Central Memory Access Instructions 16-96
Central Read 16-96
Central Write 16-98
PP Input/Output Instructions 16-100
Test/Clear 16-101
Input/Output 16-102
Activate/Deactivate 16-104
Function 16-105
Other IOU Instructions 16-106
Pass 16-106
Exchange Jump 16-107
Instruction Execution Timing 16-108

Instruction Descriptions 16

This chapter contains the CYBER 170 State central processor (CP) instruction descriptions and peripheral processor (PP) instruction descriptions.

CP Instruction Formats

NOTE

CYBER 170 CP instructions use the rightmost 60 bits in the 64-bit word. The leftmost 4 bits are undefined. For these instructions, the most-significant bit is bit 59 and the least-significant bit is bit 0.

Program instruction words are divided into 15-bit fields called parcels. The first parcel (parcel 0) is the highest-order 15 bits of the 60-bit word. The second, third, and fourth parcels (parcels 1, 2, and 3) follow in order. Figure 16-1 shows possible parcel arrangements for instructions within a program instruction word.

An instruction may occupy one, two, or four parcels. This arrangement depends on the instruction format. When an instruction occupies two parcels, it must occupy two parcels within the same program word. A program word may be filled with a one-parcel pass instruction or an instruction acting as a two-parcel pass instruction. These instructions are used to fill a program word when necessary to place a particular instruction in the first parcel of a program word or to avoid starting a two-parcel instruction in the fourth parcel of a program word. Pass instructions may also be used for branch entry points because a branch instruction destination address must begin with a new word. One-parcel pass instructions are 460xx through 463xx. Instructions 60xxx through 62xxx may be used as two-parcel pass instructions by setting the i instruction designator to zero. Refer to table 16-1 for CP instruction designators.

CP instructions 011 and 012 have special properties. They are 60-bit double instructions that must start at parcel 0. The programmer has the option of providing a branch instruction at parcels 2 and 3 in the same instruction word (to an error-handling software routine) or filling this space with pass instructions. Refer to instructions 011 and 012.

Instructions 013 and 464 through 467 are 60-bit instructions which must start at parcel 0. They ignore any information in parcels 2 and 3; however, these parcels are normally set to all zeros.

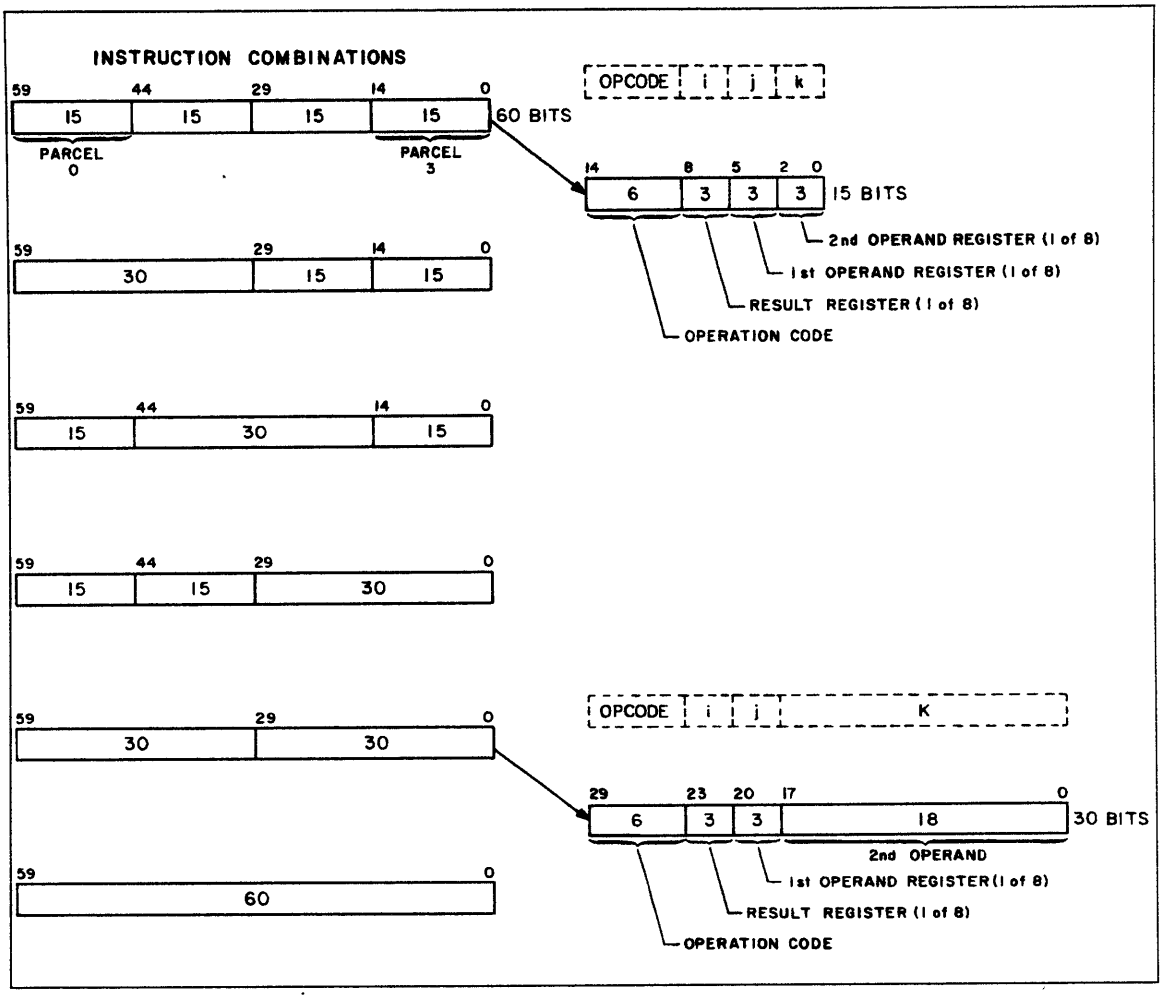


Figure 16-1. CP Instruction Parcel Arrangement

Instruction Description Nomenclature

The instruction descriptions in this chapter use the following instruction designators.

Table 16-1. CP Instruction Designators

Designator	Description
Opcode	6-bit/9-bit field specifying instruction operation code.
i,j,k	3-bit code specifying one of eight registers.
jk	6-bit code specifying amount of shift or mask.
K	18-bit operand or addresss.
x	Unused designator.
A	One of eight 18-bit address registers.
B	One of eight 18-bit index registers; B0 is fixed and equal to zero.
X	One of eight 60-bit operand registers.
()	Content of the word at a central memory address.
C1[1]	Offset (character address) of the first character in the first word of the source field.
C2[1]	Character address of the first character in the first word of the result field.
K1[1]	18-bit address indicating the central memory location of the first (leftmost) character of the source field.
K2[1]	18-bit address indicating the central memory location of the first (leftmost) character of the result field.
LL[1]	Lower 4 bits of the field length (character count) for a move or compare instruction; used with LU to specify field length.
LU[1]	Upper 9 bits of the field length (character count) for indirect move instructions or the upper 3 bits for direct instructions; used with LL to specify field length.

1. Applicable to compare/move instructions only.

CP Operating Modes

The CP executes instructions in CYBER 170 job mode, CYBER 170 monitor mode, and executive state. Changes between CYBER 170 job mode and CYBER 170 monitor mode are caused by CYBER 170 exchange jumps (CP instruction 013 and PP instructions 2600, 2610, and 2620). A hardware flag called the CYBER 170 monitor flag (MF) indicates whether the CP is in CYBER 170 job mode (flag is clear) or in CYBER 170 monitor mode (flag is set).

The executive state is invisible to the applications programmer. It sets up the CYBER 170 environment during initialization, executes certain instructions, and handles hardware-detected error conditions. Hardware-caused exchanges are called error exits. Most of these can be enabled or disabled by setting or clearing bits in the CYBER 170 exchange package. For further information on CP operating modes, refer to CYBER 170 Exchange Jump, Executive State, and Error Response in chapter 17.

CP Instruction Descriptions

The CP general instructions are divided into 16 subgroups as follows:

- Integer Arithmetic
- Branch
- Block Copy
- Shift
- Logical
- Floating Point
- Jump
- Exchange Jump
- Compare/Move
- Set
- Normalize
- Pass
- Illegal Instruction
- Mask
- Pop Count
- Read Free-Running Counter

CP Integer Arithmetic Instructions

The integer arithmetic instructions (table 16-2) perform integer arithmetic on signed twos complement words or halfwords in Xk or XkR. The sign bit is bit 0 for fullword integers or bit 32 for halfword integers.

Table 16-2. CP Integer Arithmetic Instructions

Opcode	Format	Instruction	Mnemonic
27	ijk	Pack (Xk) and (Bj) to Xi	PXi Bj Xk
26	ijk	Unpack (Xk) to Xi and Bj	UXi Bj Xk
36	ijk	Integer sum of (Xj) and (Xk) to Xi	IXi Xj+Xk
37	ijk	Integer difference of (Xj) and (Xk) to Xi	IXi Xj-Xk

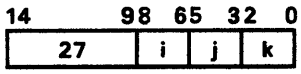
Integer Pack/Unpack

Opcode **27ijk**

Mnemonic PXi Bj, Xk

Instruction Pack (Xk) and (Bj) to Xi

Format



Remarks This instruction reads the contents of the Xk and Bj registers, packs them into a single word in floating-point format, and delivers this result to the Xi register. The coefficient for the value in the Xi register is obtained from the content of the Xk register, which is treated as a signed integer. The exponent for the value in the Xi register is obtained from the content of the Bj register, which is treated as a signed integer.

The lowest-order 48 bits in the Xi register are copied directly from the lowest-order 48 bits in the Xk register. The sign bit in the Xi register is copied directly from the sign bit in the Xk register. The exponent field in the Xi register is derived from the value in the Bj register by extracting the lowest-order 11 bits in the Bj register and modifying this quantity for exponent bias and coefficient sign.

Four sample sets of operands and packed results are listed in octal notation to illustrate the operation performed. These examples contain the four combinations of coefficient sign and exponent sign.

```
(Xk)  =  0000 4500 3333 2000 0077
(Bj)  =  00 0034
(Xi)  =  2034 4500 3333 2000 0077

(Xk)  =  0000 4500 3333 2000 0077
(Bj)  =  77 7743
(Xi)  =  1743 4500 3333 2000 0077

(Xk)  =  7777 3277 4444 5777 7700
(Bj)  =  00 0034
(Xi)  =  5743 3277 4444 5777 7700

(Xk)  =  7777 3277 4444 5777 7700
(Bj)  =  77 7743
(Xi)  =  6034 3277 4444 5777 7700
```

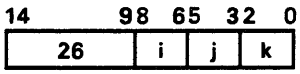
This instruction converts a number in fixed-point format to floating-point format. For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

Opcode **26ijk**

Mnemonic UXi Bj, Xk

Instruction Unpack (Xk) to Xi and Bj

Format



Remarks This instruction reads one operand from the Xk register, unpacks this word from floating-point format, and delivers the coefficient and exponents to the Xi and Bj registers, respectively. The 60-bit word delivered to the Xi register consists of the lowest 48 bits unaltered from the original operand plus the upper 12 bits, each equal to the original sign bit. This is a signed integer equal to the value of the coefficient in the original operand. The 18-bit quantity delivered to the Bj register is a signed integer equal to the value of the exponent in the original operand. The 11-bit exponent field in the operand is altered to remove the bias and then sign-extended to fill out the 18-bit quantity. The sign of the coefficient is removed in this process.

Four sample sets of operands and unpacked results are listed in octal notation to illustrate the operation performed. These examples contain the four combinations of coefficient sign and exponent sign.

```
(Xk)  =  2034 4500 3333 2000 0077
(Xi)  =  0000 4500 3333 2000 0077
(Bj)  =  00 0034

(Xk)  =  1743 4500 3333 2000 0077
(Xi)  =  0000 4500 3333 2000 0077
(Bj)  =  77 7743

(Xk)  =  5743 3277 4444 5777 7700
(Xi)  =  7777 3277 4444 5777 7700
(Bj)  =  00 0034

(Xk)  =  6034 3277 4444 5777 7700
(Xi)  =  7777 3277 4444 5777 7700
(Bj)  =  77 7743
```

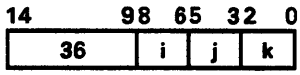
This instruction converts a number from floating-point format to fixed-point format. For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

Opcode **36ijk**

Mnemonic IXi Xj + Xk

Instruction Integer sum of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a 60-bit integer sum, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are signed integers. The resulting integer sum is delivered to the Xi register. Overflow is not detected.

This instruction adds integers too large for handling by 50 through 77 instructions. The instruction also merges and compares data fields during data processing.

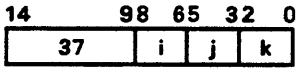
For further information, refer to Integer Arithmetic under CP Programming in chapter 17.

Opcode **37ijk**

Mnemonic IXi Xj - Xk

Instruction Integer difference of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a 60-bit integer difference, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are signed integers. The result of subtracting the quantity in the Xk register from the quantity in the Xj register is delivered to Xi. Overflow is not detected.

This instruction subtracts integers too large for handling by 50 through 77 instructions. The instruction also compares data fields during data processing.

For further information, refer to Integer Arithmetic under CP Programming in chapter 17.

CP Branch Instructions

The branch instructions (table 16-3) consist of both conditional and unconditional branch instructions. Each conditional branch instruction compares the contents of two general registers to determine whether a normal or a branch exit is taken.

Table 16-3. CP Branch Instructions

Opcode	Format	Instruction	Mnemonic
030	jK	Branch to K if (Xj) = 0	ZR
031	jK	Branch to K if (Xj) ≠ 0	NZ
032	jK	Branch to K if (Xj) is positive	PL
033	jK	Branch to K if (Xj) is negative	NG
034	jK	Branch to K if (Xj) is in range	IR
035	jK	Branch to K if (Xj) is out of range	OR
036	jK	Branch to K if (Xj) is definite	DF
037	jK	Branch to K if (Xj) is indefinite	ID
04	ijK	Branch to K if (Bi) = (Bj)	EQ
05	ijK	Branch to K if (Bi) ≠ (Bj)	NE
06	ijK	Branch to K if (Bi) ≥ (Bj)	GE
07	ijK	Branch to K if (Bi) < (Bj)	LT

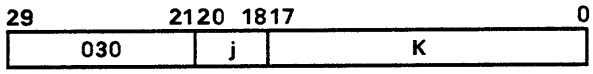
Branch

Opcode **030jK**

Mnemonic ZR Xj, K

Instruction Branch to K if (Xj) = 0

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on the content of the Xj register. The branch to address K occurs only on the following conditions. The current program sequence continues for all other cases.

Jump to K if: (Xj) = 0000 0000 0000 0000 0000 (positive zero)
Jump to K if: (Xj) = 7777 7777 7777 7777 7777 (negative zero)

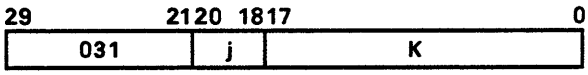
This instruction branches on a zero result from either a fixed-point or a floating-point operation.

Opcode **031jK**

Mnemonic NZ Xj, K

Instruction Branch to K if (Xj) ≠ 0

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on the content of the Xj register. The program sequence continues only on the following conditions. The branch to address K occurs for all other cases.

Continue if: (Xj) = 0000 0000 0000 0000 0000 (positive zero)
Continue if: (Xj) = 7777 7777 7777 7777 7777 (negative zero)

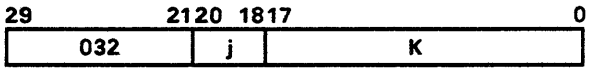
This instruction branches on a nonzero result from either a fixed-point or a floating-point operation.

Opcode **032jK**

Mnemonic PL Xj, K

Instruction Branch to K if (Xj) is positive

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on the content of the Xj register. The branch decision for this instruction is based on the value of the sign bit in the Xj register.

Jump to K if: Bit 59 of Xj = 0 (positive)
Continue if: Bit 59 of Xj = 1 (negative)

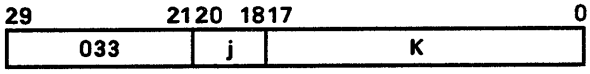
This instruction branches on a positive result from either a fixed-point or a floating-point operation.

Opcode **033jK**

Mnemonic NG Xj, K

Instruction Branch to K if (Xj) is negative

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on the content of the Xj register. The branch decision for this instruction is based on the value of the sign bit in the Xj register.

Jump to K if: Bit 59 of Xj = 1 (negative)
Continue if: Bit 59 of Xj = 0 (positive)

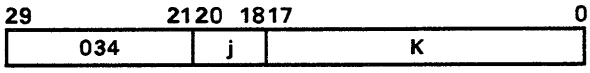
This instruction branches on a negative result from either a fixed-point or a floating-point operation.

Opcode **034jK**

Mnemonic IR Xj, K

Instruction Branch to K if (Xj) is in range

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on the content of the Xj register. The program sequence continues only on the following conditions. The branch to address K occurs for all other cases.

Continue if: (Xj) = 3777 xxxx xxxx xxxx xxxx (positive overflow)
Continue if: (Xj) = 4000 xxxx xxxx xxxx xxxx (negative overflow)

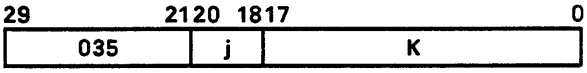
This instruction branches on a floating-point quantity within the floating-point range. The value of the coefficient is ignored in making this branch test. An underflow quantity is considered in range for purposes of this test.

Opcode **035jK**

Mnemonic OR Xj, K

Instruction Branch to K if (Xj) is out of range

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on the content of the Xj register. The branch to address K occurs only on the following conditions. The current program sequence continues for all other cases.

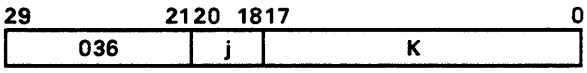
Jump to K if: (Xj) = 3777 xxxx xxxx xxxx xxxx (positive overflow)
Jump to K if: (Xj) = 4000 xxxx xxxx xxxx xxxx (negative overflow)

Opcode **036jK**

Mnemonic DF Xj, K

Instruction Branch to K if (Xj) is definite

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on the content of the Xj register. The program sequence continues only on the following conditions. The branch to address K occurs for all other cases.

Continue if: (Xj) = 1777 xxxx xxxx xxxx xxxx (positive indefinite)
Continue if: (Xj) = 6000 xxxx xxxx xxxx xxxx (negative indefinite)

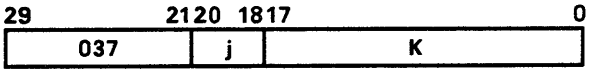
This instruction branches on a floating-point quantity that may be out of range but is still defined. The value of the coefficient is ignored in making this branch test. An overflow quantity or an underflow quantity is considered defined for purposes of this test.

Opcode **037jK**

Mnemonic ID Xj, K

Instruction Branch to K if (Xj) is indefinite

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on the content of the Xj register. The branch to address K occurs only on the following conditions. The current program sequence continues for all other cases.

Jump to K if: (Xj) = 1777 xxxx xxxx xxxx xxxx (positive indefinite)
Jump to K if: (Xj) = 6000 xxxx xxxx xxxx xxxx (negative indefinite)

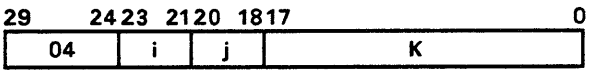
This instruction branches on a floating-point quantity that is not defined. The value of the coefficient is ignored in making this branch test. An overflow quantity or an underflow quantity is considered defined for purposes of this test.

Opcode **04ijK**

Mnemonic EQ Bi, Bj, K

Instruction Branch to K if (Bi) = (Bj)

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on a comparison of the contents of the Bi and Bj registers. The branch to address K occurs only if the two quantities are identical on a bit-by-bit comparison basis. The current program sequence continues for all other cases.

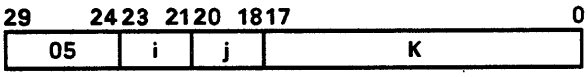
This instruction branches on an index equality test. A quantity consisting of all zeros and a quantity consisting of all ones are not equal for this test.

Opcode **05ijK**

Mnemonic NE Bi, Bj, K

Instruction Branch to K if (Bi) ≠ (Bj)

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on a comparison of the contents of the Bi and Bj registers. The program sequence continues only if the two quantities are identical on a bit-by-bit comparison basis. The branch to address K occurs for all other cases.

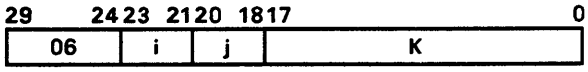
This instruction branches on an index inequality test. A quantity consisting of all zeros and a quantity consisting of all ones are not equal for this test.

Opcode **06ijK**

Mnemonic GE Bi, Bj, K

Instruction Branch to K if (Bi) ≥ (Bj)

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on a comparison of the contents of the Bi and Bj registers. Both quantities are treated as signed integers. The branch to address K occurs if the content of Bi is greater than or equal to the content of Bj. The current program sequence continues if the content of Bi is less than Bj.

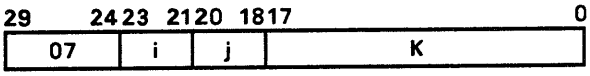
This instruction branches on an index threshold test. A positive zero quantity is considered greater than a negative zero quantity.

Opcode **07ijK**

Mnemonic LT Bi, Bj, K

Instruction Branch to K if (Bi) < (Bj)

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. Execution of this instruction causes the program sequence to terminate with a jump to address K in CM or to continue with the current program sequence, depending on a comparison of the contents of the Bi and Bj registers. Both quantities are treated as signed integers. The branch to address K occurs if the content of Bi is less than the content of Bj. The current program sequence continues if the content of Bi is greater than or equal to the content of Bj.

This instruction branches on an index threshold test. A positive zero quantity is considered greater than a negative zero quantity.

CP Block Copy Instructions

The block copy instructions (table 16-4) transfer 60-bit words between fields in CM and UEM.

Table 16-4. CP Block Copy Instructions

Opcode	Format	Instruction	Mnemonic
011	jK	Block copy (Bj + K) words from UEM to CM	RE Bj+K
012	jK	Block copy (Bj + K) words from CM to UEM	WE Bj+K

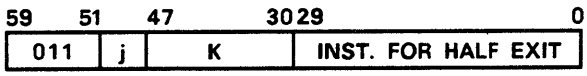
Block Copy

Opcode **011jK**

Mnemonic RE Bj + K

Instruction Block copy (Bj + K) words from UEM to CM

Format



Remarks This instruction copies a block of Bj plus K consecutive words from unified extended memory (UEM) to CM. The source UEM address is X0 plus RAE where the bits used depend on the setting of the expanded addressing select flag in the CYBER 170 exchange package. If the flag is clear (UEM is in standard addressing mode), the UEM address is calculated using bits 0 through 22 of X0; bits 24 through 59 are ignored. If the flag is set (UEM is in expanded addressing mode), the UEM address is calculated using bits 0 through 28 of X0; bits 30 through 59 are ignored.

The destination CM address is either A0 plus RAC, or X0 plus RAC, depending on the setting of the block copy flag in the CYBER 170 exchange package. When the block copy flag is clear, the CM address is A0 plus RAC. When the block copy flag is set, the CM address is calculated using bits 30 through 50 of X0. Bits 51 through 59 must be set to zero; results are undefined if these bits are not zero.

The operation leaves Bj, X0, and A0 unchanged. Bj and K are both signed 18-bit ones complement numbers, making it possible to transfer a maximum of 131 071 60-bit words. If Bj plus K is zero, the instruction acts as a 60-bit pass instruction.

If bit 21 or 22 of the result of X0 plus RAE is a one, zeros are transferred; and the next instruction is taken from parcel 2 of the same instruction word. If this is not the case, the next instruction is taken from parcel 0 of the next instruction word. If execution of the 011jK instruction is interrupted, it is restarted from the beginning.

This instruction is illegal if it does not start in parcel 0 or the UEM enable flag in the CYBER 170 exchange package is clear.

In standard addressing mode, 24 bits of X0 are checked against 23 bits of FLE with bit 23 of FLE equal to zero. In expanded addressing mode, 30 bits of X0 are checked against 29 bits of FLE with bit 29 equal to zero. If the X0 bits are greater than or equal to FLE, an address-out-of-range condition is detected.

If Bj plus K is negative, an address range error exit takes place. If the source field and the destination field overlap in physical memory, the final contents of the destination field are undefined.

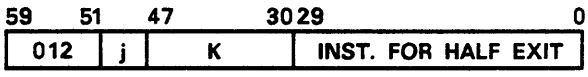
For further information, refer to Block Copy Instructions in chapter 17.

Opcode **012jK**

Mnemonic WE Bj + K

Instruction Block copy (Bj + K) words from CM to UEM

Format



Remarks This instruction copies a block of Bj plus K consecutive words from CM to UEM. The source CM address is either A0 plus RAC or X0 plus RAC, depending on the setting of the block copy flag in the CYBER 170 exchange package. When the block copy flag is clear, the CM address is A0 plus RAC. When the block copy flag is set, the CM address is calculated using bits 30 through 50 of X0. Bits 51 through 59 must be set to zero; results are undefined if these bits are not zero.

The destination UEM address is X0 plus RAE where the bits used depend on the setting of the expanded addressing select flag in the CYBER 170 exchange package. If the flag is clear (UEM is in standard addressing mode), the UEM address is calculated using bits 0 through 22 of X0; bits 24 through 59 are ignored. If the flag is set (UEM is in expanded addressing mode), the UEM address is calculated using bits 0 through 28 of X0; bits 30 through 59 are ignored.

The operation leaves Bj, X0, and A0 unchanged. Bj and K are both signed 18-bit ones complement numbers, making it possible to transfer a maximum of 131 071 60-bit words. If Bj plus K is zero, the instruction acts as a 60-bit pass instruction.

If bit 21 or 22 of the result of X0 plus RAE is a one, zeros are transferred; and the next instruction is taken from parcel 2 of the same instruction word. If this is not the case, the next instruction is taken from parcel 0 of the next instruction word. If execution of the 012jK instruction is interrupted, it is restarted from the beginning.

This instruction is illegal if it does not start in parcel 0 or the UEM enable flag in the CYBER 170 exchange package is clear.

In standard addressing mode, 24 bits of X0 are checked against 23 bits of FLE with bit 23 of FLE equal to zero. In expanded addressing mode, 30 bits of X0 are checked against 29 bits of FLE with bit 29 equal to zero. If the X0 bits are greater than or equal to FLE, an address-out-of-range condition is detected.

If Bj plus K is negative, an address range error exit takes place. If the source field and the destination field overlap in physical memory, the final contents of the destination field are undefined.

For further information, refer to Block Copy Instructions in chapter 17.

CP Shift Instructions

The shift instructions (table 16-5) shift the Xi 60-bit word through the number of bit positions determined from a computed shift count.

Table 16-5. CP Shift Instructions

Opcode	Format	Instruction	Mnemonic
20	ijk	Left shift (Xi) by jk	LXi jk
22	ijk	Left shift (Xk) nominally (Bj) places to Xi	LXi Bj Xk
21	ijk	Right shift (Xi) by jk	AXi jk
23	ijk	Right shift (Xk) nominally (Bj) places to Xi	AXi Bj Xk

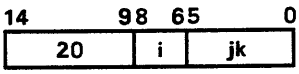
Left Shift

Opcode **20ijk**

Mnemonic LXi jk

Instruction Left shift (Xi) by jk

Format



Remarks This instruction reads one operand from Xi, shifts the 60-bit word left circularly by jk bit positions, and writes the resulting 60-bit word back into the same Xi register. The j and k designators are treated as a single 6-bit positive integer operand in this instruction.

A left-circular shift implies that the bit pattern in the 60-bit word is displaced towards the highest-order bit positions. The bits shifted off the upper end of the 60-bit word are inserted in the lowest-order bit positions in the same sequence. The resulting 60-bit word has the same quantity of bits with values of 1 and 0 as in the original operand.

A sample computation is listed in octal notation to illustrate the operation performed.

Initial (Xi) = 2323 6600 0000 0000 0111

jk = 12 (octal)

Final (Xi) = 7540 0000 0000 0022 2464

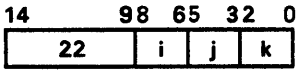
This instruction, together with instruction 21, may be used whenever a data word is to be shifted by a predetermined amount. If the amount of shift is derived in the execution of the program, use instruction 22 or 23.

Opcode **22ijk**

Mnemonic LXi Bj, Xk

Instruction Left shift (Xk) nominally (Bj) places to Xi

Format



Remarks This instruction reads a 60-bit operand from Xk, shifts the data either left or right as specified by Bj, and writes the resulting 60-bit word into Xi. If the value in Bj is positive, the data is left-shifted circularly the number of bit positions designated by the value in Bj. If the value in Bj is negative, the data is right-shifted with sign extension the number of bit positions designated by the value in Bj. Bj bit 17 determines the sign of Bj.

A left-circular shift implies that the bit pattern in the 60-bit word is displaced towards the highest-order bit positions. The bits shifted off the upper end are inserted in the lowest-order bit positions in the same sequence. The resulting 60-bit word has the same quantity of bits with values of 1 and 0 as in the original operand.

A right shift with sign extension implies that the bit pattern in the 60-bit word is displaced towards the lowest-order positions. The bits shifted off the lower end are discarded. The highest-order bit positions are filled with copies of the original sign bit.

Two sample computations are listed in octal notation to illustrate the operation performed. An example of a positive shift count resulting in a left-circular shift is as follows:

(Xk) = 2323 6600 0000 0000 0111

(Bj) = 00 0012

(Xi) = 7540 0000 0000 0022 2464

An example of the right shift with sign extension is as follows:

(Xk) = 1327 6000 0000 3333 2422

(Bj) = 77 7771

(Xi) = 0013 2760 0000 0033 3324

If Bj bits 6 through 10 are different from Bj bit 17 and Bj bit 17 is set, the shift count is greater than 63 (decimal) places right, and a result of positive zero is returned to Xi. Bj bits 11 through 16 are not tested by this instruction.

This instruction is used when the amount of shift is derived in the computation. The instruction is also used for correcting the coefficient of a floating-point number when the exponent has been unpacked into a B register.

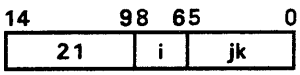
Right Shift

Opcode **21ijk**

Mnemonic AXi jk

Instruction Right shift (Xi) by jk

Format



Remarks This instruction reads one operand from Xi, shifts the 60-bit word right with sign extension by jk bit positions, and writes the resulting 60-bit word back into the same Xi register. The j and k designators are treated as a single 6-bit positive integer operand in this instruction.

A right shift with sign extension implies that the bit pattern in the 60-bit word is displaced toward the lowest-order bit positions. The bits shifted off the lower end of the word are discarded. The highest-order bit positions are filled with copies of the original sign bit.

Two sample computations are listed in octal notation to illustrate the operation performed. An example of a positive operand is as follows:

Initial (Xi) = 2004 7655 0002 3400 0004

jk = 30 (octal)

Final (Xi) = 0000 0000 2004 7655 0002

An example of a negative operand is as follows:

Initial (Xi) = 6000 4420 2222 0000 5643

jk = 10 (octal)

Final (Xi) = 7774 0011 0404 4440 0013

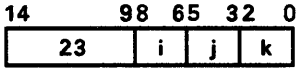
This instruction, together with instruction 20, may be used whenever a data word is to be shifted by a predetermined amount. If the amount of shift is derived in the execution of the program, use instruction 22 or 23.

Opcode **23ijk**

Mnemonic AXi Bj, Xk

Instruction Right shift (Xk) nominally (Bj) places to Xi

Format



Remarks This instruction reads a 60-bit operand from Xk, shifts the data either left or right as specified by the content of Bj, and writes the resulting 60-bit word into Xi. If the value in Bj is positive, the data is right-shifted with sign extension the number of bit positions designated by the value in Bj. If the value in Bj is negative, the data is left-shifted circularly the number of bit positions designated by the value in Bj. Bj bit 17 determines the sign of Bj.

A left-circular shift implies that the bit pattern in the 60-bit word is displaced towards the highest-order bit positions. The bits shifted off the upper end are inserted in the lowest-order bit positions in the same sequence. The resulting 60-bit word has the same quantity of bits with values of 1 and 0 as in the original operand.

A right shift with sign extension implies that the bit pattern in the 60-bit words is displaced towards the lowest-order bit positions. The bits shifted off the lower end of the word are discarded. The highest-order bit positions are filled with copies of the original sign bit.

Two sample computations are listed in octal notation to illustrate the operation performed. The following example contains a positive shift count resulting in a right shift with sign extension.

(Xk) = 1327 6000 0000 3333 2422

(Bj) = 00 0006

(Xi) = 0013 2760 0000 0033 3324

The following example contains a negative shift count resulting in a left-circular shift.

(Xk) = 2323 6600 0000 0000 0111

(Bj) = 77 7765

(Xi) = 7540 0000 0000 0022 2464

If Bj bits 6 through 10 are different from Bj bit 17, and Bj bit 17 is clear, the shift count is greater than 63 (decimal) places right, and a result of positive zero is returned to Xi. This instruction does not test Bj bits 11 through 16.

This instruction is used when the amount of shift is derived in the computation. The instruction is also used for correcting the coefficient of a floating-point number when the exponent has been unpacked into a B register.

CP Logical Instructions

The logical instructions (table 16-6) perform logical (Boolean) operations in the X registers.

Table 16-6. CP Logical Instructions

Opcode	Format	Instruction	Mnemonic
12	ijk	Logical sum of (Xj) and (Xk) to Xi	BXi Xj+Xk
16	ijk	Logical sum of (Xj) with complement of (Xk) to Xi	BXi -Xk+Xj
13	ijk	Logical difference of (Xj) and (Xk) to Xi	BXi Xj-Xk
17	ijk	Logical difference of (Xj) with complement of (Xk) to Xi	BXi -Xk-Xj
11	ijk	Logical product of (Xj) and (Xk) to Xi	BXi Xj*Xk
15	ijk	Logical product of (Xj) with complement of (Xk) to Xi	BXi -Xj*Xj

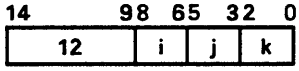
Logical Sum

Opcode **12ijk**

Mnemonic BXi Xj + Xk

Instruction Logical sum of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a result, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. The result delivered to Xi is the bit-by-bit logical sum of the two operands. Each of the 60 bits in Xj is compared with the corresponding bit in Xk to form a single bit in Xi. A sample computation is listed in octal notation to illustrate the operation performed and includes the four possible bit combinations that may occur.

(Xj) = 0000 7777 0123 4567 1010

(Xk) = 0123 4567 7777 0000 1100

(Xi) = 0123 7777 7777 4567 1110

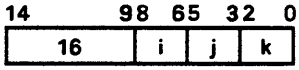
This instruction merges portions of a 60-bit word into a composite word during data processing.

Opcode **16ijk**

Mnemonic BXi -Xk + Xj

Instruction Logical sum of (Xj) with complement of (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a result, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. The result delivered to Xi is the bit-by-bit logical sum of the value in Xj and the complement of the value in Xk. Each of the 60 bits in Xj is compared with the corresponding bit in Xk to form a single bit in Xi. A sample computation is listed in octal notation to illustrate the operation performed and includes the four possible bit combinations that may occur.

(Xj) = 0000 7777 0123 4567 1010

(Xk) = 0123 4567 7777 0000 1100

(Xi) = 7654 7777 0123 7777 7677

This instruction merges portions of a 60-bit word into a composite word during data processing.

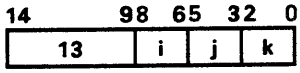
Logical Difference

Opcode **13ijk**

Mnemonic BXi Xj -Xk

Instruction Logical difference of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a result, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. The result delivered to Xi is the bit-by-bit logical difference of the two operands. Each of the 60 bits in Xj is compared with the corresponding bit in Xk to form a single bit in Xi. A sample computation is listed in octal notation to illustrate the operation performed and includes the four possible bit combinations that may occur.

(Xj) = 0123 7777 0123 4567 1010

(Xk) = 0123 4567 7777 3210 1100

(Xi) = 0000 3210 7654 7777 0110

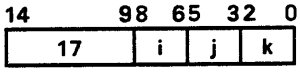
This instruction compares bit patterns or complements bit patterns during data processing.

Opcode **17ijk**

Mnemonic BXi -Xk - Xj

Instruction Logical difference of (Xj) with complement of (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a result, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. The result delivered to Xi is the bit-by-bit logical difference of the value in Xj and the complement of the value in Xk. Each of the 60 bits in Xj is compared with the corresponding bit in Xk to form a single bit in Xi. A sample computation is listed in octal notation to illustrate the operation performed and includes the four possible combinations that may occur.

(Xj) = 0123 7777 0123 4567 1010

(Xk) = 0123 4567 7777 3210 1100

(Xi) = 7777 4567 0123 0000 7667

This instruction compares bit patterns or complements bit patterns during data processing.

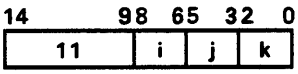
Logical Product

Opcode **11ijk**

Mnemonic BXi Xj * Xk

Instruction Logical product of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a result, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. The result delivered to Xi is the bit-by-bit logical product of the two operands. Each of the 60 bits in Xj is compared with the corresponding bit in Xk to form a single bit in Xi. A sample computation is listed in octal notation to illustrate the operation performed and includes the four possible bit combinations that may occur.

(Xj) = 7777 7000 0123 4567 1010

(Xk) = 0123 4567 0077 7700 1100

(Xi) = 0123 4000 0023 4500 1000

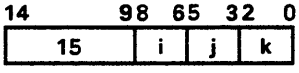
This instruction extracts portions of a 60-bit word during data processing.

Opcode **15ijk**

Mnemonic BXi -Xk * Xj

Instruction Logical product of (Xj) with complement of (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a result, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. The result delivered to Xi is the bit-by-bit logical product of the value in Xj and the complement of the value in Xk. Each of the 60 bits in Xj is compared with the corresponding bit in Xk to form a single bit in Xi. A sample computation is listed in octal notation to illustrate the operation performed and includes the four possible bit combinations that may occur.

(Xj) = 7777 7000 0123 4567 1010

(Xk) = 0123 4567 0007 7700 1100

(Xi) = 7654 3000 0120 0067 0010

This instruction extracts portions of a 60-bit word during data processing.

CP Floating-Point Arithmetic Instructions

The floating-point instructions (table 16-7) perform arithmetic operations on floating-point numbers.

Table 16-7. CP Floating-Point Instructions

Opcode	Format	Instruction	Mnemonic
30	ijk	Floating sum of (Xj) and (Xk) to Xi	FXi Xj+Xk
32	ijk	Floating double-precision sum of (Xj) and (Xk) to Xi	DXi Xj+Xk
34	ijk	Round floating sum of (Xj) and (Xk) to Xi	RXi Xj+Xk
31	ijk	Floating difference of (Xj) and (Xk) to Xi	FXi Xj-Xk
33	ijk	Floating double-precision difference of (Xj) and (Xk) to Xi	DXi Xj-Xk
35	ijk	Round floating difference of (Xj) and (Xk) to Xi	RXi Xj-Xk
40	ijk	Floating product of (Xj) and (Xk) to Xi	FXi Xj*Xk
41	ijk	Round floating product of (Xj) and (Xk) to Xi	RXi Xj*Xk
42	ijk	Floating double-precision product of (Xj) and (Xk) to Xi	DXi Xj*Xk
44	ijk	Floating divide (Xj) by (Xk) to Xi	FXi Xj/Xk
45	ijk	Round floating divide (Xj) by (Xk) to Xi	RXi Xj/Xk

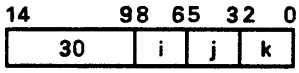
Floating Sum

Opcode **30ijk**

Mnemonic FXi Xj + Xk

Instruction Floating sum of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a floating-point sum, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format and are not necessarily normalized. The sum of the quantities in Xj and Xk is delivered to Xi in floating-point format and is not necessarily normalized.

The two operands are unpacked from floating-point format, and the exponents are compared. The coefficient with the smaller exponent is right-shifted by the difference of the two exponents such that both coefficients are the same significance. The two coefficients are then added to form a 96-bit result. The upper half of the result is then selected as a coefficient and packed along with the larger exponent to form the result sent to Xi. If coefficient overflow occurs, the sum is right-shifted one place, and the exponent is increased by one.

If the two operands have unlike signs, the result coefficient may have leading zeros. No normalize operation is built into this instruction to correct this situation. A separate normalize instruction must be programmed if the result is to be kept in a normalized form.

When the difference between the exponents is greater than 128 (decimal), the shifted sign bit is extended to the entire shifted operand. Infinite (3777xxx...x or 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands cause corresponding exit conditions to set in the CP for exit mode action.

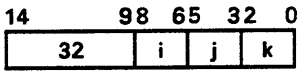
For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

Opcode **32ijk**

Mnemonic DXi Xj + Xk

Instruction Floating double-precision sum of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers; operates on them to form a double-precision, floating-point sum; and delivers the lower half of this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format and are not necessarily normalized. The sum of the quantities in Xj and Xk is delivered to Xi in floating-point format and is not necessarily normalized.

The two operands are unpacked from floating-point format, and the exponents are compared. The coefficient with the smaller exponent is right-shifted by the difference of the two exponents such that both coefficients are the same significance. The two coefficients are then added to form a 96-bit result. The lower half of the result is then selected and packed along with the larger exponent minus 48 (decimal) to form the result sent to Xi. If coefficient overflow occurs, the result is right-shifted by one place, and the exponent is increased by one. Infinite (3777xxx...x or 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands cause corresponding exit conditions to set in the CP for exit mode action.

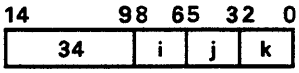
For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

Opcode **34ijk**

Mnemonic RXi Xj + Xk

Instruction Round floating sum of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a rounded floating-point sum, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format and are not necessarily normalized. The result is delivered to Xi in floating-point format and is not necessarily normalized.

The round floating-point sum is a single-precision, floating-point sum with a round bit (or bits) inserted before the add operation takes place. A round bit is always inserted in the coefficient with the larger exponent. If the two exponents are equal, the round bit is inserted in the coefficient for Xk. The round bit is equal to the complement of the sign bit and is inserted immediately to the right of the lowest-order bit in the coefficient. This has the effect of increasing the magnitude of the coefficient by one-half of the least-significant bit. A second round bit is inserted in a corresponding manner to the other coefficient if both operands are normalized or have unlike signs. The second round bit is inserted before the coefficient is shifted by the difference of the exponents. Infinite (3777xxx...x or 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands cause corresponding exit conditions to set in the CP for exit mode action.

For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

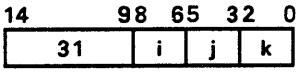
Floating Difference

Opcode **31ijk**

Mnemonic FXi Xj - Xk

Instruction Floating difference of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a floating-point difference, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format and are not necessarily normalized. The result of subtracting the quantity in Xk from the quantity in Xj is delivered to Xi in floating-point format and is not necessarily normalized.

The two operands are unpacked from floating-point format, and the exponents are compared. The coefficient with the smaller exponent is right-shifted by the difference of the two exponents such that both coefficients are the same significance. The Xk coefficient is then subtracted from the Xj coefficient to form a 96-bit result. The upper half of the result is then selected and packed along with the larger exponent to form the result sent to Xi. If coefficient overflow occurs, the result is right-shifted one place, and the exponent is increased by one.

If the two operands have like signs, the result coefficient may have leading zeros. No normalize operation is built into this instruction to correct this situation. A separate normalize instruction must be programmed if the result is to be kept in a normalized form. Infinite (3777xxx...x or 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands cause corresponding exit conditions to set in the CP for exit mode action.

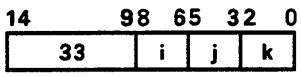
For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

Opcode **33ijk**

Mnemonic DXi Xj - Xk

Instruction Floating double-precision difference of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers; operates on them to form a double-precision, floating-point difference; and delivers the lower half of this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format and are not necessarily normalized. The result of subtracting the quantity in Xk from the quantity in Xj is delivered to Xi in floating-point format and is not necessarily normalized.

The two operands are unpacked from floating-point format, and the exponents are compared. The coefficient with the smaller exponent is right-shifted by the difference of the two exponents such that both coefficients are the same significance. The Xk coefficient is then subtracted from the Xj coefficient to form a 96-bit result. The lower half of the result is then selected and packed along with the larger exponent minus 48 (decimal) to form the result sent to Xi. If coefficient overflow occurs, the result is right-shifted one place, and the exponent is increased by one.

Infinite (3777xxx...x or 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands cause corresponding exit conditions to set in the CP for exit mode action.

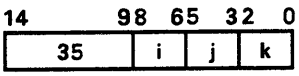
For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

Opcode **35ijk**

Mnemonic RXi Xj - Xk

Instruction Round floating difference of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a rounded floating-point difference, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format and are not necessarily normalized. The result of subtracting the quantity in Xk from the quantity in Xj is delivered to Xi in floating-point format and is not necessarily normalized.

The round floating-point difference is a single-precision, floating-point difference with a round bit (or bits) inserted before the subtract operation takes place. A round bit is always inserted in the coefficient with the larger exponent. If the two exponents are equal, the round bit is added to the coefficient for Xk. The round bit is equal to the complement of the sign bit and is inserted immediately to the right of the lowest-order bit in the coefficient. This has the effect of increasing the magnitude of the coefficient by one-half of the least-significant bit. A second round bit is inserted in a corresponding manner to the other coefficient if both operands are normalized or have like signs. The second round bit is inserted before the coefficient is shifted by the difference of the exponents. Infinite (3777xxx...x or 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands cause corresponding exit conditions to set in the CP for exit mode action.

For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

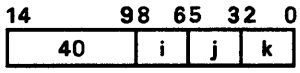
Floating Product

Opcode **40ijk**

Mnemonic FXi Xj * Xk

Instruction Floating product of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a floating-point product, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format and are not necessarily normalized. The result is delivered to Xi in floating-point format. If both operands are normalized, the result is also normalized. If both operands are not normalized, the result is not normalized.

The two operands are unpacked from floating-point format. The exponents are added with a correction factor to determine the exponent for the result. The coefficients are multiplied as signed integers to form a 96-bit integer product. The upper half of this product is extracted to form the coefficient for the result. If the original operands are normalized and the product has only 95 significant bits, a 1-bit left shift is done to normalize the result coefficient. The resulting exponent is reduced by one count in this case.

If both operands are not normalized, the resulting double-precision product has less than 96 significant bits. No test is made for the position of the most-significant bit. The upper 48 bits are read from the double-precision product register. Leading zeros occur in this result coefficient.

This instruction is used in floating-point calculations where rounding of operands is not desired, such as in multiple-precision arithmetic and in calculations involving error analysis. Infinite (3777xxx...x or 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands cause corresponding exit conditions to set in the CP for exit mode action.

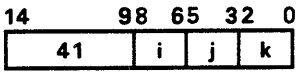
For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

Opcode **41ijk**

Mnemonic RXi Xj * Xk

Instruction Round floating product of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a rounded floating-point product, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format and are not necessarily normalized. The result is delivered to Xi in floating-point format. If both operands are normalized, the result is also normalized. If both operands are not normalized, the result is not normalized.

The two operands are unpacked from floating-point format. The exponents are added with a correction factor to determine the exponent for the result. The coefficients are multiplied as signed integers to form a 96-bit integer product. A rounding bit is added to bit position 46 of this product. The upper half of this product is extracted to form the coefficient for the result. If the original operands are normalized and the product has only 95 significant bits, a 1-bit left shift is done to normalize the result coefficient. The resulting exponent is reduced by one count in this case.

If both operands are not normalized, the resulting double-precision product has less than 96 significant bits. No test is made for the position of the most-significant bit. The upper 48 bits are read from the double-precision product register. Leading zeros occur in this result coefficient.

This instruction is used in single-precision, floating-point calculations. For multiple-precision calculations, the 40 and 42 instructions must be used. Infinite (3777xxx...x or 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands cause corresponding exit conditions to set in the CP for exit mode action.

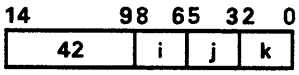
For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

Opcode **42ijk**

Mnemonic DXi Xj * Xk

Instruction Floating double-precision product of (Xj) and (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers; operates on them to form a double-precision, floating-point product; and delivers the lower half of this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format and are not necessarily normalized. The lower half of the double-precision product is delivered to Xi in floating-point format and is not necessarily normalized.

The operands are not rounded in this operation. The two operands are unpacked from floating-point format. The exponents are added to determine the exponent for the result. The result exponent is exactly 48 less than the exponent for a 40 instruction. The coefficients are multiplied as signed integers to form a 96-bit integer product. The lower half of this product is extracted to form the coefficient for the result. If the original operands are normalized and the double-precision product has only 95 significant bits, a 1-bit left shift is done to normalize the result coefficient. The resulting exponent is reduced by one count in this case.

If both operands are not normalized, the resulting double-precision product has less than 96 significant bits. No test is made for the position of the most-significant bit. The lower 48 bits are always read from the 96-bit product register.

This instruction is used in multiple-precision, floating-point calculations. This instruction also provides for integer multiplication capabilities where both operands have an exponent value of plus or minus zero, and neither coefficient has been normalized. The integer result sent to Xi is 48 bits with 60-bit sign extension. If the result exceeds 48 bits, the hardware does not detect an overflow. An overflow check can be made by executing a 40 instruction using the same two operands. If the result is nonzero, overflow is then indicated. An integer multiply operation is not intended for use with normalized operands. Infinite (3777xxx...x and 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands cause corresponding exit conditions to set in the CP for exit mode action.

For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

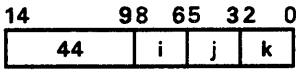
Floating Divide

Opcode **44ijk**

Mnemonic FXi Xj/Xk

Instruction Floating divide (Xj) by (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a floating-point quotient, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format. The result of dividing the content of Xj by the content of Xk is delivered to Xi. If both operands are normalized, the quotient is also normalized. The remainder from the division process is discarded.

The two operands are unpacked from floating-point format. The exponents are subtracted with a correction factor to determine the exponent for the result. The coefficient from Xj is positioned in a dividend register. The coefficient from Xk is trial-subtracted repeatedly from the dividend. The quotient bits are assembled in a quotient register. When 48 bits of the quotient are assembled, they are packed with the result exponent into floating-point format and delivered to Xi.

If the exponent subtraction causes an underflow or overflow, an underflow or overflow result is returned even with the occurrence of a divide fault.

If the dividend is not normalized, the quotient cannot be normalized. However, the quotient is correct even though there may be leading zeros in the coefficient. If the divisor is not normalized, the quotient may be incorrect. If the coefficient for the content of Xj is larger than the coefficient for the content of Xk by a factor of two or more, a divide fault causes an indefinite result to be returned to Xi.

This instruction is used in floating-point calculations where rounding of operands is not desired. In multiple-precision division, this instruction must be followed by a multiplication of the quotient by the divisor and subtracted from the dividend to reconstruct the remainder.

If infinite (3777xxx...x or 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands are used, corresponding exit conditions are set in the CP for exit mode action.

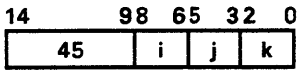
For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

Opcode **45ijk**

Mnemonic RXi Xj/Xk

Instruction Round floating divide (Xj) by (Xk) to Xi

Format



Remarks This instruction reads operands from two X registers, operates on them to form a rounded floating-point quotient, and delivers this result to a third X register. The operands for this instruction are in the Xj and Xk registers. These operands are in floating-point format. The result of dividing the content of Xj by the content of Xk is delivered to Xi. If both operands are normalized, the quotient is also normalized. The remainder from the division process is discarded.

The two operands are unpacked from floating-point format in this operation. The exponents are subtracted with a correction factor to determine the exponent for the result. The coefficient from Xj is positioned in a dividend register. The Xj quantity is modified by inserting a 2525...25 round pattern below the lowest-order bit of the dividend coefficient. The coefficient from Xk is trial-subtracted repeatedly from the dividend. The quotient bits are assembled in a quotient register. When 48 bits of the quotient are assembled, they are packed with the result exponent into floating-point format and delivered to Xi.

If the dividend is not normalized, the quotient cannot be normalized. However, the quotient is correct even though there may be leading zeros in the coefficient. If the divisor is not normalized, the quotient may be incorrect. If the coefficient for the content of Xj is larger than the coefficient for the content of Xk by a factor of two or more, a divide fault occurs. A divide fault causes an indefinite result to be returned to Xi.

This instruction is used in single-precision, floating-point calculations where rounding of operands is desired to reduce truncation errors.

If infinite (3777xxx...x or 4000xxx...x) or indefinite (1777xxx...x or 6000xxx...x) operands are used, corresponding exit conditions are set in the CP for exit mode action.

For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

CP Jump Instructions

The jump instructions (table 16-8) allow departure from sequential instruction execution.

Table 16-8. CP Jump Instructions

Opcode	Format	Instruction	Mnemonic
010	xK	Return jump to K	RJ
02	ixK	Jump to (Bi) + K	JP

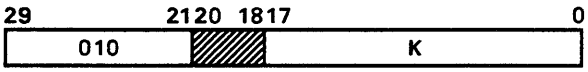
Jump

Opcode **010xK**

Mnemonic RJ K

Instruction Return jump to K

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. This instruction writes a special word into CM at relative address K. The current program sequence then terminates by a jump to address K plus one. The word stored in memory contains a jump instruction which causes an unconditional jump to the address of this return jump instruction plus 1.

This instruction calls a subroutine and inserts execution of the subroutine between execution of this instruction word and the following instruction word. Instructions appearing after the return jump instruction in the instruction word are not executed. The called subroutine exit must be at address K. The called subroutine entrance address must be K plus 1.

This instruction stores a 60-bit word at address K in memory. The upper half of this word contains an unconditional jump (0400) instruction with an address that is equal to the current program address plus 1. The lower half of the stored word is all zeros. The octal digits in the stored word then appear as illustrated with the x field indicating the location of the current program address plus 1.

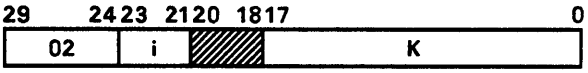
K	0400x	xxxxx	00000	00000	Subroutine exit
K + 1	yyyyy	yyyyy	yyyyy	yyyyy	Subroutine entrance

Opcode **02ixK**

Mnemonic JP Bi + K

Instruction Jump to (Bi) + K

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. The instruction causes the current program sequence to terminate with a jump to address Bi plus K in CM.

This instruction allows computed branch point destinations. This is the only instruction in which a computed parameter can specify a program branch destination address. All other jump instructions have preassigned destination addresses.

The quantities in Bi and operand K are added in an 18-bit ones complement mode. The result is treated as an 18-bit positive integer that specifies the beginning address in CM for the new program sequence. The remaining instructions, if any, in the instruction word do not execute.

CP Exchange Jump Instructions

The exchange jump instructions (table 16-9) exchange the current process registers (formatted as an exchange package) with another set stored in CM, and do the following:

- When executed with CP in Virtual State monitor mode, the processor switches from monitor to job mode.
- When executed in Virtual State job mode, the processor switches from mob to monitor mode and the system call bit sets in the monitor condition register (MCR 10).

In either case, the P register stored in the outgoing exchange package points to the next instruction that would have executed if the exchange had not occurred.

This instruction can cause the following exception conditions.

- Environment specification error.
- System call.

Refer to chapter 17 for programming information.

Table 16-9. CP Exchange Jump Instructions

Opcode	Format	Instruction	Mnemonic
013	jK	Central exchange jump to (Bj) +K (CYBER 170 monitor flag set)	XJ Bj+K
013	xx	Monitor exchange jump to MA (CYBER 170 monitor flag clear)	XJ

Exchange Jump

Opcode **013jK**

Mnemonic XJ Bj + K

Instruction Central exchange jump to (Bj) + K (CYBER 170 MF set)

Format

Remarks This instruction must start at parcel 0. Also, a CYBER 170 exchange package must be ready at address Bj plus K or at address MA.

This instruction stores P plus 1 into the outgoing CYBER 170 exchange package in hardware and then exchanges this CYBER 170 exchange package with the CYBER 170 exchange package stored in memory. If the CYBER 170 MF is set at the beginning of the instruction, the incoming CYBER 170 exchange package starts at absolute address Bj plus K. If the CYBER 170 MF is clear at the beginning, then the j and K fields of the instruction are ignored; and the incoming CYBER 170 exchange package starts at absolute address MA, which is obtained from the outgoing CYBER 170 exchange package. In either case, the CYBER 170 MF is toggled; and the outgoing CYBER 170 exchange package is stored beginning at the same CM address from where the incoming CYBER 170 exchange package is obtained. Also, the jump is always to relative address P, parcel 0, from the new CYBER 170 exchange package. Refer to CYBER 170 Exchange Jump in chapter 17.

Opcode **013xx**

Mnemonic XJ

Instruction Monitor exchange jump to MA (CYBER 170 MF clear)

Format

Remarks This instruction must start at parcel 0. Also, a CYBER 170 exchange package must be ready at address Bj plus K or at address MA.

This instruction stores P plus 1 into the outgoing CYBER 170 exchange package in hardware and then exchanges this CYBER 170 exchange package with the CYBER 170 exchange package stored in memory. If the CYBER 170 MF is set at the beginning of the instruction, the incoming CYBER 170 exchange package starts at absolute address Bj plus K. If the CYBER 170 MF is clear at the beginning, then the j and K fields of the instruction are ignored; and the incoming CYBER 170 exchange package starts at absolute address MA, which is obtained from the outgoing CYBER 170 exchange package. In either case, the CYBER 170 MF is toggled; and the outgoing CYBER 170 exchange package is stored beginning at the same CM address from where the incoming CYBER 170 exchange package is obtained. Also, the jump is always to relative address P, parcel 0, from the new CYBER 170 exchange package. Refer to CYBER 170 Exchange Jump in chapter 17.

CP Compare/Move Instructions

The compare/move instructions (table 16-10) move characters from one CM location to another and compare fields of characters either directly or through a collate table. The transmit instructions move words from one CM register to another.

Table 16-10. CP Compare/Move Instructions

Opcode	Format	Instruction	Mnemonic
10	ijx	Transmit (Xj) to Xi	BXi Xj
14	ixk	Transmit complement of (Xk) to Xi	BXi -Xk
464	jK	Move indirect	IM
465		Move direct	DM
466		Compare collated	CC
467		Compare uncollated	CU

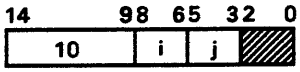
Transmit

Opcode **10ijx**

Mnemonic BXi Xj

Instruction Transmit (Xj) to Xi

Format



Remarks This instruction transfers a 60-bit word from Xj into Xi.

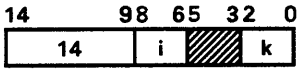
This instruction moves data from one X register to another X register. No logical function is performed on the data.

Opcode **14ixk**

Mnemonic BXi -Xk

Instruction Transmit complement of (Xk) to Xi

Format



Remarks This instruction reads a 60-bit word from Xk, complements the word, and writes the result into Xi.

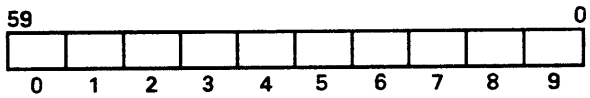
This instruction changes the sign of a fixed-point or floating-point quantity. The instruction also inverts an entire 60-bit field during data processing.

Compare/Move

The compare/move instructions (also referred to as CMU instructions) are provided for compatibility with previous systems. For better performance, recompile jobs to avoid use of CMU instructions.

CMU instructions must appear in parcel 0 or they are treated as illegal instructions.

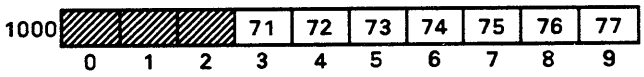
Data fields consisting of 6-bit characters may start or end with any character position (offset) of the ten 6-bit positions in each word. The character positions are designated as follows:



For move instructions, a K1 designator specifies which CM word contains the first character of the source data field, and a C1 designator specifies the character position (offset) of the first character. The K2 designator specifies the CM location in which the first character of the result data field is placed, and the C2 designator specifies the first character position. For compare instructions, both data field addresses specify source fields.

Example:

If the instruction is K1=1000 and C1=3, the first character of the source field is in position 3 of location 1000.



Therefore, the first character of the source field is 71.

An address is out of range if C1 or C2 is greater than 9, K1 plus N1 is greater than the program field length for CM (FLC), or K2 plus N2 is greater than FLC. N1 equals the number of CM references made to the source data field starting at K1, and N2 equals the number of CM references made to the result data field starting at K2. When an address-out-of-range condition occurs, the CMU instruction is not executed.

LL is the lower 4 bits, and LU is the upper 9 bits of the field length designator in numbers of characters. The maximum length of the data fields for the move direct and the compare instructions is 127 (177_8) characters. The maximum data field length for the move indirect instruction is 8191 (17777_8) characters. If L (LU and LL combined) is zero, the instruction becomes a pass.

For overlapping move instructions, the address of the source field (specified by K1) must be greater than the address of the result field (specified by K2) to provide proper field overlap. If K1 is less than K2, part of the source field is changed during execution. The amount of change is determined by the number of CM conflicts encountered. Overlapping fields should not contain more than 377 (octal) characters because an exchange jump interrupts any compare/ move operation having a decremented field length greater than 377 (octal).

To present the information in this chapter in a structured format, this page has been left blank.

Opcode **464jK**

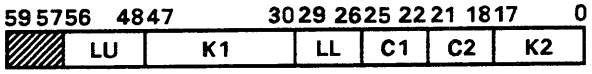
Mnemonic IM Bj + K

Instruction Move indirect

Format

Remarks Any instructions located in the lower two parcels of the instruction word do not execute.

Bj plus K specifies a relative address in CM for the following descriptor word.



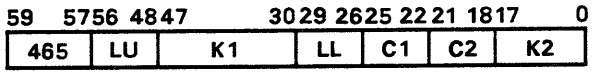
The descriptor word specifies the movement of the source field to the result field. The movement is from left to right through the field. Register X0 clears at the end of the execution.

Opcode **465**

Mnemonic DM

Instruction Move direct

Format



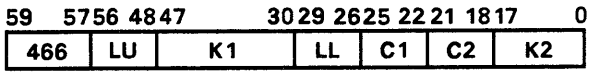
Remarks This instruction moves the source field to the result field as specified by the 60-bit instruction word. The field length is limited to a 7-bit count.

Opcode **466**

Mnemonic CC

Instruction Compare collated

Format



Remarks This instruction compares the field designated by K1,C1 with the field designated by K2,C2 as specified by the 60-bit instruction word.

The compare is from left to right through the fields until two unequal characters are found. These two characters are then collated and referenced in the collate table beginning at address A0 (table 16-11). If the table values found for the two unequal characters are equal, the compare continues until another pair of characters is unequal or until the field length is exhausted. If the table values found for the two unequal characters are unequal, X0 is set prior to instruction termination as follows:

- If field K1 is greater than field K2, set X0 to 0000 0000 0000 0000 0xxx.
- If field K1 is equal to field K2, set X0 to 0000 0000 0000 0000 0000.
- If field K1 is less than field K2, set X0 to 7777 7777 7777 7777 7yyy where yyy is the complement of xxx.

The value of the three octal numbers xxx that are stored in X0 is determined by the equation L minus N equals xxx (L is the length of the field, and N is the number of pairs of characters that were collated equal prior to instruction termination). In other words, xxx is the number of pairs of characters not yet compared plus one.

The A0 register contains the starting word address of an 8-word, 64-character collate table (table 16-11). This table must have been previously stored in consecutive CM locations.

The collated value of a character is found by examining the collate table. The upper 3 bits of the character to be collated are added to A0 to obtain the relative address of the word containing the collated value. The lower 3 bits of the character to be collated specify the character address of the collated value.

Example:

Suppose the character under examination is an octal 63. The 6 is added to the A0 to form the word address. The 3 is used to pick the correct character from that word. The value of 63 is 63 in the collate table.

Table 16-11. Collate Table

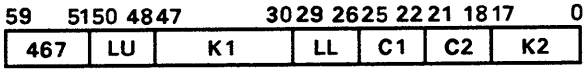
Address	Collating Character Locations							
A0	00	01	02	03	04	05	06	07
A0+1	10	11	12	13	14	15	16	17
A0+2	20	21	22	23	24	25	26	27
A0+3	30	31	32	33	34	35	36	37
A0+4	40	41	42	43	44	45	46	47
A0+5	50	51	52	53	54	55	56	57
A0+6	60	61	62	63	64	65	66	67
A0+7	70	71	72	73	74	75	76	77

Opcode **467**

Mnemonic CU

Instruction Compare uncollated

Format



Remarks This instruction is similar to the 466 instruction except that the collate table is not used. The X0 register is set when the first pair of unequal characters is encountered or when the field length is exhausted.

CP Set Instructions

Table 16-12 lists the CP set instructions. Opcodes 50 through 57 obtain operands from CM for computation and deliver the results back into CM. The remaining opcodes operate on B or X registers only.

Table 16-12. CP Set Instructions

Opcode	Format	Instruction	Mnemonic
50	ijK	Set Ai to (Aj) + K	SAi Aj+K
51	ijK	Set Ai to (Bj) + K	SAi Bj+K
52	ijK	Set Ai to (Xj) + K	SAi Xj+K
53	ijk	Set Ai to (Xj) + (Bk)	SAi Xj+Bk
54	ijk	Set Ai to (Aj) + (Bk)	SAi Aj+Bk
55	ijk	Set Ai to (Aj) - (Bk)	SAi Aj-Bk
56	ijk	Set Ai to (Bj) + (Bk)	SAi Bj+Bk
57	ijk	Set Ai to (Bj) - (Bk)	SAi Bj-Bk
60	ijK	Set Bi to (Aj) + K	SBi Aj+K
61	ijK	Set Bi to (Bj) + K	SBi Bj+K
62	ijK	Set Bi to (Xj) + K	SBi Xj+K
63	ijk	Set Bi to (Xj) + (Bk)	SBi Xj+Bk
64	ijk	Set Bi to (Aj) + (Bk)	SBi Aj+Bk
65	ijk	Set Bi to (Aj) - (Bk)	SBi Aj-Bk
66	ijk	Set Bi to (Bj) + (Bk)	SBi Bj+Bk
67	ijk	Set Bi to (Bj) - (Bk)	SBi Bj-Bk
70	ijK	Set Xi to (Aj) + K	SXi Aj+K
71	ijK	Set Xi to (Bj) + K	SXi Bj+K
72	ijK	Set Xi to (Xj) + K	SXi Xj+K
73	ijk	Set Xi to (Xj) + (Bk)	SXi Xj+Bk
74	ijk	Set Xi to (Aj) + (Bk)	SXi Aj+Bk
75	ijk	Set Xi to (Aj) - (Bk)	SXi Aj-Bk
76	ijk	Set Xi to (Bj) + (Bk)	SXi Bj+Bk
77	ijk	Set Xi to (Bj) - (Bk)	SXi Bj-Bk
660	jk	Read CM at (Xk) to Xj	CR Xj Xk
670	jk	Write Xj into CM at (Xk)	CW Xj Xk

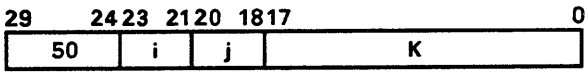
Set Ai

Opcode **50ijK**

Mnemonic SAi Aj + K

Instruction Set Ai to (Aj) + K

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. This instruction reads an operand from Aj, forms the sum of the operand plus K, and delivers the result to Ai. If the i designator is nonzero, a reference is made to CM, using the result as the relative address. The type of reference is a function of the i designator value.

i = 0	No CM reference
i = 1,2,3,4,5	Read from CM to Xi
i = 6,7	Write into CM from Xi

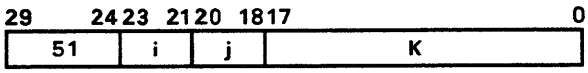
This instruction obtains operands from CM for computation and delivers the result back into CM.

Opcode **51ijK**

Mnemonic SAi Bj + K

Instruction Set Ai to (Bj) + K

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. This instruction reads an operand from Bj, forms the sum of the operand plus K, and delivers the result to Ai. If the i designator is nonzero, a reference is made to CM using the result as the relative address. The type of reference is a function of the i designator value.

i = 0	No CM reference
i = 1,2,3,4,5	Read from CM to Xi
i = 6,7	Write into CM from Xi

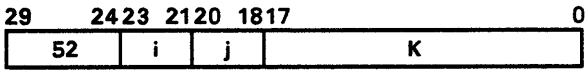
This instruction obtains operands from CM for computation and delivers the result back into CM.

Opcode **52ijK**

Mnemonic SAi Xj + K

Instruction Set Ai to (Xj) + K

Format



Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. This instruction reads an operand from Xj, forms the sum of the operand plus K, and delivers the result to Ai. If the i designator is nonzero, a reference is made to CM using the result as the relative address. The type of reference is a function of the i designator value.

i = 0	No CM reference
i = 1,2,3,4,5	Read from CM to Xi
i = 6,7	Write into CM from Xi

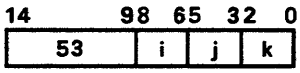
This instruction obtains operands from CM for computation and delivers the result back into CM.

Opcode **53ijk**

Mnemonic SAi Xj + Bk

Instruction Set Ai to (Xj) + (Bk)

Format



Remarks This instruction reads operands from Xj and Bk, forms the sum of the operands, and delivers the result to Ai. If the i designator is nonzero, a reference is made to CM using the result as the relative address. The type of reference is a function of the i designator value.

i = 0	No CM reference
i = 1,2,3,4,5	Read from CM to Xi
i = 6,7	Write into CM from Xi

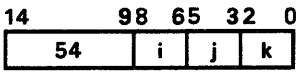
This instruction obtains operands from CM for computation and delivers the result back into CM.

Opcode **54ijk**

Mnemonic SAi Aj + Bk

Instruction Set Ai to (Aj) + (Bk)

Format



Remarks This instruction reads operands from Aj and Bk, forms the sum of the operands, and delivers the result to Ai. If the i designator is nonzero, a reference is made to CM using the result as the relative address. The type of reference is a function of the i designator value.

i = 0	No CM reference
i = 1,2,3,4,5	Read from CM to Xi
i = 6,7	Write into CM from Xi

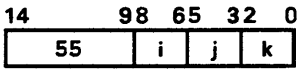
This instruction obtains operands from CM for computation and delivers the result back into CM.

Opcode **55ijk**

Mnemonic SAi Aj - Bk

Instruction Set Ai to (Aj) - (Bk)

Format



Remarks This instruction reads operands from Aj and Bk, subtracts the Bk operand from the Aj operand, and delivers the result to Ai. If the i designator is nonzero, a reference is made to CM using the result as the relative address. The type of reference is a function of the i designator value.

i = 0	No CM reference
i = 1,2,3,4,5	Read from CM to Xi
i = 6,7	Write into CM from Xi

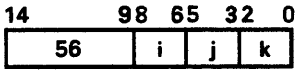
This instruction obtains operands from CM for computation and delivers the results back into CM.

Opcode **56ijk**

Mnemonic SAi Bj + Bk

Instruction Set Ai to (Bj) + (Bk)

Format



Remarks This instruction reads operands from Bj and Bk, forms the sum of the operands, and delivers the result to Ai. If the i designator is nonzero, a reference is made to CM using the result as the relative address. The type of reference is a function of the i designator value.

i = 0	No CM reference
i = 1,2,3,4,5	Read from CM to Xi
i = 6,7	Write into CM from Xi

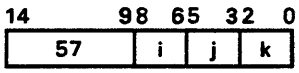
This instruction obtains operands from CM for computation and delivers the results back into CM.

Opcode **57ijk**

Mnemonic SAi Bj - Bk

Instruction Set Ai to (Bj) - (Bk)

Format



Remarks This instruction reads operands from Bj and Bk, subtracts the Bk operand from the Bj operand, and delivers the result to Ai. If the i designator is nonzero, a reference is made to CM using the result as the relative address. The type of reference is a function of the i designator value.

i = 0	No CM reference
i = 1,2,3,4,5	Read from CM to Xi
i = 6,7	Write into CM from Xi

This instruction obtains operands from CM for computation and delivers the result back into CM.

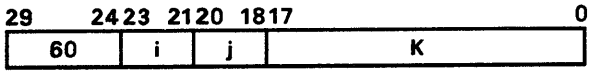
Set Bi

Opcode **60ijK**

Mnemonic SBi Aj + K

Instruction Set Bi to (Aj) + K

Format



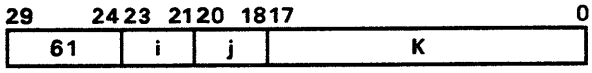
Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. This instruction reads an operand from Aj, forms the sum of the operand plus K, and delivers the result to Bi. The sum is formed in an 18-bit ones complement mode. This instruction is for address modification in the increment registers.

Opcode **61ijK**

Mnemonic SBi Bj + K

Instruction Set Bi to (Bj) + K

Format



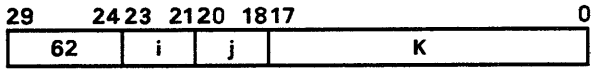
Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. This instruction reads an operand from Bj, forms the sum of the operand plus K, and delivers the result to Bi. The sum is formed in an 18-bit ones complement mode.

Opcode **62ijK**

Mnemonic SBi Xj + K

Instruction Set Bi to (Xj) + K

Format



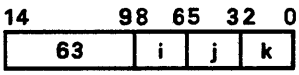
Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. This instruction reads an operand from Xj, forms the sum of the operand plus K, and delivers the result to Bi. The sum is formed in an 18-bit ones complement mode.

Opcode **63ijk**

Mnemonic SBi Xj + Bk

Instruction Set Bi to (Xj) + (Bk)

Format



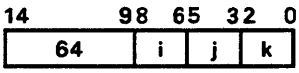
Remarks This instruction reads operands from Xj and Bk, adds the operands, and delivers the result to Bi. The sum is formed in an 18-bit ones complement mode.

Opcode **64ijk**

Mnemonic SBi Aj + Bk

Instruction Set Bi to (Aj) + (Bk)

Format



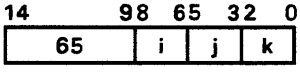
Remarks This instruction reads operands from Aj and Bk, adds the operands, and delivers the result to Bi. The sum is formed in an 18-bit ones complement mode.

Opcode **65ijk**

Mnemonic SBi Aj - Bk

Instruction Set Bi to (Aj) - (Bk)

Format



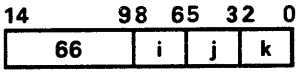
Remarks This instruction reads operands from Aj and Bk, subtracts the Bk operand from the Aj operand, and delivers the result to Bi. The difference is formed in an 18-bit ones complement mode. If the i designator is zero, this becomes a pass instruction.

Opcode **66ijk**

Mnemonic SBi Bj + Bk

Instruction Set Bi to (Bj) + (Bk)

Format



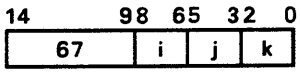
Remarks This instruction reads operands from Bj and Bk, adds the operands, and delivers the result to Bi. The sum is formed in an 18-bit ones complement mode. If the i designator is zero, this becomes a read central memory instruction.

Opcode **67ijk**

Mnemonic SBi Bj - Bk

Instruction Set Bi to (Bj) - (Bk)

Format



Remarks This instruction reads operands from Bj and Bk, subtracts the Bk operand from the Bj operand, and delivers the result to Bi. The difference is formed in an 18-bit ones complement mode. If the i designator is zero, this becomes a write central memory instruction.

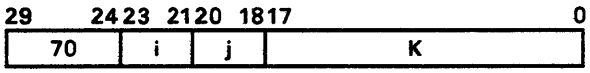
Set Xi

Opcode **70ijK**

Mnemonic SXi Aj + K

Instruction Set Xi to (Aj) + K

Format



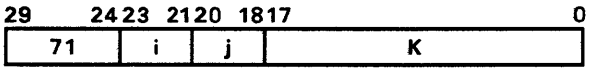
Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. This instruction reads an operand from Aj, forms the sum of the operand plus K, and delivers the result to Xi. The sum is formed in an 18-bit ones complement mode. The 18-bit result is sign-extended by copying the highest-order bit of the result into the upper 42 bit positions in Xi.

Opcode **71ijK**

Mnemonic SXi Bj + K

Instruction Set Xi to (Bj) + K

Format



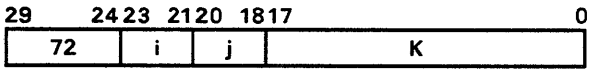
Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. This instruction reads an operand from Bj, forms the sum of the operand plus K, and delivers the result to Xi. The sum is formed in an 18-bit ones complement mode. The 18-bit result is sign-extended by copying the highest-order bit of the result into the upper 42 bit positions in Xi.

Opcode **72ijK**

Mnemonic SXi Xj + K

Instruction Set Xi to (Xj) + K

Format



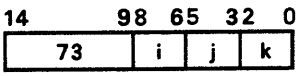
Remarks This two-parcel instruction uses the lower-order 18 bits as operand K. This instruction reads an operand from Xj, forms the sum of the operand plus K, and delivers the result to Xi. The sum is formed in an 18-bit ones complement mode. The 18-bit result is sign-extended by copying the highest-order bit of the result into the upper 42 bit positions in Xi.

Opcode **73ijk**

Mnemonic SXi Xj + Bk

Instruction Set Xi to (Xj) + (Bk)

Format



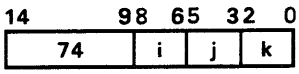
Remarks This instruction reads operands from Xj and Bk, adds the operands, and delivers the result to Xi. The sum is formed in an 18-bit ones complement mode. The 18-bit result is sign-extended by copying the highest-order bit of the result into the upper 42 bit positions in Xi.

Opcode **74ijk**

Mnemonic SXi Aj + Bk

Instruction Set Xi to (Aj) + (Bk)

Format



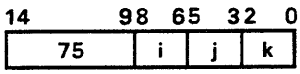
Remarks This instruction reads operands from Aj and Bk, adds the operands, and delivers the result to Xi. The sum is formed in an 18-bit ones complement mode. The 18-bit result is sign-extended by copying the highest-order bit of the result into the upper 42 bit positions in Xi.

Opcode **75ijk**

Mnemonic SXi Aj - Bk

Instruction Set Xi to (Aj) - (Bk)

Format



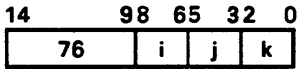
Remarks This instruction reads operands from Aj and Bk, subtracts the Bk operand from the Aj operand, and delivers the result to Xi. The difference is formed in an 18-bit ones complement mode. The 18-bit result is sign-extended by copying the highest-order bit of the result into the upper 42 bit positions in Xi.

Opcode **76ijk**

Mnemonic SXi Bj + Bk

Instruction Set Xi to (Bj) + (Bk)

Format



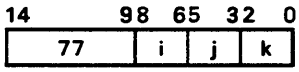
Remarks This instruction reads operands from Bj and Bk, adds the operands, and delivers the result to Xi. The sum is formed in an 18-bit ones complement mode. The 18-bit result is sign-extended by copying the highest-order bit of the result into the upper 42 bit positions in Xi.

Opcode **77ijk**

Mnemonic SXi Bj - Bk

Instruction Set Xi to (Bj) - (Bk)

Format



Remarks This instruction reads operands from Bj and Bk, subtracts the Bk operand from the Bj operand, and delivers the result to Xi. The difference is formed in an 18-bit ones complement mode. The 18-bit result is sign-extended by copying the highest-order bit of the result into the upper 42 bit positions in Xi.

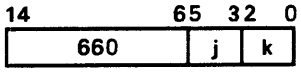
Read/Write

Opcode **660jk**

Mnemonic CR Xj, Xk

Instruction Read central memory at (Xk) to Xj

Format



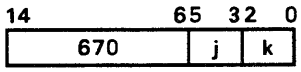
Remarks This instruction loads into Xj the word at location (Xk), where Xk is a right-justified 21-bit relative word address. Bits 21 through 59 of Xk are ignored. If the 21 bits of Xk are greater than or equal to FLC, an address-out-of-range condition is detected.

Opcode **670jk**

Mnemonic CW Xj, Xk

Instruction Write Xj into central memory at (Xk)

Format



Remarks This instruction stores Xj in location (Xk), where Xk is a 21-bit relative word address. Bits 21 through 59 of Xk are ignored. If the 21 bits of Xk are greater than or equal to FLC, an address-out-of-range condition is detected.

CP Normalize Instructions

The normalize instructions (table 16-13) perform normalizing operations in floating-point format and deliver the normalized result to Xi.

Table 16-13. CP Normalize Instructions

Opcode	Format	Instruction	Mnemonic
24	ijk	Normalize (Xk) to Xi and Bj	NXi Bj Xk
25	ijk	Round normalize (Xk) to Xi and Bj	ZXi Bj Xk

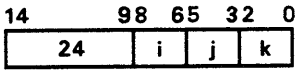
Normalize

Opcode **24ijk**

Mnemonic NXi Bj, Xk

Instruction Normalize (Xk) to Xi and Bj

Format



Remarks This instruction reads one operand from Xk, performs a normalizing operation on this word in floating-point format, and delivers the normalized result to Xi. In addition, a positive integer shift count is sent to Bj. This shift count is the number of bit positions of shift required to normalize the original operand coefficient.

The normalizing operation consists of repositioning the coefficient portion of the operand and then adjusting the exponent portion of the operand to leave the value of the result unaltered. The coefficient is shifted towards the higher-order bit positions of the word. The coefficient is shifted the minimum number of bit positions required to make bit 47 different from sign bit 59. This places the most-significant bit of the coefficient in the highest-order position. The exponent is then decreased by the number of bit positions shifted.

Two sample computations are listed in octal notation to illustrate the operation performed. The following example involves a positive floating-point number.

(Xk) = 2034 0047 6500 0000 2262

(Xi) = 2026 4765 0000 0022 6200

(Bj) = 00 0006

The following example involves a negative floating-point number.

(Xk) = 5743 7730 1277 7777 5515

(Xi) = 5751 3012 7777 7755 1577

(Bj) = 00 0006

Normalizing a number with either a positive or negative zero coefficient sets a shift count in Bj to 48 (decimal) and enters Xi with positive zero. If Xk contains an infinite quantity (3777xxx...x or 4000xxx...x) or an indefinite quantity (1777xxx...x or 6000xxx...x), no shift takes place. The content of Xk is copied to Xi, and Bj is set to zero. Corresponding infinite and indefinite exit conditions are also set in the CP for exit mode action. If the exponent is less than negative 1777 with a zero coefficient, the contents of Xi and Bj are set to zero. For further information, refer to Floating-Point Arithmetic under CP Programming in chapter 17.

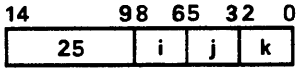
Round Normalize

Opcode **25ijk**

Mnemonic ZXi Bj, Xk

Instruction Round normalize (Xk) to Xi and Bj

Format



Remarks This instruction reads one operand from Xk, performs a rounding and then a normalizing operation in floating-point format, and delivers the round normalized result to Xi. In addition, a positive integer shift count is sent to Bj. This shift count is the number of bit positions of shift required to normalize the original operand coefficient.

The rounding operation consists of adding a bit to the coefficient portion of the operand in a bit position immediately below the least-significant bit position. This round bit has a value equal to the complement of the operand sign bit. The result increases the magnitude of the coefficient by one-half the value of the least-significant bit.

The normalizing operation consists of repositioning the coefficient and adjusting the exponent to leave the value of the resulting floating-point quantity unaltered. The coefficient is shifted towards the higher-order bit positions. The round bit is shifted along with the coefficient. The displacement is the minimum number of bit positions required to make bit 47 different from sign bit 59. This places the most-significant bit of the coefficient in the highest-order bit position. The exponent is decreased by the number of bit positions shifted.

Two sample computations are listed in octal notation to illustrate the normalizing operation performed. An example that involves a positive floating-point number is as follows.

(Xk) = 2034 0047 6500 0000 2262

(Xi) = 2026 4765 0000 0022 6420

(Bj) = 00 0006

The following example involves a negative number.

(Xk) = 5743 7730 1277 7777 5515

(Xi) = 5751 3012 7777 7755 1537

(Bj) = 00 0006

If Xk contains either an infinite quantity (3777xxx...x or 4000xxx...x) or an indefinite quantity (1777xxx...x or 6000xxx...x), no shift takes place. The content of Xk is copied to Xi, and Bj is set to zero. Corresponding infinite and indefinite exit conditions are also set in the CP for exit mode action.

Refer to Floating-Point Arithmetic under CP Programming in chapter 17.

CP Pass Instructions

The pass instructions (table 16-14) perform no operation and are used for filling words to get the next instruction properly positioned.

Table 16-14. CP Pass Instructions

Opcode	Format	Instruction	Mnemonic
460	xx	Pass	NO
461	xx	Pass	
462	xx	Pass	
463	xx	Pass	

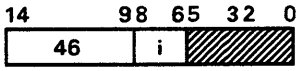
Pass

Opcode **460xx thru 463xx**

Mnemonic NO

Instruction Pass

Format



Remarks These instructions fill program instruction words where necessary to match jump destinations with word boundaries. The j and k designators are ignored, and a nonzero value has no effect in this instruction.

CP Illegal Instructions

The illegal instructions (table 16-15) cause an exchange to CYBER 170 monitor mode, when in CYBER 170 job mode, and cause a jump to executive state when in CYBER 170 monitor mode.

Table 16-15. CP Illegal Instructions

Opcode	Format	Instruction	Mnemonic
00	xxx	Error exit to MA or interrupt to executive mode	
017	jk	Illegal instruction (Trap 180)	RT
014	jk	Read one word from UEM to Xj	RXj Xk
015	jk	Write one word from Xj to UEM	WXj Xk

Error Exit

Opcode **00xxx**

Mnemonic PS

Purpose Error exit to MA when CYBER 17 MF clear Interrupt to executive mode when CYBER 170 MF set

Format

Remarks This instruction causes an illegal instruction error exit. CYBER 170 MF is the hardware monitor flag. Refer to Illegal Instructions in chapter 17.

Illegal Instruction

Opcode **017jk**

Instruction Illegal Instruction

Refer to Illegal Instructions in chapter 17.

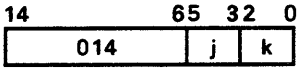
Illegal Read/Write

Opcode **014jk**

Mnemonic RXj Xk

Instruction Read one word from (Xk + RAE) to Xj

Format



Remarks This instruction is illegal if the UEM enable flag in the CYBER 170 exchange package is clear. This instruction reads the 60-bit word from UEM location Xk plus RAE into Xj. Xk is less than FLE.

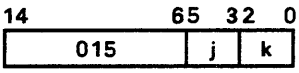
The number of bits checked for an address-out-of-range condition varies, depending on the addressing mode of UEM. In standard addressing mode, 24 bits of Xk are checked against 23 bits of FLE with bit 23 of FLE equal to zero. In expanded addressing mode, 30 bits of Xk are checked against 29 bits of FLE with bit 29 of FLE equal to zero. If Xk is greater than or equal to FLE, an address-out-of-range condition is detected.

Opcode **015jk**

Mnemonic WXj Xk

Instruction Write one word from Xj to (Xk + RAE)

Format



Remarks This instruction is illegal if the UEM enable flag in the CYBER 170 exchange package is clear. This instruction writes the 60-bit word from Xj into the UEM location Xk plus RAE. Xk is less than FLE.

The number of bits checked for an address-out-of-range condition varies, depending on the addressing mode of UEM. In standard addressing mode, 24 bits of Xk are checked against 23 bits of FLE with bit 23 of FLE equal to zero. In expanded addressing mode, 30 bits of Xk are checked against 29 bits of FLE with bit 29 of FLE equal to zero. If Xk is greater than or equal to FLE, an address-out-of-range condition is detected.

CP Mask Instruction

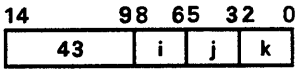
Form Mask

Opcode **43ijk**

Mnemonic MXi jk

Instruction Form mask of jk bits to Xi

Format



Remarks This instruction generates a masking word using the j and k designators as parameters. No operands are read from operating registers. The j and k designators are treated as a single, 6-bit octal quantity to designate the width of the masking field. A field of ones, beginning at the highest-order end of the word, is extended downward on a background of zeros. The completed masking word consists of one bits in the highest-order jk bit positions and zero bits in the remainder of the word. This masking word is then delivered to Xi. The following are sample parameters.

j = 2

k = 4

Xi = 7777 7760 0000 0000 0000

This instruction generates variable width masks for logical operations. This instruction, together with a shift instruction, generally creates an arbitrary field mask faster than reading a pregenerated mask from CM.

CP Pop Count Instruction

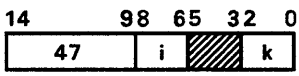
Population Count

Opcode **47ixk**

Mnemonic CXi Xk

Instruction Population count of (Xk) to Xi

Format



Remarks This instruction reads one operand from Xk, counts the number of one bits in the operand, and stores the count in Xi. The count delivered to Xi is a positive integer. If the operand is all ones, a count of 60 (decimal) is delivered to Xi. If operand is all zeros, a zeros word is delivered to Xi.

CP Read Free-Running Counter Instruction

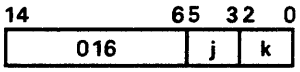
Read Free-Running Counter

Opcode **016jk**

Mnemonic RC Xj

Instruction Read free-running counter

Format



Remarks This instruction transfers the current contents of the 48-bit free-running counter to the Xj register. The leftmost 12 bits of Xj are set to zero. The k field is ignored.

This instruction is a single parcel instruction that can be located in any parcel.

PP Instruction Descriptions

The peripheral processor (PP) instruction set comprises the following eight subgroups.

- Load/Store
- Arithmetic
- Logical
- Replace
- Branch
- Central Memory Access
- Input/Output
- Other

PP Instruction Formats

Figure 16-2 shows PP instruction formats. PP instructions are 16 or 32 bits long. In instruction descriptions, the operation code is given either by two or three octal digits. The third digit, when used, indicates the state of the s-bit (0 or 1) in I/O instructions (refer to table 16-16).

The upper 4 bits of the PP instructions must be zero to ensure that the instructions operate as defined in this chapter.

Table 16-16. PP Nomenclature

Designator	Description
Opcode	Specifies instruction operation code.
s	Specifies I/O instruction subcode.
c	Specifies channel number.
A	Refers to the A register (arithmetic register) or the content of the A register.
(A)	Refers to the content of the word at the CM address specified by the A register.
P	Refers to the P register or to the content of the P register (program address register).
R	Refers to the R register or to the content of the R register (relocation register).
(d)	Refers to the content of the word at the PP memory address specified by the d field (direct mode).
((d))	Refers to the content of the word at the PP memory address specified by the content of the word at the PP memory address specified by the d field (indirect mode).
m + (d)	Refers to the PP memory address specified by the m field indexed by the content of the word at the PP memory address specified by the d field.
(m + (d))	Refers to the content of the word at the PP memory address specified by the m field indexed by the content of the word at the PP memory address specified by the d field (memory mode).

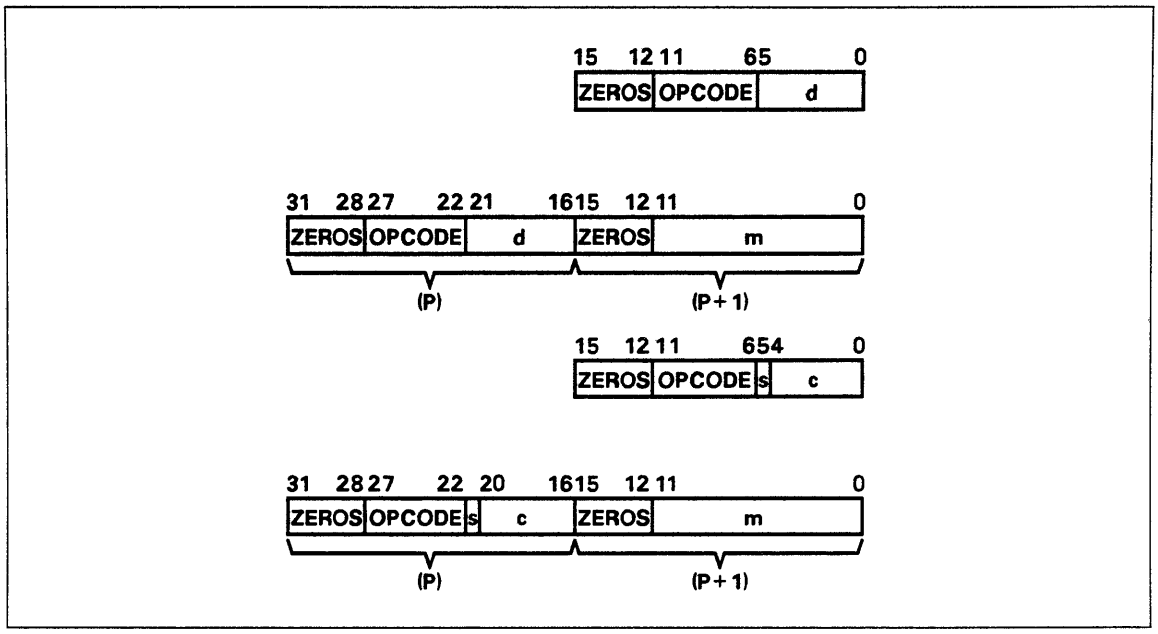


Figure 16-2. PP Instruction Formats

PP Data Format

Figure 16-3 shows PP data format and how 12-bit data is packed into 64-bit CM words or unpacked from 64-bit CM words.

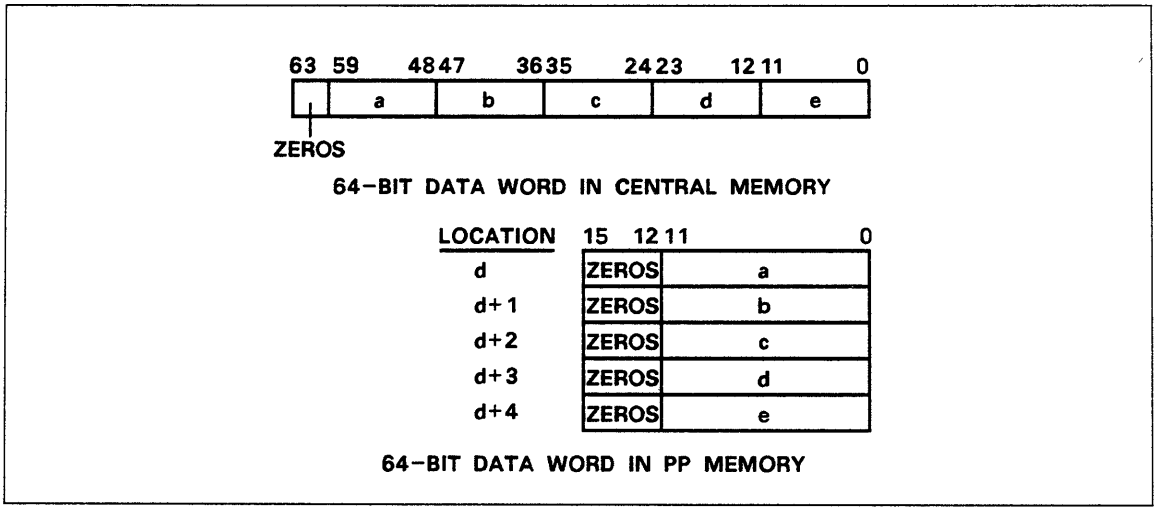


Figure 16-3. PP Data Format

PP Relocation Register Format

Figure 16-4 shows PP relocation (R) register format. This register is loaded-from/stored-into PP memory by instructions 24 and 25 (load/store R register).

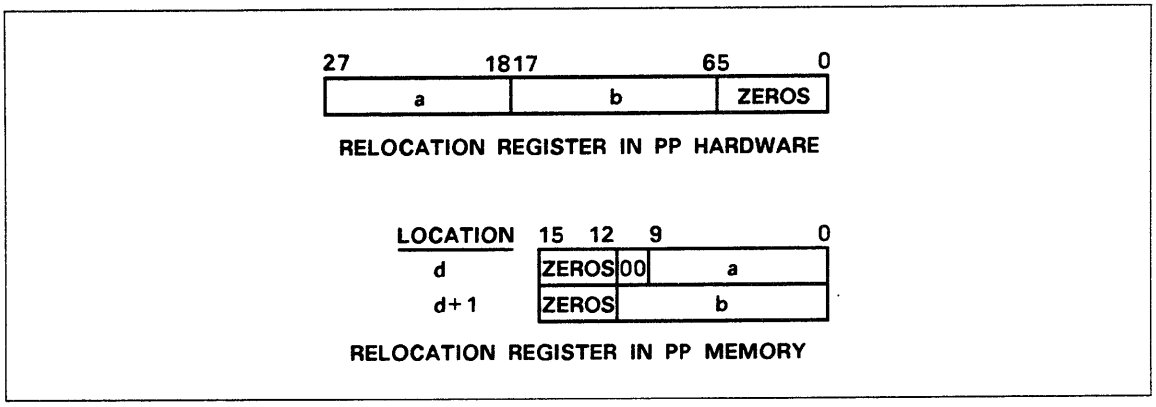


Figure 16-4. PP Relocation (R) Register Format

PP Load/Store Instructions

Load and store instructions (table 16-17) transfer 6-, 10-, 12-, and 18-bit quantities between the PP A register and the PP memory.

Table 16-17. PP Load/Store Instructions

Opcode	Format	Instruction	Mnemonic
14	d	Load d	LDN d
15	d	Load complement d	LCN d
20	dm	Load dm	LDC m,d
24	d	Load R	LRD d
30	d	Load (d)	LDD d
40	d	Load ((d))	LDI d
50	dm	Load (m+(d))	LDM m,d
25	d	Store R	SRD d
34	d	Store (d)	STD d
44	d	Store ((d))	STI d
54	dm	Store (m+(d))	STM m,d

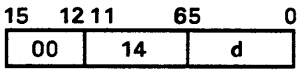
Load

Opcode **14d**

Mnemonic LDN d

Instruction Load d

Format



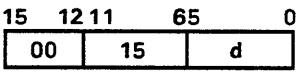
Remarks This instruction clears the A register and loads d. The upper 12 bits of A are zero.

Opcode **15d**

Mnemonic LCN d

Instruction Load complement d

Format



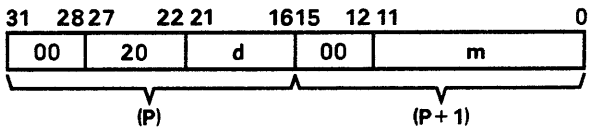
Remarks This instruction clears the A register and loads the complement of d. The upper 12 bits of A are one.

Opcode **20dm**

Mnemonic LDC dm

Instruction Load dm

Format



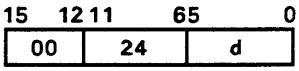
Remarks This instruction clears the A register and loads an 18-bit quantity consisting of d as the upper 6 bits and m as the lower 12 bits. The content of the location (P plus 1) which follows the present program address (P) is read to provide m.

Opcode **24d**

Mnemonic LRD d

Instruction Load R register

Format



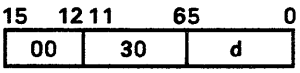
Remarks Figure 16-4 shows R register format. If d is not equal to zero, this instruction loads the upper 10 bits of the R register (bits 18 through 27) from the rightmost 10 bits of PP memory location d. The 12 bits contained in PP memory location d plus 1 are loaded into the next 12 bits of the R register (bits 6 through 17). If d equals zero, the instruction is a pass.

Opcode **30d**

Mnemonic LDD d

Instruction Load (d)

Format



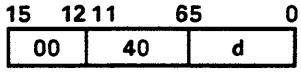
Remarks This instruction clears the A register and loads the content at location d. The upper 6 bits of A are zero.

Opcode **40d**

Mnemonic LDI d

Instruction Load ((d))

Format



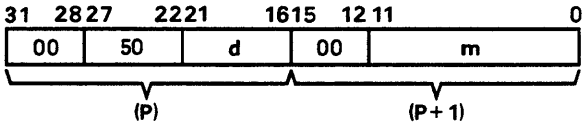
Remarks This instruction clears the A register and loads a 12-bit quantity that is obtained by indirect addressing. The upper 6 bits of A are zero. Location d is read from PPM, and the word read is used as the operand address.

Opcode **50dm**

Mnemonic LDM m,d

Instruction Load (m + (d))

Format



Remarks This instruction clears the A register and loads a 12-bit quantity. The upper 6 bits of A are zeros. The 12-bit operand is obtained by indexed direct addressing.

In indexed direct addressing, the quantity m, which is read from PPM location P plus 1, serves as the base operand address to which the content of d is added. If d equals zero, the operand address is m; but if d is not equal to zero, m plus the content in d is the operand address. Therefore, location d may be used as an index quantity to modify operand addresses.

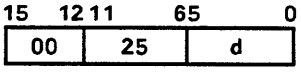
Store

Opcode **25d**

Mnemonic SRD d

Instruction Store R register

Format



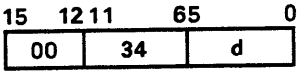
Remarks Figure 16-4 shows R register format. If d is not equal to zero, this instruction stores the upper 10 bits of the R register (bits 18 through 27) into the rightmost 10 bits of PP memory location d. The 12 bits contained in PP memory location d plus 1 are stored into the next 12 bits of the R register (bits 6 through 17). If d equals zero, the instruction is a pass.

Opcode **34d**

Mnemonic STD d

Instruction Store (d)

Format



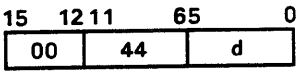
Remarks This instruction stores the lower 12 bits of the A register at location d.

Opcode **44d**

Mnemonic STI d

Instruction Store ((d))

Format



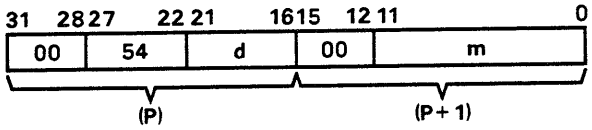
Remarks This instruction stores the lower 12 bits of the A register at the location specified by the content of location d.

Opcode **54dm**

Mnemonic STM m,d

Instruction Store (m + (d))

Format



Remarks This instruction stores the lower 12 bits of the A register in the location determined by indexed direct addressing.

In indexed direct addressing, the quantity m, which is read from PPM location P plus 1, serves as the base operand address to which the content of d is added. If d equals zero, the operand address is m; but if d is not equal to zero, m plus the content in d is the operand address. Therefore, location d may be used as an index quantity to modify operand addresses.

PP Arithmetic Instructions

The PP arithmetic instructions (table 16-18) perform integer arithmetic using the PP A register contents as one operand, with the other operand specified by the instruction. The result replaces the original contents of A. The PP considers the operands as ones complement integers and performs the arithmetic in ones complement.

Table 16-18. PP Arithmetic Instructions

Opcode	Format	Instruction	Mnemonic
16	d	Add d	ADN d
21	dm	Add dm	ADC m,d
31	d	Add (d)	ADD d
41	d	Add ((d))	ADI d
51	dm	Add (m+(d))	ADM m,d
17	d	Subtract d	SBN d
32	d	Subtract (d)	SBD d
42	d	Subtract ((d))	SBI d
52	dm	Subtract (m+(d))	SBM m,d

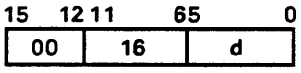
Arithmetic Add

Opcode **16d**

Mnemonic ADN d

Instruction Add d

Format



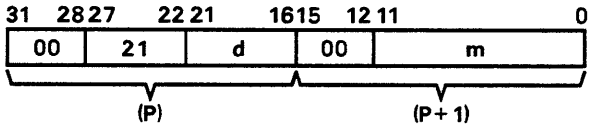
Remarks This instruction adds d (treated as a 6-bit positive quantity) to the content of the A register.

Opcode **21dm**

Mnemonic ADC dm

Instruction Add dm

Format



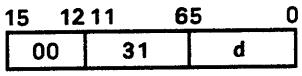
Remarks This instruction adds to the A register the 18-bit quantity consisting of d as the upper 6 bits and m as the lower 12 bits. The content of the location (P plus 1) which follows the present program address (P) is read to provide m.

Opcode **31d**

Mnemonic ADD d

Instruction Add (d)

Format



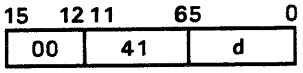
Remarks This instruction adds the content at location d (treated as a 12-bit positive quantity) to the A register.

Opcode **41d**

Mnemonic ADI d

Instruction Add ((d))

Format



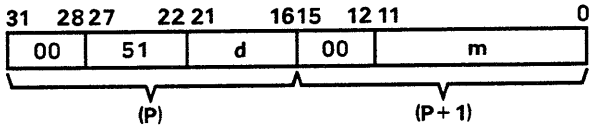
Remarks This instruction adds to the content of the A register a 12-bit operand (treated as a positive quantity) obtained by indirect addressing. Location d is read from PPM, and the word read is used as the operand address.

Opcode **51dm**

Mnemonic ADM m,d

Instruction Add (m + (d))

Format



Remarks This instruction adds the 12-bit operand (treated as a positive quantity) read by indexed direct addressing to the A register.

In indexed direct addressing, the quantity m, which is read from PPM location P plus 1, serves as the base operand address to which the content of d is added. If d equals zero, the operand address is m, but if d is not equal to zero, m plus the content in d is the operand address. Therefore, location d may be used as an index quantity to modify operand addresses.

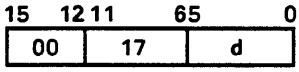
Arithmetic Subtract

Opcode **17d**

Mnemonic SBN d

Instruction Subtract d

Format



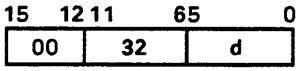
Remarks This instruction subtracts d (treated as a 6-bit positive quantity) from the content of the A register.

Opcode **32d**

Mnemonic SBD d

Instruction Subtract (d)

Format



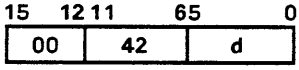
Remarks This instruction subtracts the content at location d (treated as a 12-bit positive quantity) from the A register.

Opcode **42d**

Mnemonic SBI d

Instruction Subtract ((d))

Format



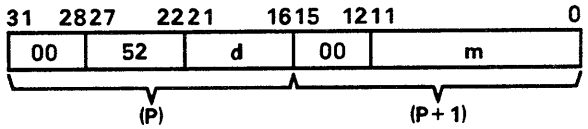
Remarks This instruction subtracts from the A register a 12-bit operand (treated as a positive quantity) obtained by indirect addressing. Location d is read from PPM, and the word read is used as the operand address.

Opcode **52dm**

Mnemonic SBM m,d

Instruction Subtract (m + (d))

Format



Remarks This instruction subtracts the 12-bit operand (treated as a positive quantity) read by indexed direct addressing from the A register.

In indexed direct addressing, the quantity m, which is read from PPM location P plus 1, serves as the base operand address to which the content of d is added. If d equals zero, the operand address is m; but if d is not equal to zero, m plus the content in d is the operand address. Therefore, location d may be used as an index quantity to modify operand addresses.

PP Logical Instructions

The logical instructions (table 16-19) perform operations with one operand as the PP A register contents, and the other as specified by the instruction. The result replaces the original contents of A.

Table 16-19. PP Logical Instructions

Opcode	Format	Instruction	Mnemonic
10	d	Shift d	SHN d
13	d	Selective clear d	SCN d
11	d	Logical difference d	LMN d
23	dm	Logical difference dm	LMC m,d
33	d	Logical difference (d)	LMD d
43	d	Logical difference ((d))	LMI d
53	dm	Logical difference (m+(d))	LMM m,d
12	d	Logical product d	LPN d
22	dm	Logical product dm	LPC m,d

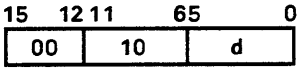
Shift

Opcode **10d**

Mnemonic SHN d

Instruction Shift d

Format



Remarks This instruction shifts the content of the A register right or left d places. If d is positive (00 through 37), the shift is left circular. If d is negative (40 through 77), the shift is right circular (end-off with no sign extension). Thus, d equal to 06 requires a left-shift of six places; d equal to 71 requires a right-shift of six places.

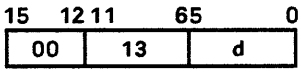
Selective Clear

Opcode **13d**

Mnemonic SCN d

Instruction Selective clear d

Format



Remarks This instruction clears any of the lower 6 bits of the A register where corresponding bits of d are one. The upper 12 bits of A are not altered.

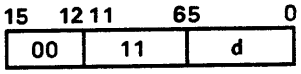
Logical Difference

Opcode **11d**

Mnemonic LMN d

Instruction Logical difference d

Format



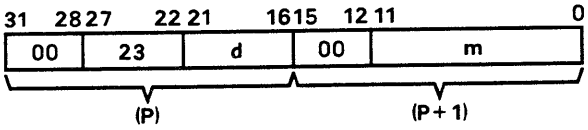
Remarks This instruction forms the bit-by-bit logical difference of d and the lower 6 bits of A in the register in A. This is equivalent to complementing individual bits of A that correspond to bits of d that are one. The upper 12 bits of A are not altered.

Opcode **23dm**

Mnemonic LMC dm

Instruction Logical difference dm

Format



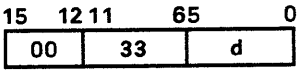
Remarks This instruction forms the bit-by-bit logical difference of the content of the A register and the 18-bit quantity dm in A. This is equivalent to complementing individual bits of A which correspond to bits of dm that are one. The upper 6 bits of the quantity consist of d, and the lower 12 bits are the content of the location (P plus 1), which follows the present program address (P).

Opcode **33d**

Mnemonic LMD d

Instruction Logical difference (d)

Format



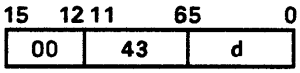
Remarks This instruction forms in the A register the bit-by-bit logical difference of the lower 12 bits of the A register and the content at location d. This is equivalent to complementing individual bits of A that correspond to bits in location d that are ones. The upper 6 bits are not altered.

Opcode **43d**

Mnemonic LMI d

Instruction Logical difference ((d))

Format



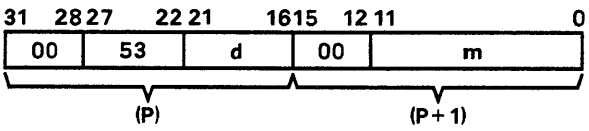
Remarks This instruction forms in the A register the bit-by-bit logical difference of the lower 12 bits of the A register and the 12-bit operand read by indirect addressing. Location d is read from PPM, and the word read is used as the operand address. The upper 6 bits of A are not altered.

Opcode **53dm**

Mnemonic LMM m,d

Instruction Logical difference (m + (d))

Format



Remarks This instruction forms the bit-by-bit logical difference of the lower 12 bits of the A register and a 12-bit operand obtained by indexed direct addressing in the A register. The upper 6 bits of A are not altered.

In indexed direct addressing, the quantity m, which is read from PPM location P plus 1, serves as the base operand address to which the content of d is added. If d equals zero, the operand address is m; but if d is not equal to zero, m plus the content in d is the operand address. Therefore, location d may be used as an index quantity to modify operand addresses.

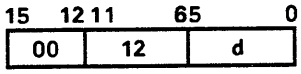
Logical Product

Opcode **12d**

Mnemonic LPN d

Instruction Logical product d

Format



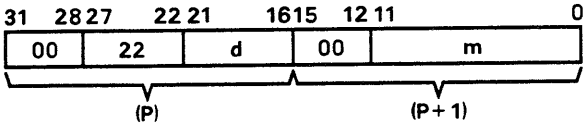
Remarks This instruction forms the bit-by-bit logical product of d and the lower 6 bits of the A register and leaves this quantity in the lower 6 bits of A. The upper 12 bits of A are zero.

Opcode **22dm**

Mnemonic LPC dm

Instruction Logical product dm

Format



Remarks This instruction forms the bit-by-bit logical product of the content of the A register and the 18-bit quantity dm in A. The upper 6 bits of this quantity consist of d, and the lower 12 bits are the content of the location (P plus 1), which follows the present program address (P).

PP Replace Instructions

The replace instructions (table 16-20) perform integer arithmetic with one operand as the contents of A and the other as specified by the instruction. The result replaces the original contents of A and the contents of the other operands location. The result stored in location d is either the rightmost 12 bits (for the normal instructions) or the rightmost 16 bits (for the long instructions) of the A register. Therefore, since A contains 18 bits, the value remaining in A cannot equal the value stored in PP memory location d. The PP considers the operands as ones complement integers and performs ones complement arithmetic.

Table 16-20. PP Replace Instructions

Opcode	Format	Instruction	Mnemonic
35	d	Replace add (d)	RAD d
36	d	Replace add 1 (d)	AOD d
45	d	Replace add ((d))	RAI d
46	d	Replace add 1 ((d))	AOI d
55	dm	Replace add (m+(d))	RAM m,d
56	dm	Replace add 1 (m+(d))	AOM m,d
37	d	Replace subtract 1 (d)	SOD d
47	d	Replace subtract 1 ((d))	SOI d
57	dm	Replace subtract 1 (m+(d))	SOM m,d

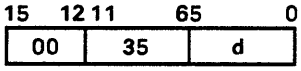
Replace Add

Opcode **35d**

Mnemonic RAD d

Instruction Replace add (d)

Format



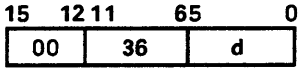
Remarks This instruction adds the quantity at location d to the content of the A register and stores the lower 12 bits of the result at location d. The result remains in A at the end of the operation, and the original content of A is destroyed.

Opcode **36d**

Mnemonic AOD d

Instruction Replace add 1 (d)

Format



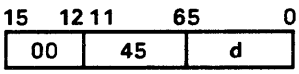
Remarks This instruction replaces the quantity at location d with its original value plus 1. The result remains in the A register at the end of the operation, and the original content of A is destroyed.

Opcode **45d**

Mnemonic RAI d

Instruction Replace add ((d))

Format



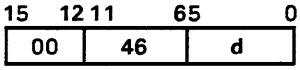
Remarks This instruction adds the operand, which is obtained from the location specified by the content at location d, to the content of the A register. The lower 12 bits of the sum replace the original operand. The result remains in A at the end of the operation.

Opcode **46d**

Mnemonic AOI d

Instruction Replace add 1 ((d))

Format



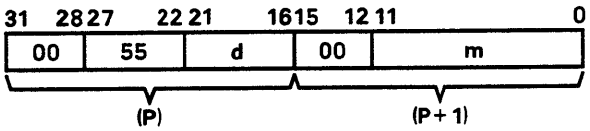
Remarks This instruction replaces the operand, which is obtained from the location specified by the content at location d, by its original value plus one. The result remains in the A register at the end of the operation, and the original content of A is destroyed.

Opcode **55dm**

Mnemonic RAM m,d

Instruction Replace add (m + (d))

Format



Remarks This instruction adds the operand, which is obtained from the location determined by indexed direct addressing, to the A register. The lower 12 bits of the sum replace the original operand in PPM. The result remains in A at the end of the operation, and the original content of A is destroyed.

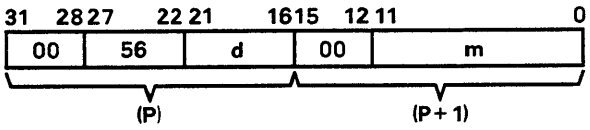
In indexed direct addressing, the quantity m, which is read from PPM location P plus 1, serves as the base operand address to which the content of d is added. If d equals zero, the operand address is m; but if d is not equal to zero, m plus the content in d is the operand address. Therefore, location d may be used as an index quantity to modify operand addresses.

Opcode **56dm**

Mnemonic AOM m,d

Instruction Replace add 1 (m + (d))

Format



Remarks This instruction replaces the operand, which is obtained from the location determined by indexed direct addressing, by its original value plus one. The result remains in the A register at the end of the operation, and the original content of A is destroyed.

In indexed direct addressing, the quantity m, which is read from PPM location P plus 1, serves as the base operand address to which the content of d is added. If d equals zero, the operand address is m; but if d is not equal to zero, m plus the content in d is the operand address. Therefore, location d may be used as an index quantity to modify operand addresses.

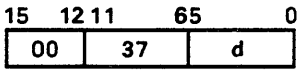
Replace Subtract

Opcode **37d**

Mnemonic SOD d

Instruction Replace subtract 1 (d)

Format



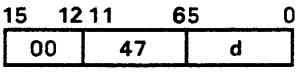
Remarks This instruction replaces the quantity at location d with its original value minus one. The result remains in the A register at the end of the operation, and the original content of A is destroyed.

Opcode **47d**

Mnemonic SOI d

Instruction Replace subtract 1 ((d))

Format



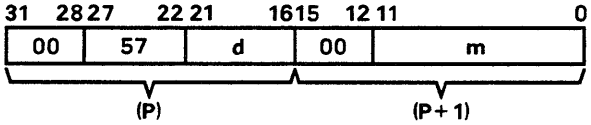
Remarks This instruction replaces the operand, which is obtained from the location specified by the content at location d, by its original value minus one. The result remains in the A register at the end of the operation, and the original content of A is destroyed.

Opcode **57dm**

Mnemonic SOM m,d

Instruction Replace subtract 1 (m + (d))

Format



Remarks This instruction replaces the operand, which is obtained from the location determined by indexed direct addressing, by its original value minus one. The result remains in the A register at the end of the operation, and the original content of A is destroyed.

In indexed direct addressing, the quantity m, which is read from PPM location P plus 1, serves as the base operand address to which the content of d is added. If d equals zero, the operand address is m; but if d is not equal to zero, m plus the content in d is the operand address. Therefore, location d may be used as an index quantity to modify operand addresses.

PP Branch Instructions

The branch instructions (table 16-21) allow departure from sequential instruction execution.

Table 16-21. PP Branch Instructions

Opcode	Format	Instruction	Mnemonic
01	dm	Long jump to m + (d)	LJM m,d
02	dm	Return jump to m + (d)	RJM m,d
03	d	Unconditional jump d	UJN d
04	d	Zero jump d	ZJN d
05	d	Nonzero jump d	NJN d
06	d	Plus jump d	PJN d
07	d	Minus jump d	MJN d
640	cm	Jump to m if channel c active	AJM m,c
650	cm	Jump to m if channel c inactive	IJM m,c
660	cm	Jump to m if channel c full	FJM m,c
661	cm	Jump to m if channel c error flag set	SFM m,40B+c
670	cm	Jump to m if channel c empty	EJM m,c
671	cm	Jump to m if channel c error flag clear	CFM m,40B+c

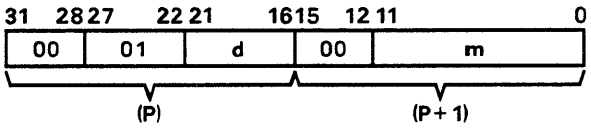
Long Jump

Opcode **01dm**

Mnemonic LJM m,d

Instruction Long jump to m + (d)

Format



Remarks This instruction jumps to the address given by m plus the content of location d. If d equals zero, m is not modified.

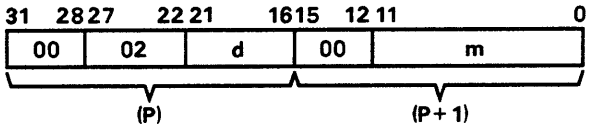
Return Jump

Opcode **02dm**

Mnemonic RJM m,d

Instruction Return jump to m + (d)

Format



Remarks This instruction jumps to the address given by m plus the content of location d. If d equals zero, m is not modified. The current program address (P) plus 2 is stored at the jump address. The next instruction starts at the jump address plus 1. The subprogram exits with a long jump or normal sequencing to the jump address minus 1, which in turn contains a long jump, 0100. This returns the original program address plus 2 to the P register.

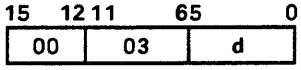
Unconditional Jump

Opcode **03d**

Mnemonic UJN d

Instruction Unconditional jump d

Format



Remarks This instruction provides an unconditional jump to any address up to 31 (decimal) locations forward or backward from the current program address. The value of d is added to the current program address. If d is positive (01 through 37), 0001 through 0037 is added, and the jump is forward. If d is negative (40 through 76), 7740 through 7776 is added, and the jump is backward. When d equals 00 or 77, the PP hangs. A deadstart is required to restart the PP.

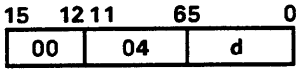
Zero/Nonzero Jump

Opcode **04d**

Mnemonic ZJN d

Instruction Zero jump d

Format



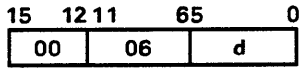
Remarks This instruction provides a conditional jump to any address up to 31 (decimal) locations forward or backward from the current program address. If the content of the A register is zero, the jump is taken. If the content of A is nonzero, the next instruction executes from P plus 1. A negative zero (777777) is treated as nonzero. If d is positive (01 through 37), 0001 through 0037 is added, and the jump is forward. If d is negative (40 through 76), 7740 through 7776 is added, and the jump is backward. When d equals 00 or 77, the PP hangs. A deadstart is required to restart the PP.

Opcode **05d**

Mnemonic NJN d

Instruction Nonzero jump d

Format



Remarks This instruction provides a conditional jump to any address up to 31 (decimal) locations forward or backward from the current program address. If the content of the A register is nonzero, the jump is taken. If the content of A is zero, the next instruction executes from P plus 1. A negative zero (777777) is treated as nonzero. If d is positive (01 through 37), 0001 through 0037 is added, and the jump is forward. If d is negative (40 through 76), 7740 through 7776 is added, and the jump is backward. When d equals 00 or 77, the PP hangs. A deadstart is required to restart the PP.

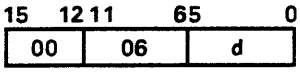
Plus/Minus Jump

Opcode **06d**

Mnemonic PJN d

Instruction Plus jump d

Format



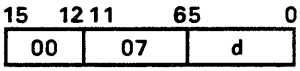
Remarks This instruction provides a conditional jump to any address up to 31 (decimal) locations forward or backward from the current program address. If the sign of the A register is positive, the jump is taken. If the sign of A is negative, the next instruction executes from P plus 1. A positive zero is treated as a positive quantity. A negative zero is treated as a negative quantity. If d is positive (01 through 37), 0001 through 0037 is added, and the jump is forward. If d is negative (40 through 76), 7740 through 7776 is added, and the jump is backward. When d equals 00 or 77, the PP hangs. A deadstart is required to restart the PP.

Opcode **07d**

Mnemonic MJN d

Instruction Minus jump d

Format



Remarks This instruction provides a conditional jump to any address up to 31 (decimal) locations forward or backward from the current program address. If the content of the A register is negative, the jump is taken. If the content of A is positive, the next instruction executes from P plus 1. A positive zero is treated as a positive quantity. A negative zero is treated as a negative quantity. If d is positive (01 through 37), 0001 through 0037 is added, and the jump is forward. If d is negative (40 through 76), 7740 through 7776 is added, and the jump is backward. When d equals 00 or 77, the PP hangs. A deadstart is required to restart the PP.

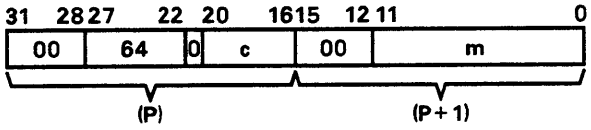
Jump To m

Opcode **640cm**

Mnemonic AJM m,c

Instruction Jump to m if channel c active

Format



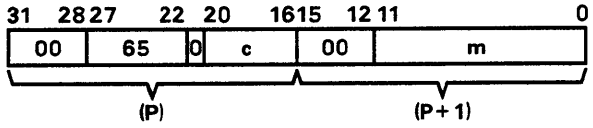
Remarks If channel c is active, this instruction causes a jump to m; otherwise, it is a pass.

Opcode **650cm**

Mnemonic IJM m,c

Instruction Jump to m if channel c inactive

Format



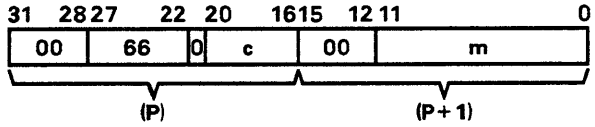
Remarks This instruction provides a conditional jump to a new address specified by m. The jump is taken if the channel specified by c is inactive. The next instruction is at P plus 2 if the channel is active.

Opcode **660cm**

Mnemonic FJM m,c

Instruction Jump to m if channel c full

Format



Remarks This instruction provides a conditional jump to a new address specified by m. The jump is taken if the channel designated by c is full. The next instruction is at P plus 2 if the channel is empty.

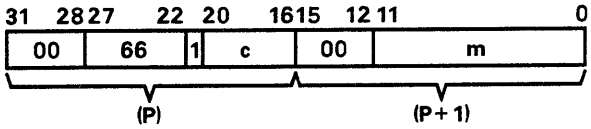
An input channel is full when the input equipment places a word in the channel and no PP has accepted that word. The channel is empty when a word has been accepted. An output channel is full when a PP places a word on the channel. The channel is empty when the output equipment accepts the word.

Opcode **661cm**

Mnemonic SFM m,c

Instruction Jump to m if channel c error flag set

Format



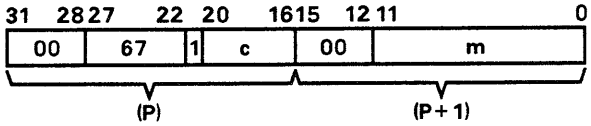
Remarks If the channel c error flag is set, this instruction clears the error flag and causes a jump to m. If this error flag is clear, the instruction is a pass. When m is set to P plus 2, the channel error flag is unconditionally cleared when the program reaches P plus 2.

Opcode **670cm**

Mnemonic EJM m,c

Instruction Jump to m if channel c empty

Format



Remarks This instruction provides a conditional jump to a new address specified by m. The jump is taken if the channel specified by c is empty. The next instruction is at P plus 2 if the channel is full.

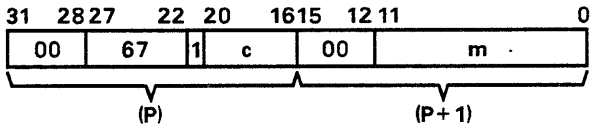
An input channel is full when the input equipment places a word in the channel and no PP has accepted that word. The channel is empty when a word has been accepted. An output channel is full when a PP places a word on the channel. The channel is empty when the output equipment accepts the word.

Opcode **671cm**

Mnemonic CFM m,c

Instruction Jump to m if channel c error flag clear

Format



Remarks If the channel c error flag is clear, this instruction causes a jump to m. If this error flag is set, the instruction clears the error flag and proceeds with the next instruction. When m is set to P plus 2, the channel error flag is unconditionally cleared when the program reaches P plus 2.

PP Central Memory Access Instructions

The PP central memory access instructions (table 16-22) provide the capability to read and write CM words to and from PP memory. The PPs have read access to all CM storage locations, while the OS bounds register controls write and exchange accesses. The IOU performs CM addressing with real memory word addresses. To address all locations in the larger CM sizes available, the IOU uses address relocation to modify the CM address in the A register of the PP. If bit 46 in A is 1 during a PP central memory read or write instruction, the IOU adds the R register contents to A register bits 47 through 63 to produce the CM address. If bit 46 of A is 0, the IOU does not perform address relocation but uses the A address. The R register contains an absolute 64-word starting boundary within CM. When relocation is desired, an absolute CM address is formed by concatenating six 0's to the rightmost end of the R contents and adding bits 47 through 63 of A.

Table 16-22. PP Central Memory Access Instructions

Opcode	Format	Instruction	Mnemonic
60	d	Central read from (A) to d	CRD d
61	dm	Central read (d) words from (A) to m	CRM m,d
62	d	Central write to (A) from d	CWD d
63	dm	Central write (d) words to (A) from m	CWM m,d

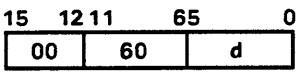
Central Read

Opcode **60d**

Mnemonic CRD d

Instruction Central read from (A) to d

Format



Remarks This instruction disassembles one 60-bit word from central memory into five 12-bit words and stores these in five consecutive PP memory locations, beginning with the leftmost 12 bits of the 60-bit word.

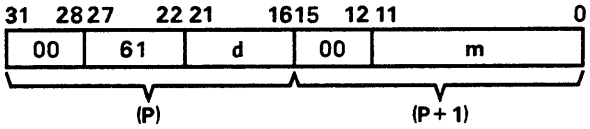
The parameters of the transfer are as follows: If bit 17 of A is zero, A bits 0 through 16 contain the absolute address of the 60-bit word transferred. If bit 17 of A is one, hardware adds relocation register R to zero-extended A bits 0 through 16 to obtain the absolute address of the 60-bit word transferred. For further information, refer to R Register under Input/Output Unit in the chapter entitled Functional Descriptions for the applicable computer system, and PP Relocation Register Format at the beginning of this section under PP Instruction Descriptions. Field d gives the PP location that receives the first 12-bit word transferred. PP memory addressing is cyclic, and location 0000 follows location 7777.

Opcode **61dm**

Mnemonic CRM d,m

Instruction Central read (d) words from (A) to m

Format



Remarks PP location 0000 is used by hardware. This instruction disassembles 60-bit words from central memory into 12-bit words and places these in consecutive PP memory locations, beginning with the leftmost 12 bits of the first 60-bit word.

The parameters of the transfer are as follows: If bit 17 of A is zero, A bits 0 through 16 contain the absolute address of the first 60-bit word transferred. If bit 17 of A is one, hardware adds relocation register R to zero-extended A bits 0 through 16 to obtain the absolute address of the first 60-bit word transferred. For further information, refer to R Register under Input/Output Unit in the chapter entitled Functional Descriptions for the applicable computer system, and PP Relocation Register Format at the beginning of this section under PP Instruction Descriptions. PP location d must contain the number of 60-bit words transferred. Field m gives the PP location into which the first 12-bit word is placed.

This instruction stores P plus 1 into PP location 0000 before beginning the transfer. After the transfer is completed, the next instruction is taken from one plus whatever address is stored in location 0000. If the transfer overwrites location 0000, execution resumes at the location specified by (0000) plus 1 and results are undefined. (PP memory addressing is cyclic, and location 0000 follows location 7777.)

The A register is incremented by one after each 60-bit word is read from central memory. If the incrementing changes A bit 17, the central memory addressing is switched between direct address and relocation address modes. Refer to Central Memory Addressing by PPs in chapter 17.

After the transfer is completed, the A register contains either the address of the last word transferred plus 1 (direct addressing) or the same address less the contents of the relocation address register (relocation addressing), except as follows: If the last word transferred is from a relative address 3777768 and relocation is in effect, then the A register is cleared, and the value returned in A may not point to the last word transferred plus 1.

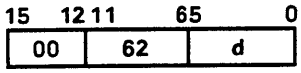
Central Write

Opcode **62d**

Mnemonic CWD d

Instruction Central write to (A) from d

Format



Remarks This instruction assembles five 12-bit words from consecutive PP memory locations into one 60-bit word and stores the 60-bit word in central memory. The first 12-bit word is stored in the leftmost 12 bits of the 60-bit word. (PP memory addressing is cyclic, and location 0000 follows location 7777.)

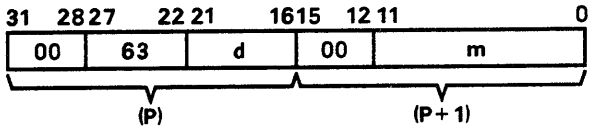
The parameters of the transfer are as follows: If bit 17 of A is zero, A bits 0 through 16 contain the absolute address of the 60-bit word stored. If bit 17 of A is one, hardware adds relocation register R to zero-extended A bits 0 through 16 to obtain the absolute address of the 60-bit word stored. For further information, refer to R Register under Input/Output Unit in chapter 2, and PP Relocation Register Format under PP Instruction Descriptions. Field d gives the PP location of the first 12-bit word transferred. The transfer is subject to the CM bounds test.

Opcode **63dm**

Mnemonic CWM m,d

Instruction Central write (d) words to (A) from m

Format



Remarks Hardware uses PP location 0000. This instruction assembles 12-bit words from consecutive PP memory locations into 60-bit words and stores these in central memory. The first 12-bit word is stored in the leftmost 12 bits of the 60-bit word. (PP memory addressing is cyclic, and location 0000 follows location 7777.)

The parameters of the transfer are as follows: If bit 17 of A is zero, A bits 0 through 16 contain the absolute address of the first 60-bit word transferred. If bit 17 of A is one, hardware adds relocation register R to zero-extended A bits 0 through 16 to obtain the absolute address of the first 60-bit word transferred. For further information, refer to R Register under Input/Output Unit in the chapter entitled Functional Descriptions for the applicable computer system, and in PP Relocation Register Format at the beginning of this section width PP Instruction Descriptions. PP location d must contain the number of 60-bit words transferred. Field m gives the PP location from where the first 12-bit word is obtained. The transfer is subject to the CM bounds test. This instruction stores P plus 1 into PP location 0000 before beginning the transfer. After the transfer is completed, the next instruction is taken from one plus whatever address is stored in location 0000.

The A register is incremented by one after each 60-bit word is written into central memory. If the incrementing changes A bit 17, the central memory addressing is switched between direct address and relocation address modes. Refer to Central Memory Addressing by PPs in chapter 17.

After the transfer is completed, the A register contains either the address of the last word transferred plus 1 (direct addressing) or the same address less the contents of the relocation address register (relocation addressing), except as follows: If the last word transferred is from a relative address 3777776_8 and relocation is in effect, then the A register is cleared, and the value returned in A may not point to the last word transferred plus 1.

PP I/O Instructions

The PP input/output instructions (table 16-23) direct activity on the I/O channels. They select an external device and transfer data to or from that device. The instructions also determine whether a channel or external device is available and ready to transfer data. The preparatory steps ensure that the channels carry out an orderly data transfer. Each external device has a set of external function codes that the PP uses to establish operation modes, and to start and stop data transfer. The devices can also detect certain errors that are indicated to the controlling PP.

Table 16-23. PP Input/Output Instructions

Opcode	Format	Instruction	Mnemonic
641	cm	Test and set channel c flag	SCF m,40B+c
651	cm	Clear channel c flag	CCF c
70	d	Input to A from channel d	IAN d
71	dm	Input A words to m from channel d	IAM m,d
72	d	Output from A on channel d	OAN d
73	dm	Output (A) words from m on channel d	OAM m,d
74	d	Activate channel d	ACN d
75	d	Deactivate channel d	DCN d
76	d	Function A on channel d	FAN d
77	dm	Function m on channel d	FNC m,d

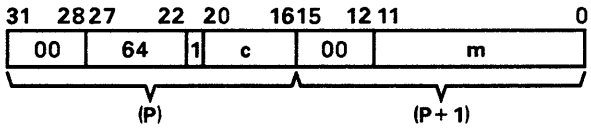
Test/Clear

Opcode **641cm**

Mnemonic SCF m,c

Instruction Test and set channel c flag

Format



Remarks If the channel c flag is set, this instruction causes a jump to m. If the channel c flag is clear, it sets this flag and continues with the next instruction. When m is set to P plus 2, the channel flag is unconditionally set when the program reaches P plus 2.

If two or more PPs simultaneously issue this instruction for the same channel, the conflict is resolved as follows:

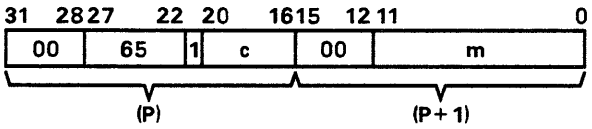
If one of the competing channels is channel 17 (maintenance channel), the PP in the lowest physical level sees the true condition of the flag; the other conflicting PPs see the flag set (and hence take a jump). If the competing channel is any other channel, software must resolve the conflict. Any five consecutively numbered PPs (in the same barrel) issue instructions at different times.

Opcode **651cm**

Mnemonic CCF m,c

Instruction Clear channel c flag

Format



Remarks This instruction clears the channel c flag. The m field is required but is not used.

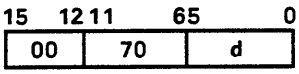
Input/Output

Opcode **70d**

Mnemonic IAN d

Instruction Input to A from channel d

Format



Remarks This instruction transfers a word from input channel d to the lower 12 bits of the A register. The upper 6 bits of A are cleared to zero.

NOTE

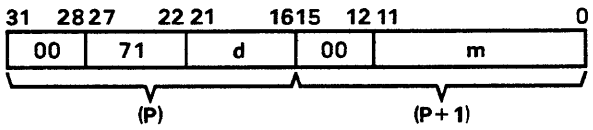
If bit 5 of d is clear and the channel is inactive, this instruction hangs the PP, waiting for the channel to go active and full, if executed. If bit 5 of d is set and the channel is inactive or deactivated before a full is received, the instruction exits. The word is not accepted, and the A register clears.

Opcode **71dm**

Mnemonic IAM m,d

Instruction Input A words to m from channel d

Format



Remarks This instruction transfers a block of 12-bit words from input channel d to PPM. The first word goes to the PPM address specified by m. The A register holds the block length. A reduces by one as each word is read. The input operation completes when A equals zero or the data channel becomes inactive. If the operation terminates by the channel becoming inactive, the next storage location in PPM is set to zero. However, the word count is not affected by this empty word. Therefore, A holds the block length minus the number of real data words read.

During this instruction, address 0000 temporarily holds P while m is held in the P register. P advances by one to hold the address for the next word as each word is stored.

NOTE

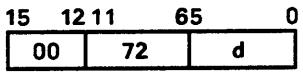
If this instruction executes when the data channel is inactive, no input operation is accomplished, and the program continues at P plus 2. However, the location specified by m is set to zero.

Opcode **72d**

Mnemonic OAN d

Instruction Output from A on channel d

Format



Remarks This instruction transfers a word from the A register (lower 12 bits) to output channel d.

NOTE

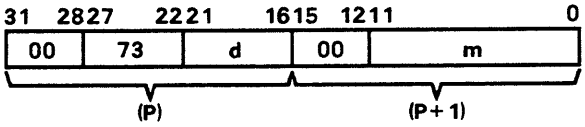
If bit 5 of d is clear and the channel is inactive, this instruction hangs the PP, waiting for the channel to go active and full, if executed. If bit 5 of d is set and the channel is inactive, the program continues at P plus 1. The word is not transferred.

Opcode **73dm**

Mnemonic OAM m,d

Instruction Output A words from m on channel d

Format



Remarks This instruction transfers a block of words from PPM to channel d. The first word is read from the address specified by m. The A register holds the number of words to be sent. A reduces by one as each word is read. The output operation completes when A equals zero or the channel becomes inactive.

During this instruction, address 0000 temporarily holds P while m is held in the P register. P advances by one to give the address of the next word as each word is read from the PPM.

NOTE

If this instruction executes when the data channel is inactive, no output operation is accomplished, and the program continues at P plus 2.

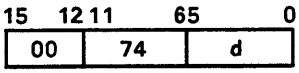
Activate/Deactivate

Opcode **74d**

Mnemonic ACN d

Instruction Activate channel d

Format



Remarks This instruction activates the channel specified by d and sends the active signal on the channel to equipment connected to the channel. Activating a channel, which must precede a 70 through 73 instruction, prepares I/O equipment for the exchange of data.

NOTE

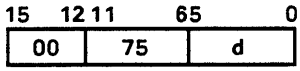
If this instruction executes when the data channel is already active and bit 5 of d is set, the program continues at P plus 1. Otherwise, activating an already active channel causes the PP to wait until the channel goes inactive. The PP hangs if the channel does not go inactive.

Opcode **75d**

Mnemonic DCN d

Instruction Deactivate channel d

Format



Remarks This instruction deactivates the channel specified by d. As a result, the I/O data transfer stops.

NOTE

If this instruction executes when the data channel is already inactive and bit 5 of d is set, the program continues at P plus 1. The channel remains inactive, and no inactive signal is sent to the I/O equipment. Deactivating an already inactive channel causes the PP to hang until the channel becomes active.

If an output instruction is followed by a disconnect instruction without first establishing that the input device (check for channel empty) has accepted the information, the last word transmitted may be lost.

Do not deactivate a channel before putting a useful program in the associated PP. PPs other than 0 are hung on an input instruction (71) after deadstart. Deactivating a channel after deadstart causes an exit to the address specified by the content of location 0000 plus 1 and execution of that program. If the channel is deactivated without a valid program in that PP, the PP executes whatever program was left in PPM. Therefore, the PP could run wild.

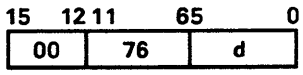
Function

Opcode **76d**

Mnemonic FAN d

Instruction Function A on channel d

Format



Remarks This instruction sends the external function code in the lower 12 bits of the A register on channel d.

NOTE

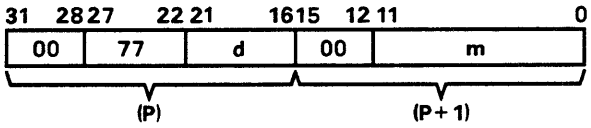
If this instruction executes with bit 5 of d clear and the channel active, PP execution stops until a deadstart or another PP causes the channel to become inactive. If bit 5 of d is set and the channel is active, the program continues at P plus 1. Neither the function signal nor the function word transmits. The channel remains active, and execution continues.

Opcode **77dm**

Mnemonic FNC m,d

Instruction Function m on channel d

Format



Remarks This instruction sends the external function code specified by m on channel d.

NOTE

If this instruction executes with bit 5 of d clear and the channel active, PP execution stops until a deadstart or another PP causes the channel to become inactive. If bit 5 of d is set and the channel is active, the program continues at P plus 2. Neither the function signal nor the function word transmits. The channel remains active, and execution continues.

Other IOU Instructions

Table 16-24 lists the other IOU instructions.

Table 16-24. Other IOU Instructions

Opcode	Format	Instruction	Mnemonic
00	xx	Pass	-
27	d	Pass	
260	x	Exchange Jump	EXN
261	x	Monitor exchange jump	MXN
262	x	Monitor exchange jump to MA	MAN

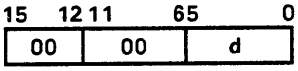
Pass

Opcode **00xx**

Mnemonic PSN

Instruction Pass

Format



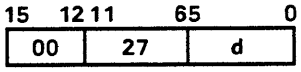
Remarks This instruction specifies that no operation is to be performed. The instruction provides a means of padding out a program.

Opcode **27d**

Mnemonic KPT d

Instruction Pass

Format



Remarks This instruction is not an operation. However, it generates a pulse to a testpoint (keypoint) for optional monitoring by external equipment.

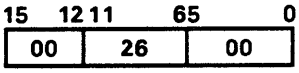
Exchange Jump

Opcode **2600**

Mnemonic EXN

Instruction Exchange jump

Format



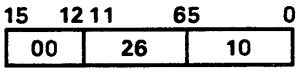
Remarks This instruction causes an unconditional exchange jump in the CP, leaving the CP CYBER 170 monitor flag unaltered. The new CYBER 170 exchange package begins at central memory location R plus A when the leftmost bit in A is set. When this bit is clear, A specifies the address. The PP waits until the exchange is completed before proceeding with the next instruction.

Opcode **2610**

Mnemonic MXN

Instruction Monitor exchange jump

Format



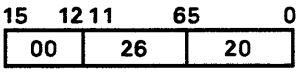
Remarks If the CP is in the CYBER 170 monitor mode, this instruction is a pass. If the CP is in the CYBER 170 job mode, it causes a CYBER 170 exchange jump in the CP, switching the CP to the CYBER 170 monitor mode (MF equals one). The new CYBER 170 exchange package begins at central memory location R plus A when the leftmost bit in A is set. When this bit is clear, A specifies the address. The PP waits until the exchange is completed before proceeding with the next instruction.

Opcode **2620**

Mnemonic MAN

Instruction Monitor exchange jump to MA

Format



Remarks If the CP is in CYBER 170 monitor mode, this instruction is a pass. If the CP is in CYBER 170 job mode, it causes a CYBER 170 exchange jump in the CP, switching the CP to CYBER 170 monitor mode (MF equals one). The new CYBER 170 exchange package begins at the absolute address given in the MA field of the outgoing CYBER 170 exchange package. The PP waits until the exchange is completed before proceeding with the next instruction.

Instruction Execution Timing

Table 16-25 lists approximate execution times for the PP instructions. These times are listed with the assumption that no conflicts occur. The numbers in the timing notes column refer to the notes at the end of the table. Execution times are given in 250-ns major cycles.

NOTE

These execution times are approximations only and are subject to change without notice. Accurate timings can come only from benchmark tests. Control Data Corporation is not responsible for assumptions made based on the times listed here.

Table 16-25. PP Instruction Timing

Instruction Code	Description	Execution Time in 250-ns	Timing Notes
00xx	Pass	1	-
01dm	Long jump to m + (d)	3	-
02dm	Return jump to m + (d)	4	-
03d	Unconditional jump d	1	-
04d	Zero jump d	1	-
05d	Nonzero jump d	1	-
06d	Plus jump d	1	-
07d	Minus jump d	1	-
10d	Shift d	1	-
11d	Logical difference d	1	-
12d	Logical product d	1	-
13d	Selective clear d	1	-
14d	Load d	1	-
15d	Load complement d	1	-
16d	Add d	1	-
17d	Subtract d	1	-
20dm	Load dm	2	-
21dm	Add dm	2	-
22dm	Logical product dm	2	-
23dm	Logical difference dm	2	-
24d	Load R register from (d) and (d) + 1	3	-
25d	Store R register at (d) and (d) + 1	4	-

(Continued)

Table 16-25. PP Instruction Timing *(Continued)*

Instruction Code	Description	Execution Time in 250-ns	Timing Notes
260x	Exchange jump	2	1
261x	Monitor exchange jump	2	1
262x	Monitor exchange jump to MA	2	1
27d	Pass	1	-
30d	Load (d)	2	-
31d	Add (d)	2	-
32d	Subtract (d)	2	-
33d	Logical difference (d)	2	-
34d	Store (d)	2	-
35d	Replace add (d)	4	-
36d	Replace add one (d)	5	-
37d	Replace subtract one (d)	5	-
40d	Load ((d))	3	-
41d	Add ((d))	3	-
42d	Subtract ((d))	3	-
43d	Logical difference ((d))	3	-
44d	Store ((d))	3	-
45d	Replace add ((d))	5	-
46d	Replace add one ((d))	6	-

Timing Notes:

1. No assembly-disassembly unit (ADU) conflicts and no outstanding CYBER 170 exchange jump request in the ADU.

(Continued)

Table 16-25. PP Instruction Timing *(Continued)*

Instruction Code	Description	Execution Time in 250-ns	Timing Notes
47d	Replace subtract one ((d))	6	-
50dm	Load (m + (d))	4	-
51dm	Add (m + (d))	4	-
52dm	Subtract (m + (d))	4	-
53dm	Logical difference (m + (d))	4	-
54dm	Store (m + (d))	4	-
55dm	Replace add (m + d))	6	-
56dm	Replace add one (m + (d))	7	-
57dm	Replace subtract one (m + (d))	7	-
60d	Central read from (A) to d	12	1
61dm	Central read (d) words from (A) to m	-	1,2
62d	Central write to (A) from d	6	1
63dm	Central write (d) words to (A) from m	-	1,3
640cm	Jump to m if channel c active	2	-
641cm	Test and set channel c flag	2	-
650cm	Jump to m if channel c inactive	2	-
651cm	Clear channel c flag	2	-
660cm	Jump to m if channel c full	2	-

Timing Notes:

1. No ADU conflicts. No central memory conflicts. Add a possible trip due to resynchronization (CM read instructions only).

2. Seven major cycles for instruction set-up and instruction exit. Five major cycles for every CM word.

3. Six major cycles for instruction set-up and instruction exit. Five major cycles for every CM word.

(Continued)

Table 16-25. PP Instruction Timing *(Continued)*

Instruction Code	Description	Execution Time in 250-ns	Timing Notes
661cm	Jump to m if channel c error flag set	2	-
670cm	Jump to m if channel c empty	2	-
671cm	Jump to m if channel c error flag clear	2	-
70d	Input to A from channel d	2	-
71dm	Input A words to m from channel d	-	1
72d	Output from A on channel d	2	-
73dm	Output (A) words from m on channel d	-	1
74d	Activate channel d	2	-
75d	Deactivate channel d	2	-
76d	Function A on channel d	2	-
77dm	Function m on channel d	2	-

Timing Notes:

1. Five major cycles for instruction set-up and exit. One major cycle per word (nonconflict case) or two major cycles per word (conflict case).

Nonconflict case occurs when two PPs communicating to each other are not in the slot at the same time.

Conflict case occurs when two PPs communicating with each other are in the slot at the same time.

Programming Information 17

CP Programming 17-1
CYBER 170 Exchange Jump 17-1
Executive State 17-4
Floating-Point Arithmetic 17-4
Format 17-4
Packing 17-5
Overflow 17-7
Underflow 17-7
Indefinite 17-7
Nonstandard Operands 17-8
Normalized Numbers 17-10
Rounding 17-10
Double-Precision Results 17-10
Fixed-Point Arithmetic 17-12
Integer Arithmetic 17-13
Compare/Move Arithmetic 17-13
Instruction Lookahead Purge Control 17-14
Model 835 Purge Control 17-14
Models 840, 845, 850, 855, and 860 Purge Control 17-14
Model 990 and CYBER 990E, 995E, and 994 Purge Control 17-15
Error Response 17-15
Illegal Instructions 17-24
Processor-Detected Malfunctions (EC=20_8) 17-25
Conditional Software Errors (EC=01_8, 02_8, and 04_8) 17-25
Monitor Condition Register Errors (EC=67_8) 17-25

Memory Programming 17-26
Addressing Modes 17-28
Direct Read/Write Instructions (014, 015, 660, 670) 17-28
Block Copy Instructions (011, 012) 17-28

PP Programming 17-29
Central Memory Addressing by PPs 17-29
PP Memory Addressing by PPs 17-29
Direct 6-Bit Operand 17-29
Direct 18-Bit Operand 17-29
Direct 6-Bit Address 17-30
Direct 12-Bit Address 17-30
Indexed 12-Bit Address 17-30
Indirect 6-Bit Address 17-30
Central Memory Read/Write Instructions 17-31
PP Central Memory Read Instructions (60, 61) 17-31
PP Central Memory Write Instructions (62, 63) 17-31
Input/Output Channel Communications 17-32
Inter-PP Communications 17-34
PP Program Timing Considerations 17-34
Channel Operation 17-34
Channel Control Flags 17-34
Channel Active/Inactive Flag 17-35
Register Full/Empty Flag 17-35
Channel (Marker) Flag Instructions (641, 651) 17-36
Error Flag Instructions (661, 671) 17-36
Channel Transfer Timing 17-36

Input/Output Transfers 17-38
Data Input Sequence 17-38
Data Output Sequence 17-40

System Console Programming 17-42
Keyboard 17-42
Data Display 17-42
Character Mode 17-42
Dot Mode 17-45
Codes 17-45
Programming Example 17-47
Programming Timing Considerations 17-47

Real-Time Clock Programming 17-49

Two-Port Multiplexer Programming 17-49
Two-Port Multiplexer Operation 17-50
Terminal Select (7XXX) 17-50
Terminal Deselect (6XXX) 17-50
Read Status Summary (00XX) 17-51
PP Read Terminal Data (01XX) 17-51
Data Set Ready (Bit 52) 17-51
Data Set Ready (DSR) and Data Carrier Detector (DCD) (Bit 53) 17-51
Over Run (Bit 54) 17-51
Framing or Parity Error (Bit 55) 17-52
Data Character (Bits 56 through 63) 17-52
PP Write Output Buffer (02XX) 17-52
Set Operation Mode to the Terminal (03XX) 17-53
Set/Clear Data Terminal Ready (04XX) 17-54
Set/Clear Request to Send (05XX) 17-54
Master Clear (07XX) 17-54
Programming Considerations 17-55
Output Data 17-55
Input Data 17-55
Request to Send and Data Terminal Ready 17-55

Maintenance Channel Programming 17-56
Maintenance Channel 17-56
MCH Function Words 17-57
MCH Control Words 17-61
MCH Programming for Halt/Start (Opcode 0/1) 17-61
MCH Clear LED (Opcode 3) 17-61
MCH Programming for Read/Write (Opcode 4/5) 17-62
MCH Programming for Master Clear/Clear Errors (Opcode 6/7) 17-63
MCH Echo (Opcode 8) 17-63
MCH Programming for Read IOU Status Summary (Opcode C, IOU Only) .. 17-63

Programming Information 17

This chapter contains special programming information about the CP, CM, PPs, system console, real-time clock, two-port multiplexer, and maintenance channel.

CP Programming

CYBER 170 Exchange Jump

The CP operates in either CYBER 170 job mode, which is interruptable, or CYBER 170 monitor mode, which is not interruptable. A hardware flag called the CYBER 170 monitor flag (MF) indicates the mode in which the CP is executing a job.

The CP uses a CYBER 170 exchange jump operation to switch from CYBER 170 job mode to CYBER 170 monitor mode and back again. The execution of a CYBER 170 exchange jump permits the CP to send pertinent information from the operating and control registers to CM and permits CM to send new information to the same registers. The information that flows from and into the operating and control registers during a CYBER 170 exchange jump is called a CYBER 170 exchange package (figure 17-1).

The CP 013 instruction and the PP 2600, 2610, and 2620 instructions initiate a CYBER 170 exchange jump operation. A CYBER 170 exchange jump instruction starts or interrupts the CP and provides CM with the first address of a 16-word exchange package. For the 013 instruction with MF set (CP in monitor mode), the starting address of the CYBER 170 exchange package is Bj plus K. With MF clear (CP in job mode), the address is the monitor address (MA). For the 2600 instruction, the CYBER 170 exchange package address is A plus R when bit 17 of the A register is set. When this bit is clear, the address is A. For the 2610 instruction with MF set, the instruction is a pass. With MF clear, the CYBER 170 exchange package address is A plus R when bit 17 of the A register is set. When this bit is clear, the address is A. For the 2620 instruction with MF set, the instruction is a pass. With MF clear, the CYBER 170 exchange package address is MA of the outgoing CYBER 170 exchange package.

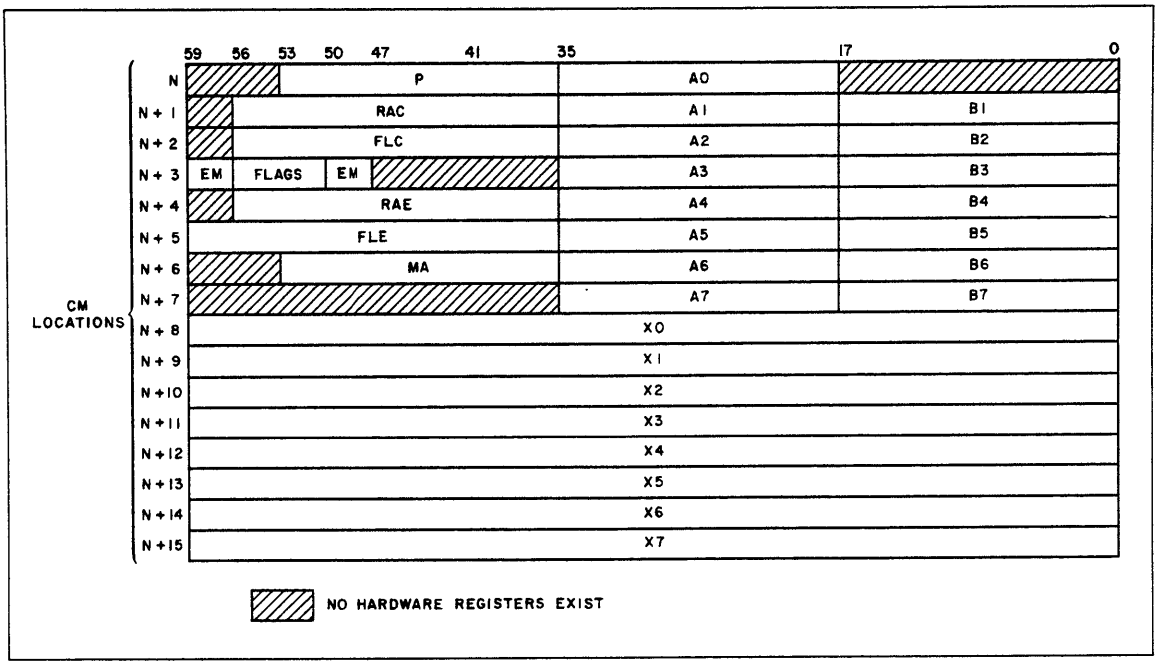


Figure 17-1. CYBER 170 Exchange Package

The CYBER 170 exchange package contains the following registers which provide information for program execution.

- 18-bit program address (P) register.
- 21-bit reference address for CM (RAC) register.
- 21-bit field length for CM (FLC) register.
- 6-bit exit mode (EM) register.
- 6-bit flag register.
- 21- or 24-bit reference address for UEM (RAE); 21 bits with lower 6 bits assumed to be zero in standard addressing mode; 24 bits right-shifted with 6 bits assumed to be zeros in expanded addressing mode.
- 21- or 24-bit field length for UEM (FLE); 21 bits in standard addressing mode and 24 bits in expanded addressing mode; lower 6 bits are assumed to be zero.
- 18-bit monitor address (MA) register.
- Initial contents of eight 60-bit X registers.
- Initial contents of eight 18-bit A registers.
- Initial contents of 18-bit B registers B1 through B7; B0 contains a constant zero.

The time that a particular CYBER 170 exchange package resides in the CP hardware registers is the execution interval. The execution interval begins with a CYBER 170 exchange jump that swaps the CYBER 170 exchange package information in CM with the information contained in the CP registers. The execution interval ends with the next CYBER 170 exchange jump.

Executive State

The executive state uses a combination of hardware, software, and microcode to handle the following items.

- System initialization.
- Compare/move instructions.
- Software errors and unimplemented instructions that occur in CYBER 170 monitor mode.
- Processor-detected hardware errors.
- Hardware integrity verification (diagnostics).

In General, executive state determines the cause of an interrupt and decides whether to return the CP to the interrupted mode, to halt the CP, or to simulate a CYBER 170 exchange and return control to CYBER 170 monitor mode. Refer to Error Response in this chapter.

Floating-Point Arithmetic

Format

Floating-point arithmetic expresses a number in the form kB^n.

k = Coefficient

B = Base number

n = Exponent or power to which the base number is raised

B is assumed to be 2 for binary-coded quantities. In the 60-bit, floating- point format (figure 17-2), the binary point is considered to be to the right of the coefficient. The lower 48 bits express the integer coefficient, which is the equivalent of 15 decimal digits. The sign of the coefficient is separated from the rest of the coefficient and appears in the highest-order bit of the packed word. Negative numbers are represented in ones complement notation. The exponent is biased by complementing the exponent sign bit.

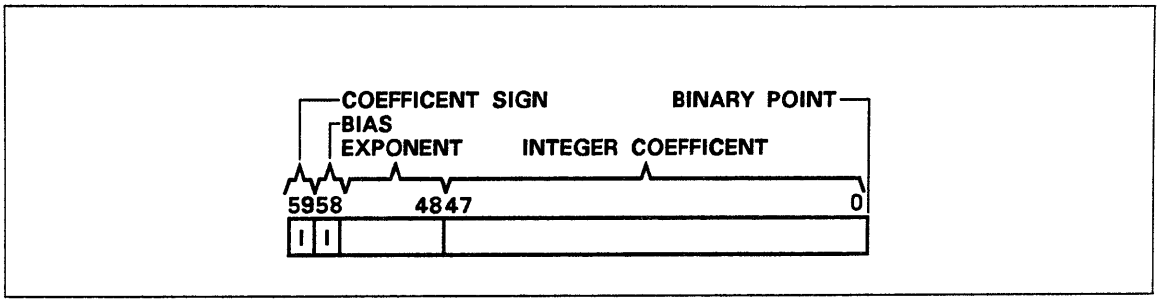


Figure 17-2. Floating-Point Format

Table 17-1 summarizes the configurations of bits 58 and 59 and the implications regarding signs of the possible combinations.

Table 17-1. Bits 58 and 59 Configurations

Bit 59	Bit 58	Coefficient Sign	Exponent Sign
0	1	Positive	Positive
0	0	Positive	Negative
1	0	Negative	Positive
1	1	Negative	Negative

Packing

Packing refers to the conversion of numbers in the form kB^n to floating-point format. A shortcut method of packing exponents can be derived by considering the representation of positive zero and negative zero exponents. Assuming a positive coefficient, zero exponents are packed as follows:

Positive zero exponent: 2000x,...,x

Negative zero exponent: 1777x,...,x

Since positive exponents are expressed in true form, begin with a bias of 2000 (positive zero) and add the magnitude of the exponent. The range of positive exponents is 0000 through 1777. In packed form, the range is 2000 through 3777.

When the coefficient is negative, the packed positive exponent is complemented to become 5777 through 4000.

Negative exponents are expressed in complement form by beginning with a bias of 1777 (negative zero) and then subtracting the magnitude of the exponent. The range of negative exponents is negative 0000 through negative 1777. In packed form, the range is 1777 through 0000.

When the coefficient is negative, the packed negative exponent is complemented to become 6000 through 7777.

Examples of packed and unpacked floating-point numbers are shown in octal notation to illustrate the packing process. Examples 1 and 2 are different forms of the integer positive 1. Example 3 is positive 100 (decimal), and example 4 is negative 100 (decimal). Examples 5 and 6 are large and small positive numbers. The unpacked values are shown as they might appear in the X and B registers prior to a pack operation.

The packed negative zero exponent is not used for normal operation. Instead, 1777 is used to indicate the special error condition of indefinite.

1.	Unpacked coefficient	0000	0000	0000	0000	0001
	Unpacked exponent	00	0000			
	Packed format	2000	0000	0000	0000	0001
2.	Unpacked coefficient	0000	4000	0000	0000	0000
	Unpacked exponent	77	7720			
	Packed format	1720	4000	0000	0000	0000
3.	Unpacked coefficient	0000	6200	0000	0000	0000
	Unpacked exponent	77	7726			
	Packed format	1726	6200	0000	0000	0000
4.	Unpacked coefficient	7777	1577	7777	7777	7777
	Unpacked exponent	77	7726			
	Packed format	6051	1577	7777	7777	7777
5.	Unpacked coefficient	0000	4771	3000	0044	7021
	Unpacked exponent	00	1363			
	Packed format	3363	4771	3000	0044	7021
6.	Unpacked coefficient	0000	6301	0277	4315	6033
	Unpacked exponent	77	6210			
	Packed format	0210	6301	0277	4315	6033

Overflow

Overflow of the floating-point range is indicated by an exponent value of positive 1777 (3777 or 4000 in packed form). This is the largest exponent value that can be represented in the floating-point format. This exponent value may result from the calculation in which this exponent value, together with the computed coefficient value, is a correct representation of the result. This situation is called a partial overflow. However, further computation using this result generates an overflow.

A complete overflow occurs whenever a result requires an exponent larger than positive 1777. In this case, a complete overflow value results. This result has a positive 1777 exponent and a zero coefficient. The sign of the coefficient is the same as that which generates if the result had not overflowed the floating-point range.

Underflow

Underflow of the floating-point range is indicated by an exponent value of negative 1777 (0000 or 7777 in packed form). This is the smallest exponent value that can be represented in the floating-point format. This exponent value may result from the calculation in which this exponent value, together with the computed coefficient value, is a correct representation of the result. This situation is called a partial underflow. Further computation using this result may be detected as an underflow.

A complete underflow occurs whenever a result requires an exponent smaller than negative 1777. In this case, a complete underflow value results. This result has a negative 1777 exponent and a zero coefficient. The complete underflow indicator is a word of all zeros, and it is the same as a zero word in integer format.

Indefinite

An indefinite result indicator generates whenever the calculation is unresolvable. An example is division when the divisor is 0 and the dividend is also 0. Another example is multiplication of an overflow number times an underflow number. The indefinite result indicator is a value that cannot occur in normal floating-point calculations. This indicator corresponds to a negative zero exponent and a zero coefficient (177770,...,0 in packed form).

Any indefinite indicator used as an operand generates an indefinite result no matter what the other operand value is. Although indefinite indicators always generate with a positive sign, they may occur as operands with a negative sign.

Nonstandard Operands

In summary, the special operand forms in octal are:

Positive overflow (+∞)	3777x,...,x
Negative overflow (-∞)	4000x,...,x
Positive indefinite (+IND)	1777x,...,x
Negative indefinite (-IND)	6000x,...,x
Positive underflow (+0)	0000x,...,x
Negative underflow (-0)	7777x,...,x

Table 17-2 through 17-5 indicate the resulting forms when various combinations of underflow, overflow, and indefinite forms are used in floating-point operations. The designations W and N are defined as follows:

W Any word except + ∞ and + IND

N Any word except + ∞ , + IND, and + 0

Table 17-2. Xj Plus Xk (30, 32, and 34 Instructions)

		Xk			
		W	+∞	-∞	+ IND
Xj	W		+∞	-∞	IND
	+∞	+∞	+∞	IND	IND
	-∞	-∞	IND	-∞	IND
	± IND	IND	IND	IND	IND

Table 17-3. Xj Minus Xk (31, 33, and 35 Instructions)

		Xk			
		W	+∞	-∞	+ IND
Xj	W		-∞	+∞	IND
	+∞	+∞	IND	+∞	IND
	-∞	-∞	-∞	IND	IND
	± IND	IND	IND	IND	IND

Table 17-4. Xj Multiplied by Xk (40, 41, and 42 Instructions)

		Xk						
		+N	-N	+0	-0	+∞	-∞	+ IND
Xj	+N			0	0	+∞	-∞	IND
	-N			0	0	-∞	+∞	IND
	+0	0	0	Integer † multiply		IND	IND	IND
	-0	0	0			IND	IND	IND
	+∞	+∞	-∞	IND	IND	+∞	-∞	IND
	-∞	-∞	+∞	IND	IND	-∞	+∞	IND
	± IND	IND	IND	IND	IND	IND	IND	IND

† If both operands used in the integer multiply are normalized, an underflow results.

Table 17-5. Xj Divided by Xk (44 and 45 Instructions)

		Xk						
		+N	-N	+0	-0	+∞	-∞	+ IND
Xj	+N			+∞	-∞	0	0	IND
	-N			-∞	+∞	0	0	IND
	+0	0	0	IND	IND	0	0	IND
	-0	0	0	IND	IND	0	0	IND
	+∞	+∞	-∞	+∞	-∞	IND	IND	IND
	-∞	-∞	+∞	-∞	+∞	IND	IND	IND
	± IND	IND	IND	IND	IND	IND	IND	IND

Normalized Numbers

A normalized floating-point number has as large a coefficient and as small an exponent as possible. A floating-point number in packed format is normalized if the coefficient sign bit is different from bit 47. This condition indicates that the coefficient has been left-shifted until bit 47 contains the most-significant bit in the coefficient; therefore, the floating-point number has no leading sign bits in the coefficient. The normalized instructions perform the coefficient shift. The floating-multiply and floating-divide instructions deliver normalized results when provided with normalized operands. The floating-add instructions may deliver unnormalized results even when both operands are normalized. Therefore, it is necessary to perform the normalize operation after each sequence of floating-add or floating-subtract operation if the result is to be kept in a normalized form.

Rounding

Floating-point instructions round the results in single-precision computation. These instructions execute in the same amount of time as the unrounded versions. The operands are modified to accomplish the rounding function. The amount of bias introduced by the rounding operation varies and is affected by the coefficient value in the operands. The descriptions of the round instructions define the effects of rounding in detail.

Double-Precision Results

The floating-point arithmetic instructions generate double-precision results. Use of unrounded instructions allows separate recovery of upper- and lower-half results with proper exponents. Rounded instructions allow only upper-half results to be obtained. Two instructions, one single-precision and one double-precision, are required to retrieve an entire double-precision result.

To add or subtract two floating-point numbers, the coefficient with the smaller exponent enters the upper half of an accumulator and is right-shifted by the difference of the exponents. The other coefficient is then added into the upper half of the accumulator. The result is a double-length register (figure 17-3).

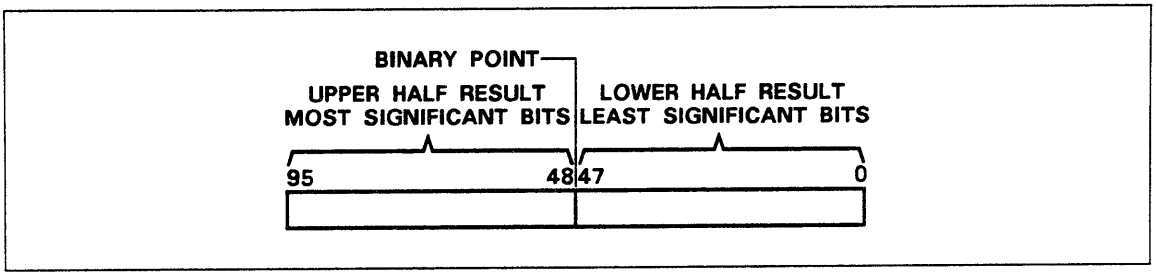


Figure 17-3. Floating-Add Result Format

If single precision is selected, the upper 48 bits of the 96-bit result and the larger exponent are returned as the result. Selecting double precision causes only the lower 48 bits of the 96-bit result and the larger exponent minus 60 (octal) to be returned as the result. The subtraction of 60 (octal) is necessary because the binary point is effectively moved from the right of bit 48 to the right of bit 0. A 96-bit product generates from two 48-bit coefficients. The result of a multiply is a double-length register (figure 17-4).

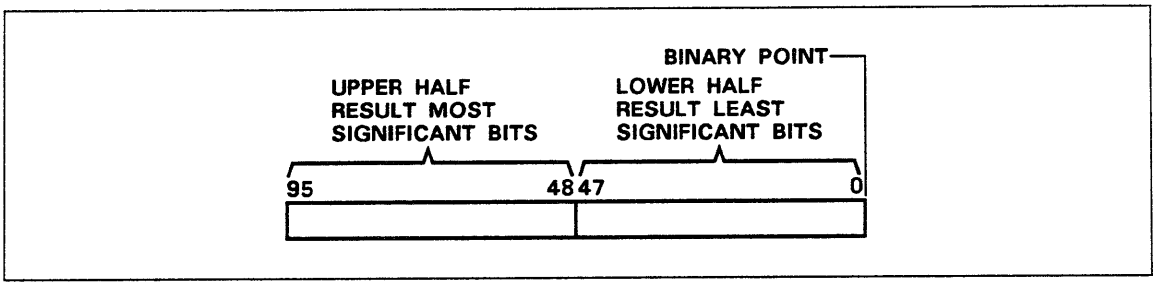


Figure 17-4. Multiply Result Format

If single precision is selected, the upper 48 bits of the product and the sum of the exponents plus 60 (octal) are returned as the result. The addition of 60 (octal) is necessary because the binary point effectively moves from the right of bit 0 to the right of bit 48 when the upper half of the 96-bit result is selected. If double precision is selected, the result is the lower 48 bits of the product and the sum of the exponents.

Fixed-Point Arithmetic

Fixed-point addition and subtraction of 60-bit numbers are handled by the long-add instructions (36 and 37). Negative numbers are represented in ones complement notation, and overflows are ignored. The sign bit is in the high-order bit position (bit 59), and the binary point is to the right of the low-order bit position (bit 0).

The increment instructions (50 through 77) handle fixed-point addition and subtraction of 18-bit numbers. Negative numbers are represented in ones complement notation, and overflows are ignored. The sign bit is in the high-order bit position (bit 17), and the binary point is to the right of the low-order position (bit 0).

Integer multiplication is handled as a subset operation of the floating-multiply (42) instruction. The integer multiply requires that both 47-bit integer operands have zero exponents and are not normalized. The result is 48 bits with sign extension. Normalized operands cause underflow results to be reported. If the results exceed 48 bits, overflow is not detected.

An integer divide takes several steps. For example, an integer quotient X1 equal to X2/X3 is produced by the following steps.

Instructions	Remarks
1. Pack X2 from X2 and B0	Pack X2
2. Pack X3 from X3 and B0	Pack X3
3. Normalize X3 in X0 and B0	Normalize X3 (divisor) Normalize X2 (dividend)
4. Normalize X2 in X2 and B0	Divide
5. Floating quotient of X2 and X0 to Xi	Unpack quotient
6. Unpack X1 to X1 and B7	Shift to integer position
7. Shift X1 nominally left B7 places	

The divide requires that both integer (2^{47} maximum) operands must be in floating-point format, and the dividend coefficient must be less than two times the divisor coefficient. The normalize X3 instruction ensures this condition.

The normalize X3 instruction left-shifts the divisor n places ($n \geq 0$), providing a divisor exponent of negative n. The quotient exponent is then 0 minus (-n) minus 48 equals n minus $48 < 0$.

After unpacking and left-shifting nominally, the negative (or zero) value in B7 right-shifts the quotient 48 minus n places, producing an integer quotient in X1. A remainder may be obtained by an integer multiply of X1 and X3 and subtracting the result from X2.

Integer Arithmetic

Integer divide packs the integers into floating-point format, using the pack instruction with a zero-exponent value.

In integer multiplication, a 48-bit product can be formed by using the double-precision multiply instruction. Both operands must have an exponent value of ± 0, and the coefficients cannot both be normalized. The result is sign-extended to 60 bits and sent to an X register.

In integer division, the divisor must be normalized, but the dividend does not have to be normalized. The resulting quotient must be unpacked and the coefficient must be shifted by the amount of the unpacked exponent using the left-shift (22) instruction to obtain the integer quotient.

Compare/Move Arithmetic

The compare/move arithmetic provides multiple-character manipulation. The characters are 6 bits long. Characters can be moved from one CM location to another, and fields of characters can be compared either directly or through a collate table.

The move direct instruction moves a field of up to 127 characters from one location to another location as specified in the instruction. The move indirect instruction performs the same kind of move, but a CM reference is used to obtain the parameters. The move indirect instruction moves a field of up to 8181 characters.

The compare collated instruction compares two fields of up to 127 characters. When two characters are unequal, the characters are referenced in a collate table, and the values are compared. If those values are unequal, the field with the larger character is indicated. The compare uncollated instruction compares two fields of up to 127 characters and indicates the larger of the first character pair that is found to be unequal.

CMU instructions are provided for compatibility with previous systems. For better performance, recompile jobs to avoid use of CMU instructions.

Instruction Lookahead Purge Control

Prefetching of instructions at a branch target address by instruction lookahead hardware can lead to program failures if a program modifies its own code dynamically. Under normal conditions, the lookahead registers are purged by execution of a return jump instruction (010), UEM read instruction (011), exchange jump instruction (013), or unconditional branch instruction (02). Selecting extended purge control extends these conditions. When extended purge control is in effect, lookahead registers are also purged by execution of any conditional jump instruction (03 through 07) or any CM store instruction (50 through 57 when i equals 6 or 7). To enable extended purge control, the system sets bit 52 of the flag register in the CYBER 170 exchange package. When self-modifying code is present, it may be helpful to set extended purge control; however, the additional purging causes a degradation in execution and does not cover all cases of code modification.

Model 835 Purge Control

If normal purge conditions are in effect, a store instruction that modifies a sequential instruction must modify at least P plus five words ahead to ensure execution of the modified code. A store instruction followed by a branch to a modified instruction will execute the modified code only if that code is at least at the branch's target address plus two words, or the branch is at least at P plus four words following the store instruction.

If the extended purge option is selected, a store instruction can modify the next sequential instruction and be assured of executing the modified instruction. Likewise, a store instruction followed by a branch to a modified instruction always executes the modified code.

Models 840, 845, 850, 855, and 860 Purge Control

If normal purge conditions are in effect, a store instruction that modifies a sequential instruction must modify at least P plus six words ahead to ensure execution of the modified code. In addition, a store instruction followed by a branch to a modified instruction will execute the modified code only if there are at least 12 executed instructions between the store and the modified code.

If the extended purge option is selected, a store instruction can modify the next sequential instruction and be assured of executing the modified instruction. Likewise, a store instruction followed by a branch to a modified instruction always executes the modified code.

Model 990 and CYBER 990E, 995E, and 994 Purge Control

If normal purge conditions are in effect, a store instruction that modifies a sequential instruction must modify at least P plus 64 words ahead to ensure execution of the modified code. In addition, a store instruction followed by a branch to a modified instruction will execute the modified code only if there are at least 64 executed instructions between the store and the modified code and only if there are four branch instructions to four different 16-word blocks of instructions between the store and the modified code (the modified code must not reside in one of the blocks jumped to).

If the extended purge option is selected, a store instruction can modify the next sequential instruction and be assured of executing the modified instruction. Likewise, a store instruction followed by a branch to a modified instruction always executes the modified code.

Error Response

When the CP detects or is informed of an error, it records the error. Depending on the type of error and the exit mode selection bits set in the EM register, the program in execution may be interrupted. If the error is an illegal instruction or an address-range error on an RNI or branch, the program interruption is unconditional. For other types of errors, the exit mode selection bits determine whether or not the program is interrupted. If the exit mode selection bit is set and the corresponding condition is detected, the program is interrupted. The exit mode selection bits are contained in word N plus 3 of the exchange package. Figure 17-5 shows the format of the exit condition register at (RAC). Table 17-6 describes the possible contents of the register. Table 17-7 and 17-8 list CP error responses.

The CP has the following error conditions: illegal instructions, processor-detected malfunctions (parity errors), conditional software errors, and monitor condition register (MCR) errors.

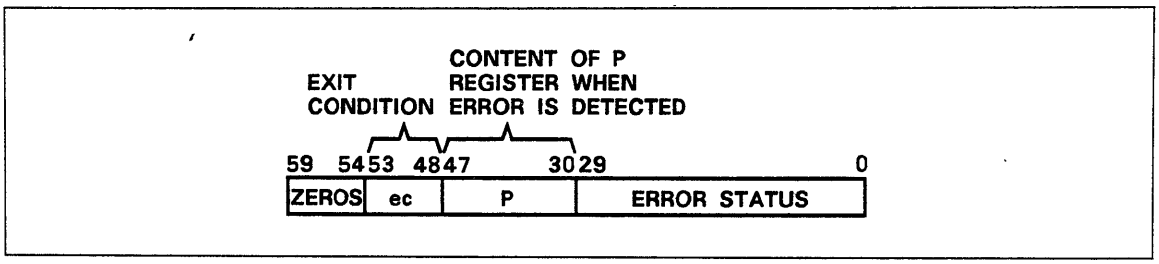


Figure 17-5. Format of Exit Condition Register at (RAC)

Table 17-6. Contents of Exit Condition Register at (RAC)

Field	Description
ec	6-bit exit condition code:

Code (Octal)	Condition
00	Illegal instruction.
01	Address-range error (bit 48).
02	Floating-point infinite (bit 49).
04	Floating-point indefinite (bit 50).
20	Processor-detected malfunction (bit 52).
67	MCR errors.

P	When an error exit occurs, the content of the P register may not correspond to the address of the instruction that caused the error exit. The P register may have been incremented prior to the execution of the instruction.
ERROR STATUS	Nonzero information in bits 0 through 29 is error status for customer engineering and maintenance. MCR uses bits 0 through 15.

Table 17-7. Error Exits in CYBER 170 Monitor Mode (MF=1)

Error Condition	Error Response Exit Mode Selected	Error Response Exit Mode Not Selected
Illegal instruction or 00 instruction. EC=00_8	1. The instruction is not executed. 2. Store P and exit condition bits (00) at location RAC. P equals address of illegal instruction. 3. Interrupt to executive state. 4. CP stops in executive state.	1. N/A (exit mode is always selected).
Exit condition bit 48 set by an incremental read with an address out of range (AOR). EC=01_8	1. The X register is unchanged. 2. The A register contains the AOR address. 3. Store P and exit condition bits (01) at location RAC. P equals address of increment instruction or address of instruction following the increment. 4. Interrupt to executive state. 5. CP stops in executive state.	1. Inhibit read, X unchanged. 2. Continue execution.
Exit condition bit 48 set by an incremental write with an address out of range (AOR). EC=01_8	1. Block write operation; content of CM is unchanged. 2. The A register contains the AOR address. 3. Store P and exit condition bits (01) at location RAC. P equals address of instruction or address of instruction following the increment. 4. Interrupt to executive state. 5. CP stops in executive state.	1. Inhibit write, CM unchanged. 2. Continue execution.

(Continued)

Table 17-7. Error Exits in CYBER 170 Monitor Mode (MF=1) *(Continued)*

Error Condition	Error Response Exit Mode Selected	Error Response Exit Mode Not Selected
Exit condition bit 48 set by an RNI or branch address out of range. EC=01_8	1. Inhibit execution. 2. Store P and exit condition bits (01) at location RAC. P equals address of instruction required by RNI or address of branch destination instruction. 3. Interrupt to executive state. 4. CP stops in executive state.	1. N/A (exit mode is always selected regardless of status of EM register bit 48).
Exit condition bit 48 set on CMU instruction. 1. C1 or C2 greater than 9. 2. K1 or K2 address out of range. EC=01_8	1. Detected by executive state during the execution of compare/move instruction. 2. Condition 1 omits reading/writing; CM is unchanged. Condition 2 causes the instruction to go unexecuted. 3. Store P and exit bits (01) at RAC. 4. CP stops in executive state.	1. Detected by executive state during the execution of compare/move instruction. 2. Condition 1 omits reading/writing; CM is unchanged. Condition 2 causes the instruction to go unexecuted. 3. Continue with next instruction.
Exit condition bit 48 set by a UEM address range check for instructions 011 and 012. EC=01_8	1. Execute instruction as a pass. 2. Store P and exit bits (01) at RAC. 3. Interrupt to executive state. 4. CP stops in executive state.	1. Execute instruction as a pass. 2. Exit to next 60-bit word and continue execution.

(Continued)

Table 17-7. Error Exits in CYBER 170 Monitor Mode (MF=1) *(Continued)*

Error Condition	Error Response Exit Mode Selected	Error Response Exit Mode Not Selected
Exit condition bit 48 set by a UEM address range check for instructions 014 and 015. EC=01_8	1. Execute instruction as a pass. 2. Store P and exit condition bits (01) at RAC. P equals address of following instruction. 3. Interrupt to executive state. 4. CP stops in executive state.	1. Execute instruction as a pass. 2. Exit to next parcel and continue execution.
Exit condition bit 49 set by infinite condition, or bit 50 set by indefinite condition. EC=02_8 or 04_8	1. Store P and exit condition bits (02 for infinite or 04 for indefinite). P equals address of arithmetic instruction or address of instruction following. 2. Interrupt to executive state. 3. CP stops in executive state.	1. Continue execution.
Any hardware parity error or double SECDED error. EC=20_8	1. Interrupt to executive state. 2. Executive state stores P and exit condition bits (20) at RAC. 3. CP stops in executive state.	1. Interrupt to executive state. 2. Executive state stores P and exit condition bits (20) at RAC. 3. CP stops in executive state.

Table 17-8. Error Exits in CYBER 170 Job Mode (MF=0)

Error Condition	Error Response Exit Mode Selected	Error Response Exit Mode Not Selected
Illegal instruction or 00 instruction. EC=00_8	1. The instruction is not executed. 2. Store P and exit condition bits (00) at location RAC. P equals address of illegal instruction. 3. Exchange jump to MA and set CYBER 170 MF.	1. N/A (exit mode is always selected).
Exit condition bit 48 set by an incremental write with an address out of range (AOR). EC=01_8	1. The X register is unchanged. 2. The A register contains the AOR address. 3. Store P and exit condition bits (01) at location RAC. P equals address of increment instruction or address of instruction following the increment. 4. Exchange jump to MA and set CYBER 170 MF.	1. Inhibit read, X unchanged. 2. Continue execution.
Exit condition bit 48 set by an incremental write with an address out of range (AOR). EC=01_8	1. Block write operation; content of CM is unchanged. 2. The A register contains the AOR address. 3. Store P and exit condition bits (01) at location RAC. P equals address of instruction or address of instruction following the increment. 4. Exchange jump to MA and set CYBER 170 MF.	1. Inhibit write, CM unchanged. 2. Continue execution.

(Continued)

Table 17-8. Error Exits in CYBER 170 Job Mode (MF=0) *(Continued)*

Error Condition	Error Response Exit Mode Selected	Error Response Exit Mode Not Selected
Exit condition bit 48 set by an RNI or branch address out of range. EC=01_8	1. Inhibit execution. 2. Store P and exit condition bits (01) at location RAC. P equals address of instruction required by RNI or address of branch destination instruction. 3. Exchange jump to MA and set CYBER 170 MF.	1. N/A (exit mode is always selected regardless of status of EM register bit 48).
Exit condition bit 48 set on CMU instruction. 1. C1 or C2 greater than 9. 2. K1 or K2 address out of range. EC=01_8	1. Detected by executive state during the execution of compare/move instruction. 2. Condition 1 omits reading/writing; CM is unchanged. Conditon 2 causes the instruction to go unexecuted. 3. Store P and exit bits (01) at RAC. 4. Exchange jump to MA and set CYBER 170 MF.	1. Detected by executive state during the execution of compare/move instruction. 2. Condition 1 omits reading/ writing; CM is unchanged. Condition 2 causes the instruction to go unexecuted. 3. Continue with next instruction.
Exit condition bit 48 set by a UEM address range check for instructions 011 and 012. EC=01_8	1. Execute instruction as a pass. 2. Store P and exit bits (01) at RAC. 3. Exchange jump to MA and set CYBER 170 MF.	1. Execute instruction as a pass. 2. Exit to next 60-bit word and continue execution.

(Continued)

Table 17-8. Error Exits in CYBER 170 Job Mode (MF=0) *(Continued)*

To present the information in this chapter in a structured format, this page has been left blank.

(Continued)

Table 17-8. Error Exits in CYBER 170 Job Mode (MF=0) *(Continued)*

Error Condition	Error Response Exit Mode Selected	Error Response Exit Mode Not Selected
Exit condition bit 48 set by a UEM address range check for instructions 014 and 015. EC=01_8	1. Execute instruction as a pass. 2. Stop CP. 3. Store P and exit condition bits (01) at location RAC. 4. Exchange jump to MA and set CYBER 170 MF.	1. Execute instruction as a pass. 2. Exit to next parcel and continue execution.
Exit condition bit 49 set by infinite condition, or bit 50 set by indefinite condition. EC=02_8 or 04_8	1. Store P and exit condition bits (02 for infinite or 04 for indefinite). P equals address of arithmetic instruction or address of instruction following. 2. Exchange jump to MA and set CYBER 170 MF.	1. Continue execution.
Any hardware parity error or double SECDED error. EC=20_8	1. Interrupt to executive state. 2. Executive state stores P and exit condition bits (20) at RAC. 3. Exchange jump to MA and set CYBER 170 MF.	1. Interrupt to executive state. 2. Executive state stores P and exit condition bits (20) at RAC. 3. Exchange jump to MA and set CYBER 170 MF.

Illegal Instructions

An instruction is illegal when it has an illegal operating code, an illegal operating parameter, or when it is positioned so that it begins in one instruction word and extends into the next instruction word. In the CYBER 170 job mode, illegal instructions cause an exchange to the CYBER 170 monitor mode. In the CYBER 170 monitor mode, illegal instructions cause a jump to executive state. The CP stops. CP illegal instructions are:

- 017.
- 011, 012, 013, 464, 465, 466, and 467 if they do not begin at parcel 0.
- 011, 012, 014, and 015 if the UEM enable flag in the flag register of the CYBER 170 exchange package is clear.
- Any 30-bit instruction that begins at parcel 3.

Processor-Detected Malfunctions (EC = 20_8)

Processor detected malfunctions (PDMs) are: data parity errors, address parity errors, and double-bit errors. If the CP is in CYBER 170 job mode, a PDM causes a jump to executive state, which returns to CYBER 170 monitor mode. If the CP is in CYBER 170 monitor mode, a PDM causes a jump to executive state. The CP halts. The instruction being executed when such a fault is detected is not necessarily connected with the fault.

Conditional Software Errors (EC = 01_8, 02_8, and 04_8)

Conditional software errors are caused by address-range errors and floating-point infinite/indefinite operands or results. A conditional software error causes action, depending on bits set in the EM field in the current CYBER 170 exchange package. If the bit reserved for use with the specific type of error is clear, the error is ignored in both CYBER 170 job and CYBER 170 monitor modes. If the bit is set and the error occurs in the CYBER 170 job mode, it causes an exchange to the CYBER 170 monitor mode.

If the bit is set and the error occurs in the CYBER 170 monitor mode, it causes an interrupt to executive state.

Monitor Condition Register Errors (EC = 67_8)

Monitor Condition Register (MCR) errors are generally program environment errors. For a listing of error causes and actions taken following an error, refer to the Monitor Condition Register chart in the Codes Booklet listed in About This Manual.

Memory Programming

All references to CM by the CP for instructions or read/write data are made relative to RAC. The RAC defines the lower limit of the addresses of a program in CM. The upper limit of the program addresses is defined by FLC added to RAC.

All references to UEM by the CP for instructions or read/write data are made relative to RAE. The RAE defines the lower limit of the addresses of a program/data in UEM. The upper limit of the addresses is defined by FLE added to RAE.

The field length is a number of 60-bit words established by the operating system prior to program execution. All references to CM or UEM for a program/data must be within the field established for that program.

During a CYBER 170 exchange jump, RAC and FLC are loaded into respective registers to define the CM limits of the program that is initiated by the CYBER 170 exchange jump. RAE and FLE are loaded to define the UEM limits of a program.

Figure 17-6 shows the absolute and relative memory addresses, RAC, FLC, RAE, and FLE register relationships. For a program to operate within the established limits, the following conditions must exist.

- For absolute memory addresses: $RAC \leq (RAC + P) < (RAC + FLC)$
- For relative memory addresses: $0 \leq P < FLC$

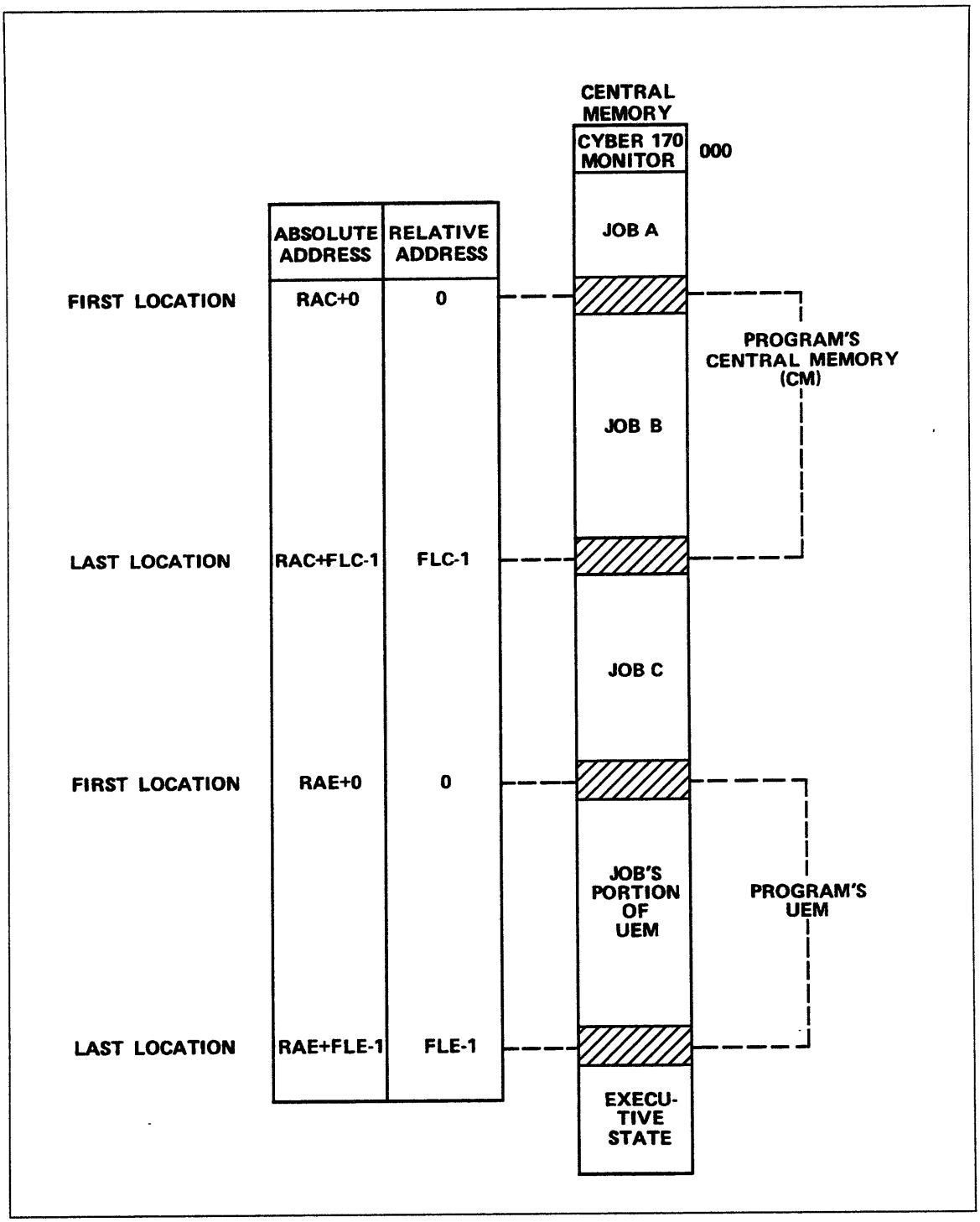


Figure 17-6. Memory Map

Addressing Modes

UEM can be used in either of two addressing modes: standard or expanded. Standard addressing mode provides addressing up to 21 bits in a 24-bit format. Expanded addressing mode provides addressing up to 24 bits in a 30-bit format. Addressing mode is determined by the expanded addressing select flag, bit 55 of word 3, in the CYBER 170 exchange package.

Direct Read/Write Instructions (014, 015, 660, 670)

These instructions transfer one 60-bit word between the selected X register and a memory location, using a 21-bit relative address. Instructions 660 and 670 use the memory address Xk (21 bits) plus RAC (21 bits) to address CM. Instructions 014 and 015 use the memory address Xk (21 bits) plus RAE (21 bits) to address UEM.

Block Copy Instructions (011, 012)

These instructions transfer up to 131 071 60-bit words between fields in CM and UEM. The UEM address is X0 plus RAE (bits 0 through 22 in standard addressing mode; bits 0 through 28 in expanded addressing mode). The CM address is A0 plus RAC (if the block copy flag is clear in the CYBER 170 exchange package) or X0 (bits 30 through 50) plus RAC (if the block copy flag is set).

The transfers occur in blocks of up to 64 words, during which other CP activities are suspended.

These instructions are 30-bit instructions that must start at parcel 0. If the UEM address has bit 21 or bit 22 set in standard addressing mode (bit 28 if in expanded addressing mode), zeros are transferred to CM and the next instruction is taken from parcel 2 of the same instruction word. If this is not the case on a block read, the next instruction is taken from parcel 0 of the next instruction word. A transfer of all zeros can be made to central memory using the 011 instruction and setting bit 21 or 22 (or bit 28) of the address (X0 + RAE) when FLE is sufficiently large.

PP Programming

The PPs have access to all CM storage locations. One 64-bit word or a block of 64-bit words can be transferred from a peripheral processor memory (PPM) to CM or from CM to PPM. (Five 12-bit PP words equal one 64-bit CM word, with the leftmost 4 bits undefined.) Data from external devices is read into a PPM, and with additional instructions, is transferred to CM. Conversely, data is transferred from CM to a PPM and is then transferred by additional instructions to external devices. Addresses sent to CM from PPs are absolute or relocation addresses.

Central Memory Addressing by PPs

PPs address central memory using either absolute or relocation addressing. Every PP can read all central memory locations without restriction. Every PP has write access to central memory. The bounds register in central memory may also be set to limit write access from the IOU.

Instructions 24/25 load/store the relocation (R) register. If bit 17 of the A register is zero, bits 0 through 16 of A specify an absolute central memory address 0 through 377777_8. If bit 17 of A is one, bits 0 through 16 of A are added to the 28-bit R register to specify an absolute central memory address 0 through 0 007777777_8. If bit 17 of A changes during a transfer, the address ing mode also changes accordingly. The leftmost 7 bits of R represent unused extra addressing capacity. The rightmost 6 bits of R are appended zeros. Instruction 24 loads R from two consecutive PP memory locations. Instruction 25 stores R into two PP memory locations. Figure 4-4 shows how R is stored in PP memory.

PP Memory Addressing by PPs

PP instructions use 6-bit or 18-bit direct operands or access PP memory through direct, indirect, or indexed addressing.

Direct 6-Bit Operand

PP instructions in this category are no-address instructions. They have the format OPCODEd. The d field is used as a 6-bit direct operand, zero-extended to 18 bits in calculations.

Direct 18-Bit Operand

PP instructions in this category are constant address instructions. They have the format OPCODEdm. The combined d and m fields are used as an 18-bit operand.

Direct 6-Bit Address

PP instructions in this category are direct-address instructions. They have the format OPCODEd. The d field is used as a 6-bit direct address, accessing PP memory locations 0 to 77_8.

Direct 12-Bit Address

PP instructions in this category are indexed direct-address instructions with zero index. They have the format OPCODEdm where d equals zero. The m field is used as a 12-bit direct address that accesses PP memory locations 0 through 7777_8.

Indexed 12-Bit Address

PP instructions in this category are indexed direct-address instructions. They have the format OPCODEdm where d equals zero. The m field is used as a 12-bit direct address (base address). The d field specifies a PP memory location from 1 to 77_8; the contents of which is a 12-bit ones complement number index. The indexed direct address is formed by adding the index to the base address as signed ones complement numbers. Overflow is ignored. When m plus (d) equals 7777, the result is set to 0000, except as follows: adding 7777 plus 7777 equals 7777. In general, adding 0000 or 7777 leaves the other number unchanged, except when the other number is also 0000 or 7777.

Indirect 6-Bit Address

PP instructions in this category are indirect-address instructions. They have the format OPCODEd. The 6-bit d field is used to read a 12-bit number from PP locations 0 through 77_8. This number is used as a 12-bit address to access PP memory locations 0 through 7777_8.

Central Memory Read/Write Instructions

PP instructions can read and write to central memory either single words or blocks of words.

PP Central Memory Read Instructions (60, 61)

Instruction 60 transfers one CM word into five 12-bit PP memory words. Instruction 61 transfers a block of 1 through 811 CM words into 5 through 4095 12-bit PP words. It is possible to transfer up to 4096 CM words overwriting PP memory cyclically; location 0, however, has special properties. The Central Read description in chapter 16 has more information on instruction 61.

PP Central Memory Write Instructions (62, 63)

Instruction 62 transfers five 12-bit PP memory words into one CM word. Instruction 63 transfers 5 through 4095 PP memory words into 1 through 811 CM words. It is possible to transfer up to 20 480 PP memory words, repeating information from PP memory cyclically.

Input/Output Channel Communications

Data transfers to and from external devices are controlled by PP instructions 64 through 77. The assignment of PPs, transfer priorities, and resolution of conflicts are software responsibilities.

Channel parity and reservation must be provided for, using the channel marker flag and/or software interlocks in central memory. After any conflicts have been resolved, proceed as follows:

Action		Typical Instruction
1.	Clear the error flag.	Jump if the error flag is set, and clear the flag (661).
2.	Verify inactive status.	Jump if active (640).
3.	Verify read status:	
	Prepare for reading the summary status.	Function m (77).
	Verify that the device responded.	Jump if active (640).
	Activate the channel.	Activate (74).
	Read the summary status.	Input to A (70).
	Verify the error flag is clear.	Jump if the error flag is set (661).
	Analyze the summary status.	Logical product (12). Zero jump (04).
4.	Enter the number of words to A.	Load d (14).
5.	Prepare for input/output:	
	Verify inactive status.	Jump if active (640).
	Prepare for read/write.	Function m (77).
	Verify that the device responded.	Jump if active (640).

Action		Typical Instruction
6.	Read/write data:	
	Activate the channel.	Activate (74).
	Read/write data.	Input/output A words (71/73).
	If write, loop until empty.	Jump if full (660).
	Disconnect the channel.	Deactivate (75).
	Verify inactive status.	Jump if active (640).
7.	Verify transfer integrity:	
	Verify A words were transferred (refer to note).	Nonzero jump (05).
	Verify the error flag is clear.	Jump if error flag set (661).
	Verify inactive status.	Jump if active (640).
	Prepare for reading device status.	Function m (77).
	Verify that the device responded.	Jump if active (640).
	Activate the channel.	Activate (74).
	Read the device status.	Input to A (70).
	Verify the error flag is clear.	Jump if error flag set (661).
	Analyze device status.	Logical product (12). Nonzero jump (05).
	Disconnect the channel.	Deactivate (75).

NOTE

If A equals the original value, no words were transferred.

If A is not equal to zero, the device or another PP ended the transfer.

Inter-PP Communications

Any PP can communicate with any other PP using any channel (except the real-time clock) by omitting the conditioning of the external devices of that channel for a data transfer. Both single-word and block transfers can be used. Either the sending or the receiving PP can activate the channel used, after which the sending PP outputs data into the channel register of the channel concerned and the receiving PP inputs data from the same register. The transfer rate is one word every 250 ns, except when the transfer is between PPs in different barrels but in the same time slot. In such a case, the transfer rate is one word every 500 ns. PPs that use the same time slots are as follows:

Slot	PP Number
1	0, 5, 20, 25
2	1, 6, 21, 26
3	2, 7, 22, 27
4	3, 10, 23, 30
5	4, 11, 24, 31

Software resolves priority and reservation problems arising in inter-PP communications by interlocks stored in CM or by other means.

PP Program Timing Considerations

Some external equipment may require timing considerations in issuing function, activate, and input instructions. Refer to the applicable external equipment reference manual. Such timing considerations may, for example, be required to ensure that the equipment attains a proper speed before data is sent (required by some magnetic tape equipment). Also, equipment that terminates a data transfer by resetting the active flag to inactive often requires timing considerations in issuing the next function instruction.

Channel Operation

Channel Control Flags

Channel operation is affected by the channel active/inactive and full/empty flags and, depending on the status of these two flags, the channel is said to be active, inactive, full, or empty. Each channel also has a marker flag for software use and an error flag for indicating transmission parity errors.

Channel Active/Inactive Flag

A channel is normally activated by a function (76 or 77) instruction or by an activate channel (74) instruction. An external device can also activate the channel.

A function instruction conditions the external device for a coming data or status information transfer. The instruction places a 12-bit function word plus parity in the channel register and sets the active and full flags. The function word and a function signal are sent to the external device. No active or full signals are sent during a function instruction. The external device accepts the function word and sends an inactive signal, which clears the channel active and full flags, clearing the channel register.

An activate channel instruction prepares a channel for data transfer and sends an active signal to the external device. Subsequent input or output instructions transfer data. A disconnect channel (75) instruction after a data transfer returns the channel to an inactive state, and an inactive signal is sent to the external device.

Register Full/Empty Flag

A register is full when it contains a function or data word for an external device or contains a word received from the external device. The register is empty when the flag clears. The flag is turned on or off as the register changes state. A channel can only be full when it is active.

On data output, the processor places a word in the channel register (the channel should be active and empty) and sets a full flag. The data word plus parity and a full signal are sent to the external device. The external device accepts the word and sends an empty signal to the channel, which clears the full flag, clearing the channel register. The active and empty status of the channel signals the PP to send the next word to the register.

On data input, the external device sends a word and a full signal to the data channel. The word is placed in the channel register, and the full flag sets. The PP stores the word and clears the full flag, clearing the data register. An empty signal is sent to the external device, signaling it to send the next data word.

Channel (Marker) Flag Instructions (641, 651)

Software uses this flag software as a marker. This flag does not affect hardware operation. When PPs in the same time slot use this flag, priority conflicts exist. For channel 17_8 (maintenance channel) marker flag, hardware resolves priority problems. For other channels, software must resolve such conflicts. Any five consecutively numbered PPs are not in the same time slot.

Error Flag Instructions (661, 671)

This flag indicates an input data parity error on the specific channel being tested. The flag also indicates an output data parity error on channels that have the capability of sending an error signal to the IOU in case of such an error. The status register of the device concerned must be read to verify output data integrity.

Channel Transfer Timing

Figure 17-7 shows channel transfer timing. All signal pulses are 25 $\pm$5 ns in width and occur 25 $\pm$5 ns following the 10-MHz clock.

To maintain the fastest possible cycle time (500 ns), a function/full/empty pulse from the PP must be answered with an inactive/empty/full pulse, respectively, within 310 $\pm$35 ns. If the maximum speed is not required, this response time may be increased by multiples of 100 ns.

The PP master clock frequency can be varied by $\pm$2 percent. The peripheral devices used must tolerate this frequency variation.

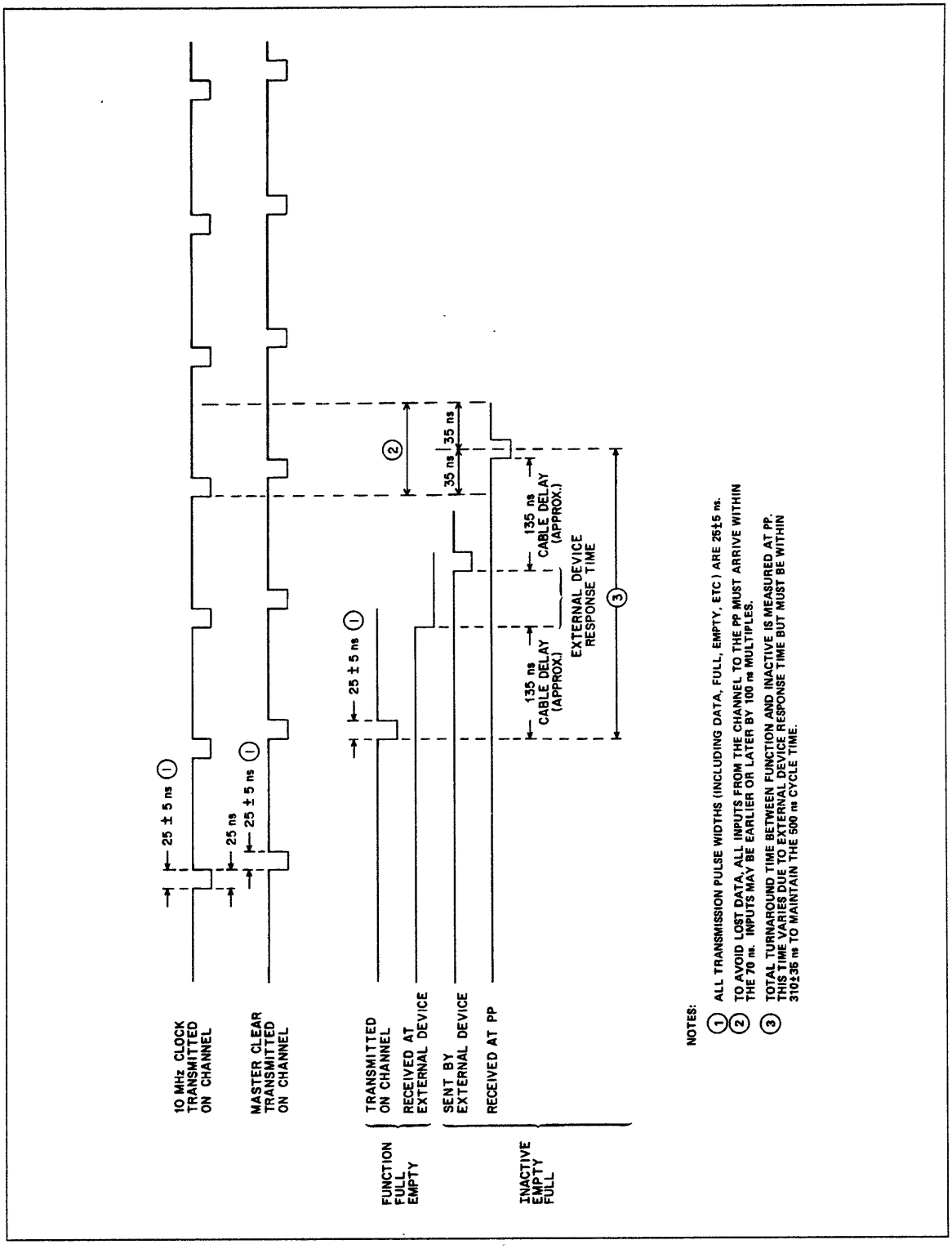


Figure 17-7. Channel Transfer Timing

Input/Output Transfers

The following paragraphs discuss input/output transfers with the PP.

Data Input Sequence

The external device sends data (figure 17-8) to the PP via the controller as follows:

1. The PP places a function word in the channel register and sets the full flag and the channel active flag. At the same time, the PP sends the first of a group of words and function signals to all controllers. The function signals cause all controllers to sample the words and identify the words as function codes rather than data words. Connect codes select controllers and modes of operation and clear nonselected controllers. Only selected controllers are connected.

2. The controller sends an inactive signal to the PP, indicating acceptance of the function code. The signal drops the channel active flag, which in turn drops the full flag and clears the channel register.

3. The PP sets the channel active flag and sends an active signal to the controller, which signals the input equipment to start sending data.

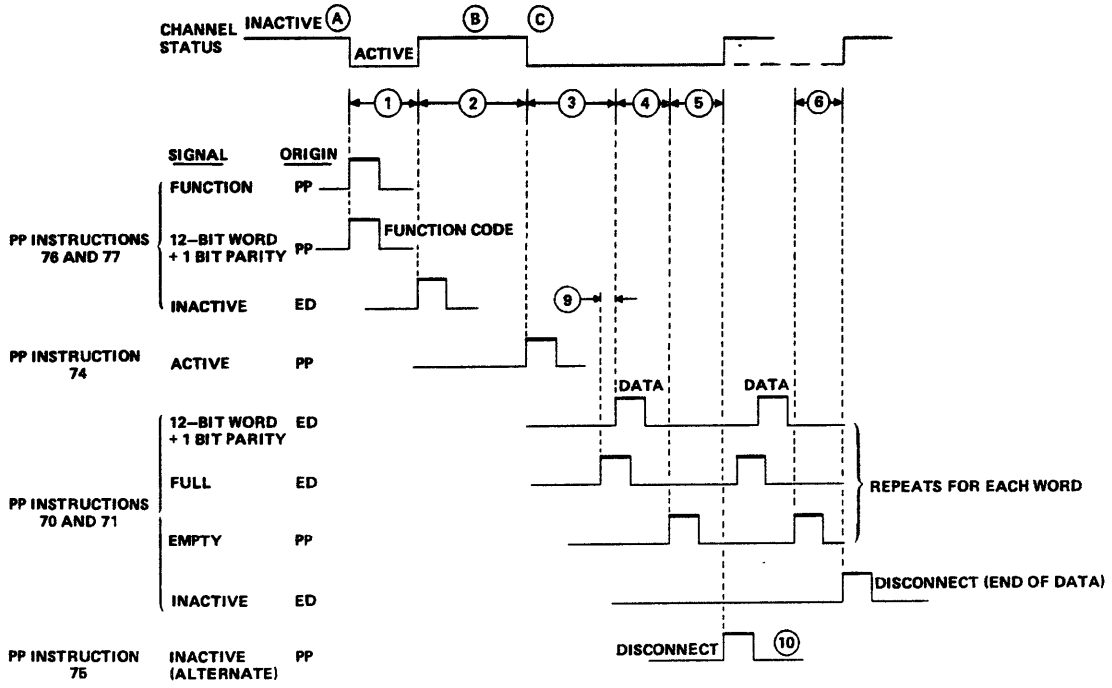
4. The input equipment reads a 12-bit data word plus one parity bit and then sends the word with parity to the channel register with a full signal, which sets the channel full flag (10 to 15 nanoseconds after the data arrives).

5. The PP stores the word, drops the full flag, and returns an empty signal, indicating acceptance of the word. The input equipment clears its data register and prepares to send the next word.

6. Steps 4 and 5 repeat for each word transferred.

7. At the end of the transfer, the controller clears its active condition and sends an inactive signal to the PP to indicate the end of the data. The signal clears the channel active flag to disconnect the controller and the PP from the channel.

8. As an alternative, the PP may choose to disconnect from the channel before the input equipment has sent all its data. The PP does this by dropping the active flag and sending an inactive signal to the controller, which immediately clears its active condition and sends no more data, although the input equipment may continue to the end of its record or cycle (for example, a magnetic tape unit would continue to end-of-record and stop in the record gap).



NOTES:

1. TIME IS A FUNCTION OF EXTERNAL DEVICE (ED). PP RECOGNIZES INACTIVE 1 MAJOR CYCLE (OR A MULTIPLE OF MAJOR CYCLES) AFTER FUNCTION. THE PP MUST PREVIOUSLY RECEIVE INACTIVE.
2. TIME IS A FUNCTION OF PERIPHERAL PROCESSOR (PP). MINIMUM TIME IS 1 MINOR CYCLE. ACTUAL TIME IS A FUNCTION OF THE PP PROGRAM.
3. TIME IS A FUNCTION OF ED.
4. TIME IS A FUNCTION OF PP. MINIMUM TIME IS 1 MINOR CYCLE. MAXIMUM TIME IS UP TO 4 MINOR CYCLES TO ALLOW OPERATION WITHIN 1 MAJOR CYCLE.
5. TIME IS A FUNCTION OF PP. MINIMUM TIME IS 2 MAJOR CYCLES. MAXIMUM TIME IS AN INTEGRAL MULTIPLE OF MAJOR CYCLES.
6. TIME IS A FUNCTION OF ED.
7. MAJOR CYCLE TIME IS 250 NS.
8. MINOR CYCLE IS 50 NS.
9. TIME IS A FUNCTION OF ED. FULL SHOULD PROCEED THE DATA BY A MINIMUM OF 5 NS (15 NS MAXIMUM) TO REMOVE THE CLEAR ON THE INPUT DATA RECEIVERS.
10. PP MAY DISCONNECT AFTER EMPTY SIGNAL OF ANY ED WORD. STATUS REQUEST DISCONNECTS IN THIS MANNER.

A. CHANNEL MUST BE PREVIOUSLY INACTIVE.

B. CHANNEL REMAINS ACTIVE UNTIL ED SENDS INACTIVE.

C. CHANNEL MUST BE PREVIOUSLY INACTIVE.

Figure 17-8. Data Input Sequence Timing

Data Output Sequence

The PP sends data (figure 17-9) to the external device as follows.

1. The PP places a function word in the channel register and sets the full flag and the channel active flag. The function signal causes all controllers to sample the word and identify the word as a function code rather than a data word. Connect codes select controllers and modes of operation and clear nonselected controllers. Only selected controllers are connected.

2. The controller sends an inactive signal to the PP, indicating acceptance of the function code. The signal drops the channel active flag, which in turn, drops the full flag and clears the channel register.

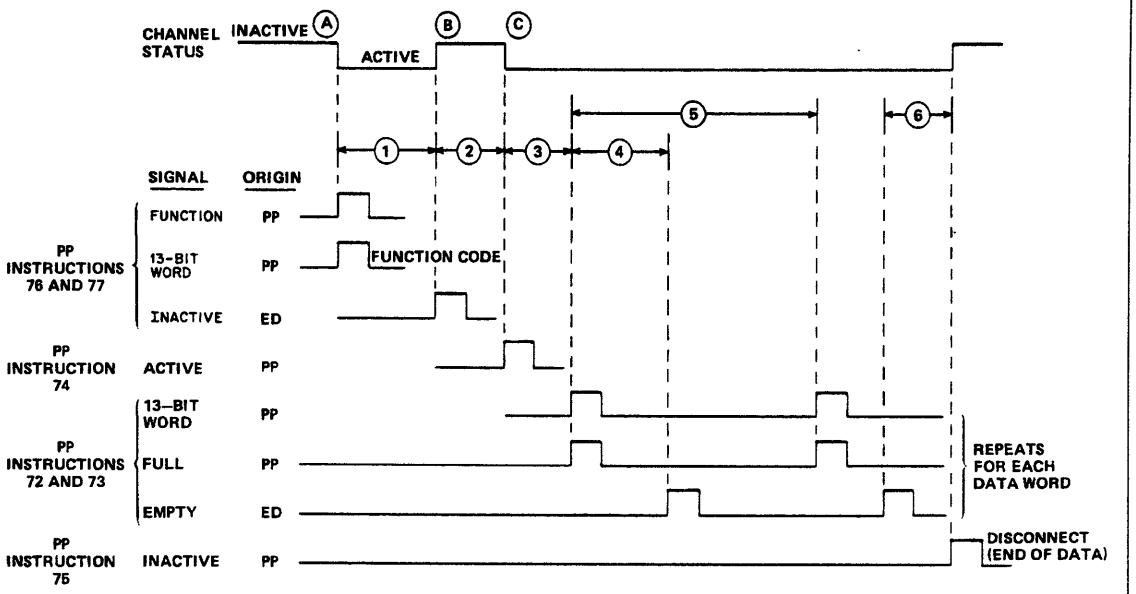
3. The PP sets the channel active flag and sends an active signal to the controller, which signals the output equipment that data flow is starting.

4. The PP places a 12-bit data word plus one parity bit in the channel register and sets the full flag. Coincidently, the PP sends a word with parity and a full signal to the controller.

5. The controller accepts the word and sends an empty signal to the PP where the signal clears the channel register and drops the full flag.

6. Steps 4 and 5 repeat for each PP word.

7. After the last word is transferred and acknowledged by the controller empty signal, the PP drops the channel active flag and turns off the controller with an inactive signal.



NOTES

1. TIME IS A FUNCTION OF EXTERNAL DEVICE (ED). PP RECOGNIZES INACTIVE 1 MAJOR CYCLE (OR A MULTIPLE OF MAJOR CYCLES) AFTER FUNCTION. THE PP MUST PREVIOUSLY RECEIVE INACTIVE.
2. TIME IS A FUNCTION OF PERIPHERAL PROCESSOR (PP) MINIMUM TIME IS 1 MINOR CYCLE ACTUAL TIME IS A FUNCTION OF THE PP PROGRAM
3. TIME IS A FUNCTION OF ED.
4. TIME IS A FUNCTION OF PP MINIMUM TIME IS 1 MINOR CYCLE. MAXIMUM TIME IS UP TO 4 MINOR CYCLES TO ALLOW OPERATION WITHIN 1 MAJOR CYCLE
5. TIME IS A FUNCTION OF PP. MINIMUM TIME IS 2 MAJOR CYCLES MAXIMUM TIME IS AN INTEGRAL MULTIPLE OF MAJOR CYCLES.
6. TIME IS A FUNCTION OF ED
7. MAJOR CYCLE TIME IS 250 NS.
8. MINOR CYCLE IS 50 NS.

A. CHANNEL MUST BE PREVIOUSLY INACTIVE.
B. CHANNEL REMAINS ACTIVE UNTIL ED SENDS INACTIVE.
C. CHANNEL MUST BE PREVIOUSLY INACTIVE.

Figure 17-9. Data Output Sequence Timing

System Console Programming

Keyboard

A PP transmits function code 7020_8 to request data from the keyboard of the system console. The PP then activates the input channel and inputs one character from the keyboard. This character enters as the lower 6 bits of the word; the upper bits are cleared. There is no status report by the keyboard. Table 17-9 lists the keyboard character codes.

Data Display

Data is displayed within an 8- by 11-inch area of a cathode-ray tube (CRT). The display can be in character mode (alphanumeric) and/or dot mode (graphic). Two presentation areas (left and right) are displayed. Each is made up of 262 144 dot locations arranged in a 512- by 512-dot format. Each dot position is determined by the intersection of X and Y coordinates. The lower left corner dot is octal address X=6000 and Y=7000, and the upper right corner dot is octal address X=6777 and Y=7777. An optional CC634-B or CC598-B system console is available. The CC598-B emulates the CC634-B console. Refer to the hardware reference manual for the CC634-B listed in About This Manual for additional information regarding these optional terminals.

Character Mode

In character mode, three sizes are provided. Large characters are arranged in a 32- by 32-dot format with 16 characters per line. Medium characters are arranged in a 16- by 16-dot format with 32 characters per line. Small characters are arranged in an 8- by 8-dot format with 64 characters per line. Table 17-10 lists the display character codes.

Table 17-9. Keyboard Character Codes

Character	Code	Character	Code
No data	00	0	33
A	01	1	34
B	02	2	35
C	03	3	36
D	04	4	37
E	05	5	40
F	06	6	41
G	07	7	42
H	10	8	43
I	11	9	44
J	12	+	45
K	13	-	46
L	14	*	47
M	15	/	50
N	16	(	51
O	17	)	52
P	20	Left blank key	53
Q	21	=	54
R	22	Right blank key	55
S	23	,	56
T	24	.	57
U	25	Carriage return	60
V	26	Backspace	61
W	27	Space	62
X	30		
Y	31		
Z	32		

Table 17-10. Display Character Codes

Character	Code	Character	Code
A	01	1	34
B	02	2	35
C	03	3	36
D	04	4	37
E	05	5	40
F	06	6	41
G	07	7	42
H	10	8	43
I	11	9	44
J	12	+	45
K	13	-	46
L	14	*	47
M	15	/	50
N	16	(	51
O	17	)	52
P	20	Space	53
Q	21	=	54
R	22	Space	55
S	23	,	56
T	24	.	57
U	25		
V	26		
W	27		
X	30		
Y	31		
Z	32		

Dot Mode

In dot mode, display dots are positioned by the X and Y coordinates. The X coordinates position the dots horizontally. The Y coordinates position the dots vertically and unblank the CRT for each dot. A series of X and Y coordinates form horizontal lines. A single X coordinate and a series of Y coordinates form vertical lines.

Codes

A single function word is transmitted to select the presentation, mode, and character size (character mode only). Figure 17-10 illustrates the function word format. The word following the function word specifies the starting coordinates for the display (for either mode). Figure 17-11 illustrates the coordinate data word. In character mode, the words that follow are display character codes. Figure 17-12 illustrates the character data word.

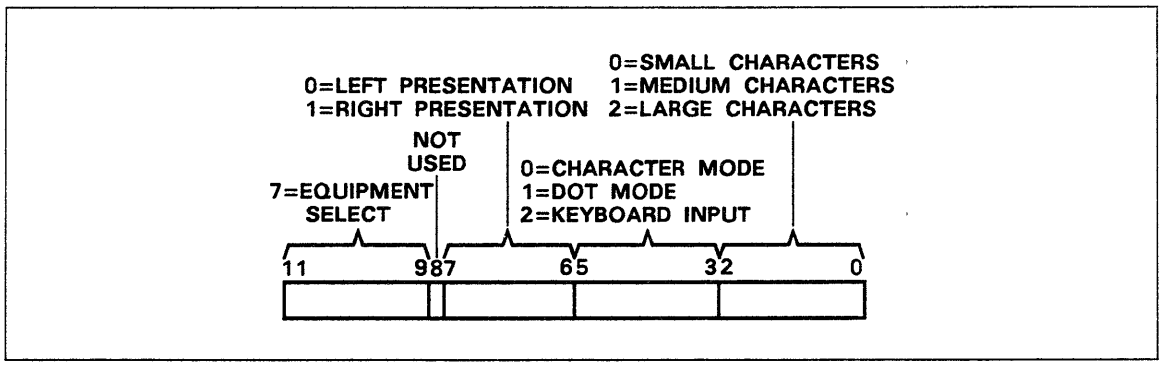


Figure 17-10. Display Station Output Function Code

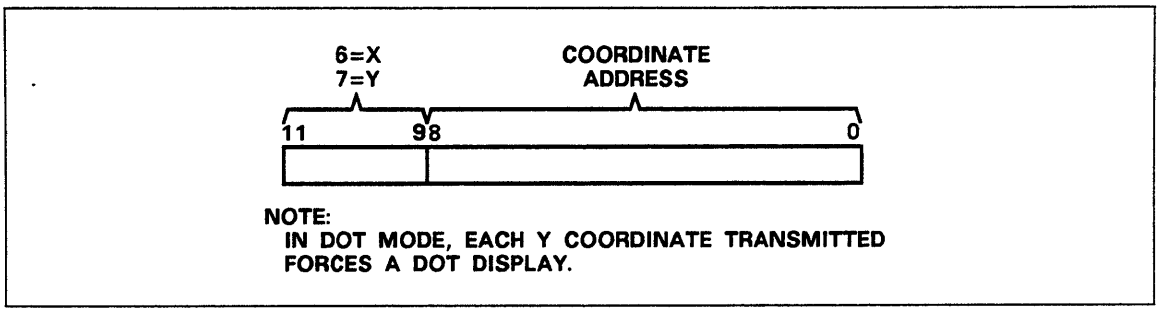


Figure 17-11. Coordinate Data Word

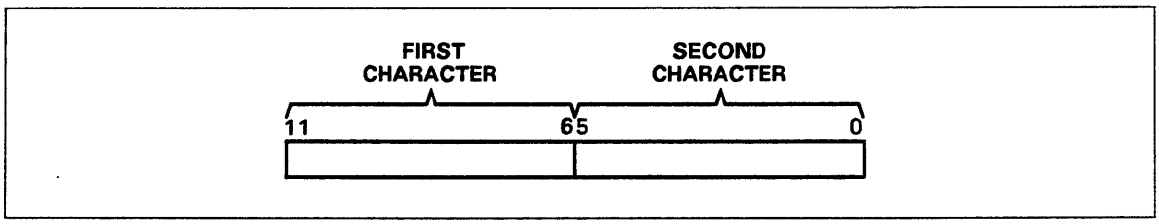


Figure 17-12. Character Data Word

When the display operation has started, the controller regulates character spacing on the line. A new coordinate data word must be sent to start each line. If new coordinates are not specified, data is written on the line specified by the active coordinate word, and information already on that line is overwritten. Character sizes can be mixed by sending a new function word and coordinate word for each size change. Spacing on a line can be varied by sending a coordinate word for the character that is to be spaced differently.

Programming Example

The following programming example (figure 17-13) requests an input of one line of data from the system console and displays this data on the CRT as it is being typed.

Programming Timing Considerations

When performing an output operation, the computer must wait at the end of the output for a channel-empty condition to prevent a loss of coordinates or data. A full jump at the end of the output ensures that the channel is empty and the display controller accepts the last word of the output before disconnecting from the channel.

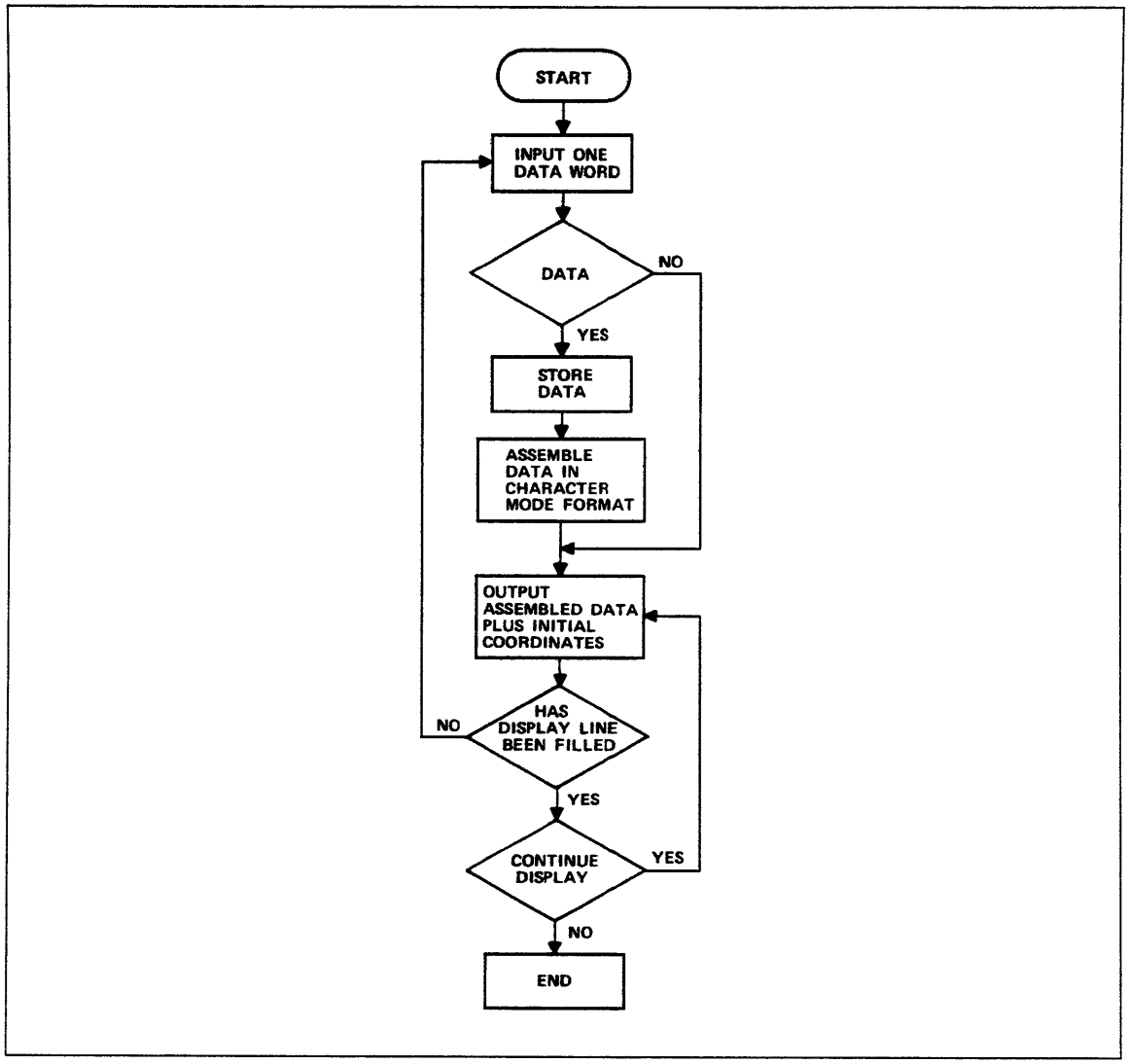


Figure 17-13. Receive and Display Program Flowchart

Real-Time Clock Programming

Channel 14_8 is reserved for the real-time clock. This channel, which is always active and full, may be read at any time. The real-time clock is a 12-bit, free-running counter incrementing at a 1-MHz rate from 0 through 4095_{10}.

Two-Port Multiplexer Programming

NOTE

For two-port multiplexer programming, bit numbering within words is 0 through 63 from left to right.

Channel 15_8 is reserved for communications with one or two external devices through the two-port multiplexer. One port is reserved for maintenance purposes, and the other is reserved for future use. The two-port multiplexer can communicate with all external devices that use EIA standard RS232C serial interface. The multiplexer can accommodate data with odd/even parity, 5 to 8 bits per character and 1 or 2 stop bits. Issuing appropriate function codes sets the format. The rate is switch selectable for each channel for operation between 110 and 9600 baud. These switches are located internally on the two-port multiplexer.

Two-Port Multiplexer Operation

The two-port multiplexer uses the rightmost 12 bits on channel 15_8. A 12-bit (octal) function word from the PP is translated to specify the following operating conditions.

Code	Function
7XXX	Terminal select.
6XXX	Terminal deselect.
00XX	Read status summary.
01XX	Read terminal data.
02XX	Write output buffer.
03XX	Set operation mode to terminal.
04XX	Set/clear terminal control signal, data terminal ready (DTR).
05XX	Set/clear terminal control signal, request to send (RTS).
06XX	Not used.
07XX	Master clear selected port.

Terminal Select (7XXX)

The PP sends this select code to specify the terminal to which the function codes and data transmissions apply. Code 7000 selects port 0 (for future use), and code 7001 selects port 1 (maintenance console).

Terminal Deselect (6XXX)

The PP sends this code, which deselects the two-port multiplexer from channel 15_8 so the 16-bit channel is available for inter-PP communications.

Read Status Summary (00XX)

This code permits the PP to input status from the selected terminal. A one-word input must follow to read the status response. The response is 12 bits, which are defined as follows.

Bit	Status
52-58	Not used.
59	Output buffer not full.
60	Input ready.
61	Data carrier detect or carrier on.
62	Data set ready.
63	Ring indication.

PP Read Terminal Data (01XX)

This code permits the PP to input the terminal data from the selected terminal. Channel 15_8 must be activated, and a one-word input must follow to read in the terminal data. The data word is 12 bits, which are defined as follows.

Bit	Status
52	Data set ready.
53	Data set ready and data carrier detector.
54	Over run.
55	Framing or parity error.
56-63	8-bit data.

Data Set Ready (Bit 52)

When the data set ready signal is active, this bit sets.

Data Set Ready (DSR) and Data Carrier Detector (DCD) (Bit 53)

When both data set ready and data carrier detector signals are active, this bit sets.

Over Run (Bit 54)

When the previously received character is not read by the PP before the present character is transferred to the data holding register, the overrun bit sets.

Framing or Parity Error (Bit 55)

When the received character does not have a valid stop bit (framing error) or when this bit sets, the received character parity does not agree with the select parity (parity error).

Data Character (Bits 56 Through 63)

The lower 8 bits of the input word form the data character. The multiplexer forms this character directly from the Universal Asynchronous Receiver and Transmitter (UART).

PP Write Output Buffer (02XX)

This code prepares the multiplexer for an output operation to the 64-character output buffer memory. Before an output operation can proceed, channel 15_8 must be activated. The output operation is terminated when the multiplexer receives an inactive signal from the PP or when no more locations are available in the output buffer. In the latter case, an inactive (instead of empty) signal is sent back to the channel, which in turn, terminates the output operations.

Set Operation Mode to the Terminal (03XX)

This code permits the PP to set the terminal operation mode register. A 12-bit function code word from the PP specifies the operation of the terminal. This word is decoded in the function register. Segments of the word define the mode as follows:

Bit	Status
58	Not used.
59	No parity. When this bit is set, it eliminates the parity bit from the transmitted and received characters. The stop bit(s) immediately follow the last data bit.
60	Number of stop bits. This bit selects the number of stop bits, 1 or 2, to be appended immediately after the parity bit. When this bit is clear, it inserts 1 stop bit; and when set, it inserts 2 stop bits.
61-62	Number of bits per character. These 2 bits are internally decoded to select 5, 6, 7, or 8 data bits per character.

Bit 61	Bit 62	Bits Per Character
0	0	5
0	1	6
1	0	7
1	1	8

63	Odd/even parity select. This bit selects the type of parity that will be appended immediately after the data bits. It also determines the parity that will be checked on read data.

Set/Clear Data Terminal Ready (04XX)

This code permits the PP to set or clear the terminal control signal, data terminal ready (DTR). When bit 63 is set, DTR is active; and when bit 63 is clear, DTR is inactive.

Set/Clear Request to Send (05XX)

This code permits the PP to set or clear the terminal control signal, request to send (RTS). When bit 63 is set, RTS is active; and when bit 63 is clear, RTS is inactive.

Master Clear (07XX)

This code permits the PP to master clear the selected port including its output buffer memory and UART. The terminal operation mode register and terminal control signals are not cleared.

Programming Considerations

Channel 15_8 communicates with the terminals connected to the external interface, one at a time. To establish communications between a PP and the terminal, the PP issues a function for select. The function word for select is formed by the least-significant 12 bits, which are sent to channel 15_8, and specifies the following information.

- A select code to select the multiplexer (7XXX).
- The terminal with which the PP would like to establish communication (7XXX).

When the connect is established, the two-port multiplexer routes all data to the terminal designated by the select code. The multiplexer responds with the inactive signal to acknowledge the receipt of the function code of 7XXX for select, 6XXX for deselect, and 0XXX for operation. Otherwise, the multiplexer ignores the function.

Output Data

The multiplexer accepts a maximum data block length of 64 characters per terminal. During the block data transfer, the multiplexer terminates the output operation either when it receives an inactive signal from the channel or when the output buffer is full. When the output buffer is full, the multiplexer sends back an inactive signal instead of an empty signal to the channel on the last output word. The signal indicates the output buffer accepts the last output word and it cannot receive anymore data from the PP. The multiplexer does not allow output to a full buffer. The multiplexer sends back an inactive signal to deactivate channel 15_8 after the multiplexer decodes the previous function code, which is 02XX (PP write output buffer), and receives an activate signal from the PP.

Input Data

The multiplexer does not store the input data from the terminal. A lost data condition exists if the PP does not input the previous data before the new data arrives from the terminal. The multiplexer allows input from an empty input buffer.

Request to Send and Data Terminal Ready

The hardware brings up request to send and data terminal ready automatically under the following conditions regardless of the software RTS and DTR bits.

- Data in the UART output register.
- Data in the FIFO output register.

When no data is in the FIFO or UART, the software bit determines RTS and DTR.

Maintenance Channel Programming

NOTE

Maintenance registers are numbered 0 through 63 from left to right.

Maintenance Channel

A PP in the IOU can perform any or all of the following operations through the maintenance channel (MCH) to each system element, such as the CP, IOU, and CM.

- Initializing registers, controls, and memories.
- Monitoring and recording error information.
- Verifying error-detection and correction hardware.

The maintenance channel consists of the maintenance channel interface on channel 17_8, a maintenance channel interface in each system element, and a set of interconnecting cables.

The IOU maintenance channel interface contains a selector that connects to one of up to seven system elements. The IOU is element 0, and its maintenance access control is internally connected to the selector. All other system elements are assigned arbitrary element numbers. A single cable connects each maintenance access control to the selector. This arrangement results in a radial connection that allows any system element to be shut down or removed without affecting communication with the other elements.

MCH Function Words

The MCH function word consists of the connect, opcode, and type fields, which are used as described in the next three paragraphs and tables 17-11, 17-12, and 17-13.

The connect field specifies the unit to which the MCH is connected (CP, CM, or IOU) controlling selection within the IOU only. The unit remains connected until another connect code selects a different unit. Connect codes 10_8 to 17_8 leave the MCH unconnected; in this state, the interface can be used for PP to PP communications.

The OPCODE field controls the unit selected by the connect code, preparing the unit for a coming read/write/echo operation or causing the unit to halt, start, clear, or deadstart.

The use of the TYPE field depends on the connected unit. When the CP is the connected unit, type codes 1 through A_{16} (model 835) or 1 through 7 (models 840, 845, 850, 855, and 860) specify the data type in the operation to be performed. Also, for the CP, type code 0 specifies that the internal address of the CP register to be connected is specified in a control word which is sent as two data words immediately following the function word. When IOU is the connected unit, type codes 0 through 7 specify the starting byte number for read/write operations (all models except 990 and CYBER 990E, 995E, and 994). For model 990 and CYBER 990E, 995E, and 994, the TYPE field must be set to all zeros. The exceptions are reading the options installed and element identifier registers. On the model 835, CM ignores the type code. On models 840, 845, 850, 855, and 860, CM uses A_{16} to access the maintenance registers.

Table 17-11. MCH Function Word Bit Assignments (CP and CM)

Field				Description
MCH Function Word to Model 835 CP				
CONNECT (bits 8-11)	Code	2_{16}	=	Connect CP maintenance registers
OPCODE (bits 4-7)	Code	0_{16}	=	Halt processor
		1_{16}	=	Start processor
		4_{16}	=	Prepare for read (control word required)
		5_{16}	=	Prepare for write (control word required)
		6_{16}	=	Master clear
		7_{16}	=	Clear errors
TYPE (bits 0-3)	Code	0_{16}	=	Control word required
MCH Function Word to Model 835 CM				
CONNECT (bits 8-11)	Code	1_{16}	=	Connect CM maintenance registers
OPCODE (bits 4-7)	Code	4_{16}	=	Prepare for read (control word required)
		5_{16}	=	Prepare for write (control word required)
		6_{16}	=	Master clear
		7_{16}	=	Clear fault status register
MCH Function Word to Models 840, 845, 850, 855, 860, and 990 and CYBER 990E, 995E, and 994 CP and CM				
CONNECT (bits 8-11)	Code	1_{16}	=	Required for models 845 and 855 CP and CM
TYPE (bits 0-3)	Code	0_{16}	=	CP and CP registers
OPCODE (bits 4-7)	Code	0_{16}	=	Halt processor
		1_{16}	=	Start processor
		4_{16}	=	Prepare for read
		5_{16}	=	Prepare for write
		6_{16}	=	Master clear
		7_{16}	=	Clear errors
TYPE (bits 0-3)	Code	1_{16}	=	Control store memory
OPCODE (bits 4-7)	Code	4_{16}	=	Prepare for read
		5_{16}	=	Prepare for write

(Continued)

Table 17-11. MCH Function Word Bit Assignments (CP and CM) *(Continued)*

TYPE (bits 0-3)	Code	$3\text{-}7_{16}$	=	Internal memories (all models except 990 and CYBER 990E and 995E)
		3_{16}	=	Unused (Model 990 and CYBER 990E, 995E, and 994)
		4_{16}	=	ACU Control Memories (Model 990 and CYBER 990E, 995E, and 994)
		5_{16}	=	BP3 Decode Memories (Model 990 and CYBER 990E, 995E, and 994)
		6_{16}	=	Operand Cache (OCA) (Model 990 and CYBER 990E, 995E, and 994)
		7_{16}	=	Register File (RGU) (Model 990 and CYBER 990E, 995E, and 994)
		8_{16}	=	Load and Store Section (LSU) Control Memories (Model 990 and CYBER 990E, 995E, and 994)
		9_{16}	=	Error Processing Network (EPN) (Model 990 and CYBER 990E, 995E, and 994)
OPCODE (bits 4-7)	Code	4_{16}	=	Prepare for read
		5_{16}	=	Prepare to write
TYPE (bits 0-3)	Code	A_{16}	=	CM and CM Registers
OPCODE (bits 4-7)	Code	4_{16}	=	Prepare for read
		5_{16}	=	Prepare for write
		6_{16}	=	Master clear
		7_{16}	=	Clear errors

Table 17-12. MCH Function Word Bit Assignments, All Models Except 990 and CYBER 990E, 995E, and 994 (IOU)

Field				Description (MCH Function Word to IOU)
CONNECT (bits 8-11)	Code	0_{16}	=	Connect IOU maintenance registers
OPCODE (bits 4-7)	Code	4_{16}	=	Prepare for read (control word required)
		5_{16}	=	Prepare for write (control word required)
		6_{16}	=	Master clear
		7_{16}	=	Clear fault status registers
		C_{16}	=	Read IOU status summary (reads one byte, control word not required)
TYPE (bits 0-3)	Code	$0\text{-}7_{16}$	=	IOU registers are read circularly (byte 0 follows byte 7) from the byte specified by the TYPE field

Table 17-13. MCH Function Word Bit Assignments, Model 990 and CYBER 990E, 995E, and 994 (IOU)

Field				Description (MCH Function Word to IOU)
CONNECT (bits 8-11)	Code	0_{16}	=	Connect IOU maintenance registers
OPCODE (bits 4-7)		3_{16}	=	Clear LED
		4_{16}	=	Read
		5_{16}	=	Write
		6_{16}	=	Master clear ADU/CMI
		7_{16}	=	Clear Fault Status Register
		8_{16}	=	Echo
		C_{16}	=	Request Summary Status Byte
TYPE (bits 0-3)				Type code must equal 0

MCH Control Words

Some function words must be followed by two 8-bit control words, which specify the internal address of the register to be accessed. This is accomplished by transmitting two PP words where the rightmost 8 bits in each word are used. Control words are required for the following.

- Function words to CP with opcodes 4/5.
- Function words to CM and IOU with opcodes 4/5.
- Function words to CP, CM, and IOU with opcode 8 (echo).

Refer to tables 17-14 through 17-16 for CP, CM, and IOU internal address assignments.

MCH Programming for Halt/Start (Opcode 0/1)

These operations consist of the output of a function word. A halt opcode halts the processor without damaging the process being executed, including the integrity of the interunit communication of the halted processor such as CYBER 170 exchange request communication, central memory communications, and the process state. If the process is subsequently restarted without performing any other MCH operations or after performing read/write with certain precautions, the process continues without damage.

MCH Clear LED (Opcode 3)

This operation clears all LEDs associated with pak errors and is intended, but not required, for use at system initialization. For maintenance reasons, this operation can also clear LEDs without initializing and master-clearing.

MCH Programming for Read/Write (Opcode 4/5)

Refer to Programming for PP Data Input/Output in this chapter for a more complete procedure. In general terms, proceed as follows:

1. Issue the function with opcode 4/5.
2. Output the first control word.
3. Verify the error flag is clear.
4. Output the second control word.
5. Verify the error flag is clear.
6. Input/output the required number of data words.
7. Verify the error flag is clear.

Reading a nonexistent register returns all zeros. Writing to a read-only register or to a nonexistent register does not alter any register. Most registers are read/write as 64-bit (8-byte) registers, requiring the input/ output of eight MCH data words. Most registers that are physically smaller than eight bytes are right-justified with zero-fill. Exceptions are as follows:

- Reading a status summary register repeats the status information in each byte.
- The IOU may disconnect the MCH without affecting subsequent MCH operations in the following cases:
  - After reading 1 to 8 bytes from any maintenance register.
  - After writing 1 byte to a corrected error log register.
  - After writing 1 byte to an uncorrected error log register.

The following MCH operations on CP registers can be performed with the CP running or halted.

- Read CP status summary register.
- Read CP fault status register.
- Read CP corrected error log registers.
- Read CP options installed registers.
- Read CP element identifier register.
- Read/write CP dependent environmental control register.
- Read/write test mode control registers.
- Clear errors.

To read/write other CP registers, the CP must be running since these registers are accessed by microcode. Refer to the Maintenance Register Codes Booklet listed under Additional Related Manuals in About This Manual for register bit assignments.

MCH Programming for Master Clear/Clear Errors (Opcode 6/7)

These operations consist of the output of a single function word. The master clear immediately and arbitrarily clears the connected unit without regard to possible information loss. Clear errors clears the error indicators in the connected unit. To avoid loss of error information while the errors are cleared, the unit concerned should be halted.

MCH Echo (Opcode 8)

This operation checks the data path between the MCH and the IOU MAC. Following the operation MCH is activated and 2 bytes are sent to IOU MAC. IOU ignores the first byte and latches the second byte in the Address Holding Register in any data pattern. MCH is deactivated after the second byte is accepted in IOU MAC, and the channel is activated followed by an input sequence. IOU MAC sends data (contents of Address Holding Register) upon receiving the Active signal and subsequent Empty signals. There is no restriction on the number of data words read.

MCH Programming for Read IOU Status Summary (Opcode C, IOU Only)

This operation is an alternative, faster means of reading the IOU status summary register.

1. Issue function with opcode C.
2. Input status summary byte.

Table 17-14. CP Internal Address Assignments

Hex[1]	Octal[1]	Type[2]	Type[2]	Description
00	000	R	A	Status summary register.
10	020	R	A	Element identifier register.
30	060	R	A	Dependent environment control register.
42	082	R	M	Monitor condition register.
80-89	200-211	R	A	Processor fault status registers 1 through 9.

Notes:

1. The internal address is the second byte of two 8-bit control words, which must be supplied after a function word output with OPCODE = 4/5. The first byte is discarded.

2. R = read, W = write; A = always accessible, M = microcode accessible.

Table 17-15. CM Internal Address Assignments

Hex[1]	Octal[1]	Type[2]	Description
00	000	R	Status summary register.
10	020	R	Element identifier register.
12	022	R	Options installed register.
A0	240	R/W	Corrected error log register.
A4	244	R/W	Uncorrected error log 1 register.
A8	250	R/W	Uncorrected error log 2 register.

Notes:

1. The internal address is the second byte of two 8-bit control words, which must be issued after a function word output with OPCODE = 4/5. The first byte is discarded.

2. R = read, W = write.

Table 17-16. IOU Internal Address Assignments

Hex[1]	Octal[1]	Type[2]	Description
00	000	R	Status summary register.
10	020	R	Element identifier register.
12	022	R	Options installed register.
18	030	R/W	Fault status mask register.
40	100	R	Status register.
80	200	R/W	Fault status 1 register.
81	201	R/W	Fault status 2 register.
A0	240	R/W	Test mode.

Notes:

1. The internal address is the second byte of two 8-bit control words, which must be issued after a function word output with OPCODE = 4/5. The first byte is discarded.

2. R = read, W = write.

Glossary A

A

ADU
Assembly/disassembly unit

A register
Address register

ASCII
American Standard Code for Information Interchange

ASID
Active segment identifier

B

B register
Indexing register

BAS
Barrel and slot

BC
Base constant

BCD
Binary-coded decimal

BDP
Business data processing

BMRS
Broadcast master reset

BN
Byte number

BS
Binding section

BSP
Binding section pointer

BSR
Bit significant response

BSS
Bus slave select

C

CBP
Code base pointer

CCEL/MCEL
Cache/map corrected error log

CEJ/MEJ
Central exchange jump/monitor exchange jump

CEL
Corrected error log

CEM
Configuration environment monitor

CF
Critical frame pointer

CFF
Critical frame flag

CIO
Concurrent input/output

CM
Central memory

CMC
Central memory control

CP
Central processing unit

CRT
Cathode-ray tube

CSF
Current stack frame pointer

D

DC
Debug code

DCD
Data carrier detector

DEC
Model-dependent environment control

D/F
Data/function bit

DI
Debug index

DLP
Debug list pointer

DM
Debug mask

DMA
Direct memory access

DMR
Debug mask register

DSC
Display station controller

DSP
Dynamic space pointer

DSR
Data set ready

DTR
Data terminal ready

DUE
Dependent environment control

E

EBCDIC
Expanded binary coded decimal interchange code

EC
Environment control

ECL
Emitter-coupled logic

ECM
Extended central memory

ECS
Extended core storage

EIA
Electronics Industries Association

EID
Element identifier

EM
Exit mode

EPF
External procedure flag

EQ
Equal condition

ES
End suppression toggle (BDP edit instruction)

ESM-II
Extended semiconductor memory II

F

FIFO
First-in, first-out

FL
Field length

FLC
Central memory field length register

FLE
Extended core storage field length register

FP
Floating-point

FS
Fault status

G

G/L
Global/local

I

IC
Integrated circuit

ILH
Instruction look-ahead

IPI
Intelligent peripheral interface

I/O
Input/output

IOU
Input/output unit

ISI
Intelligent standard interface

J

JPS
Job process state pointer

K

KEY
Key

L

LED
Light-emitting diode

LOCK
Lock

LPID
Last processor identification

LRN
Largest ring number

LSB
Least significant bit

LSI
Large-scale integration

M

MA
Monitor address

MAC
Maintenance access control

MCH
Maintenance channel

MCR
Monitor condition register

MCU
Maintenance control unit

MDF
Model-dependent flags

MDW
Model-dependent word

MF
Monitor flag

MMR
Monitor mask register

MOP
Micro-operator (BDP edit instruction)

MOS
Metal-oxide-semiconductor

MPS
Monitor process state pointer

MSB
Most significant bit

N

NIO
Noncurrent input/output

NOS
Network Operating System

NOS/VE
Network Operating System/Virtual Environment

NS
Negative sign toggle

ON
Occurrence number

OPCODE
Operation code

OS
Operating system

P

P register
Program address register

PCB
Printed-circuit board

PE
Parity error

PFA
Page frame address

PFS
Processor fault status

PID
Processor identifier

PIT
Process interval timer

PMF
Performance monitoring flag

PN
Page number

PND
Process-not-damaged flag

PO
Page offset

PP
Peripheral processor

PPM
Peripheral processor memory

PROM
Programmable read-only memory

PSA
Previous save area pointer

PSF
Previous stack frame

PSM
Page size mask

PTA
Page table address

PTE
Page table entry

PTL
Page table length

PTM
Processor test mode

PVA
Process virtual address

R

RAC
Central memory reference address register

RAE
Extended core storage reference address register

RAM
Random-access memory

RI
Radial interface

RMA
Real memory address

RN
Ring number

ROM
Read-only memory

RP
Read permission (segment descriptor field)

RTS
Request to send

S

SCT
Special characters table (BDP edit instruction)

SDE
Segment descriptor table entries

SDT
Segment descriptor table

SECDED
Single error correction/double error detection

SEG
Process segment number

SFSA
Stack frame save area

SIT
System interval timer

SM
The symbol (BDP edit instruction)

SN
Negative sign (BDP edit instruction)

SPID
Segment page identifier

SPT
System page table

SR
Select reset

SRT
Subscript range table

SS
Status summary

STA
Segment table address

STL
Segment table length

SV
Specification value

SVA
System virtual address

T

T'
T-prime register

TE
Trap enable

TED
Trap-enable delay

TEF
Trap-enable flip-flop

TER
Terminate

TM
Test mode

TOS
Top of stack

TP
Trap pointer

U

UART
Universal asynchronous receiver-transmitter

UCR
User condition register

UEL
Uncorrected error log

UEM
Unified extended memory

UMR
User mask register

UTP
Untranslatable pointer

UVMID
Untranslatable virtual machine identifier

V

V
Valid bit

VC
Search control code (page descriptor field)

VL
Segment validation (segment descriptor field)

VMCL
Virtual machine capability list

VMID
Virtual machine identifier

W

WP
Write access control (segment descriptor field)

WR
Write/read

X

X register
Operand register

XP
Execute access control (segment descriptor field)

Z

ZF
Zero field toggle (BDP edit instruction)

ZFI
Zero fill inhibit

Index

Index

A

A barrel failure
835 3-5
840, 850, and 860 9-5
845 and 855 6-5
A register, PP
Set instructions
835 2-5
840, 850, and 860 8-5
845 and 855 5-5
990 11-5
990E, 995E, and 994 14-5
835 2-26
840, 850, and 860 8-26
845 and 855 5-28
990 11-28
990E, 995E, and 994 14-28
A registers, CP
Set instructions 16-50, 51, 52, 53, 54
835 1-3, 6; 2-10
840, 850, and 860 7-4, 7; 8-10
845 and 855 4-5, 8; 5-10
990 8-10; 10-4, 7
990E, 995E, and 994 13-5, 8; 14-10
Absolute CM address formation 17-26
835 2-26
840, 850, and 860 8-26
845 and 855 5-28
990 11-28
990E, 995E, and 994 14-28
Absolute UEM address formation
835 2-22
840, 850, and 860 8-22
845 and 855 5-24
990 11-22
990E, 995E, and 994 14-22
Access, CM
835 2-16, 29
840, 850, and 860 8-16, 30
845 and 855 5-16, 17, 32
990 11-16, 34
990E, 995E, and 994 14-16, 35
Activate channel 16-104; 17-35
Active channel 17-35
Add instructions 16-78, 86
Address designator 16-3
Address format, CM
835 2-15
840, 850, and 860 8-15
845 and 855 5-15, 16
990 11-15
990E, 995E, and 994 14-15
Address out of range error 17-17, 20
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Address registers, see A registers
Addressing
Absolute address formation 17-26
835 2-22, 26
840, 850, and 860 8-22, 26
845 and 855 5-24, 28
990 11-22, 28
By PPs 17-29
CM 17-28
CM by PPs 17-29
PPM by PPs 17-29
Relative address formation 17-26, 27
Section in CP
835 1-6; 2-1, 14
840, 850, and 860 7-7; 8-1, 14
845 and 855 4-8; 5-1, 14
990 10-7; 11-1, 14
990E, 995E, and 994 13-8; 14-1, 14
990E, 995E, and 994 14-22, 28
Addressing mode
Expanded 17-3, 28
Standard 17-3, 28
Alphanumeric characters 17-42
Assembly/disassembly
835 2-29
840, 850, and 860 8-30
845 and 855 5-32
990 11-34
990E, 995E, and 994 14-35
Auto mode bit
835 3-3
840, 850, and 860 9-3
845 and 855 6-3

B

B registers
Set instructions 16-50, 55, 56, 57
835 2-5
840, 850, and 860 8-5
845 and 855 5-5
990 11-5
990E, 995E, and 994 14-5
835 1-3, 6; 2-10
840, 850, and 860 7-4, 7; 8-10
845 and 855 4-5, 8; 5-10
990 10-4, 7; 11-10
990E, 995E, and 994 13-5, 8; 14-10
Bank cycle times
835 2-16
840, 850, and 860 8-16
845 and 855 5-17
990 11-16
990E, 995E, and 994 14-16

Bank select field
835 2-15
840, 850, and 860 8-15
845 and 855 5-15
990 11-15
990E, 995E, and 994 14-15
Banks, CM
835 1-4, 8; 2-15
840, 850, and 860 7-5, 9; 8-15
845 and 855 4-6, 10; 5-15
990 10-5, 9; 11-15
990E, 995E, and 994 13-6, 10; 14-15
Barrel and slot
835 2-24
840, 850, and 860 8-24
845 and 855 5-26
990 11-26
990E, 995E, and 994 14-26
Barrels
Logical barrel 0 selection
835 3-4, 11
840, 850, and 860 9-4, 12
845 and 855 6-4, 16
990 12-6, 8
990E, 995E, and 994 15-6, 8
Reconfiguration
835 3-4, 11
840, 850, and 860 9-4, 12
845 and 855 6-4, 16
990 12-7
990E, 995E, and 994 15-7
835 1-9; 2-23
840, 850, and 860 7-10; 8-23
845 and 855 4-11; 5-25
990 11-24
990E, 995E, and 994 14-24
Bit numbering 7; 16-1
Block copy flag
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Block copy from CM instruction 16-16, 18; 17-28
Block copy from UEM instruction 16-16, 17; 17-28
Block copy instruction sequence
835 2-7
840, 850, and 860 8-7
845 and 855 5-7
990 11-7
990E, 995E, and 994 14-7
Block copy instructions 16-9
Boolean instruction sequence
835 2-2
840, 850, and 860 8-2
845 and 855 5-2
990 11-2
990E, 995E, and 994 14-2
Bounds register
835 1-4; 2-22
840, 850, and 860 7-5; 8-22
845 and 855 4-6; 5-24
990 10-5; 11-22
990E, 995E, and 994 13-6; 14-22
Branch destination address 16-40
Branch instructions 16-9
Definite 16-9, 12
Equal to zero 16-9
In range 16-9, 11
Indefinite 16-9, 13
Negative 16-9, 11
Not equal to zero 16-9, 10
Out of range 16-9, 12
Positive 16-9, 10
Register greater than or equal 16-9, 14
Register less than 16-9, 15
Registers equal 16-9, 13
Registers not equal 16-9, 14
835 2-8
840, 850, and 860 8-8
845 and 855 5-8
990 11-8
990E, 995E, and 994 14-8
Branch target address
835 2-1
840, 850, and 860 8-1
845 and 855 5-1
990 11-1
990E, 995E, and 994 14-1
Buffers, queuing
835 1-8
840, 850, and 860 7-9
845 and 855 4-10
990 10-9
990E, 995E, and 994 13-10

C

Cache corrected error log register 17-64
Cache invalidation bus interface, 835 1-9
Cache memory
835 1-7; 2-14
840, 850, and 860 7-8; 8-14
845 and 855 4-9; 5-14
990 10-8; 11-14
990E, 995E, and 994 13-9; 14-14
Cathode-ray tube, see CRT
Central exchange jump instruction 16-41, 42
Central memory control (CMC)
840, 850, and 860 7-8; 8-14
845 and 855 4-9; 5-14
990 10-9; 11-14
990E, 995E, and 994 13-9; 14-14
Central memory, see CM
Central processor, see CP

Central read instruction 16-96, 97
Central write instruction 16-96, 98, 99
Channel active flag 17-35
Channel control flag 17-34
Channel error flag 17-36
Channel flag instruction 17-36
Channel, I/O, see I/O channels
Channel, maintenance, see Maintenance channel
Channel marker flag 17-36
Channel transfer timing 17-36
Character address designator 16-3
Character codes 17-45
Character manipulation 17-13
Character mode 17-42
Character size 17-42
Characteristics,
  Functional
    835 1-2
    840, 850, and 860 7-3
    845 and 855 4-4
    990 10-3
    990E, 995E, and 994 13-4
  Physical
    835 1-1
    840, 850, and 860 7-1
    845 and 855 4-1
    990 10-1
    990E, 995E, and 994 13-1
Chassis configuration, see System configuration
Chip address field
  835 2-15
  840, 850, and 860 8-15
  845 and 855 5-15
  990 11-15
  990E, 995E, and 994 14-15
CIF (CMU interrupted flag)
  835 2-12
  840, 850, and 860 8-12
  845 and 855 5-12
  990 11-12
  990E, 995E, and 994 14-12
CLEAR AUTO switch
  835 3-3
  840, 850, and 860 9-3
  845 and 855 6-3
Clear channel error flag instruction 16-95
Clear channel flag instruction 16-95
CM
  Access
    835 2-29
    840, 850, and 860 8-30
    845 and 855 5-32
    990 11-34
    990E, 995E, and 994 14-35
  Access time
    835 2-16
    840, 850, and 860 8-16
    845 and 855 5-16, 17
    990 11-16
    990E, 995E, and 994 14-16
Address format
  835 2-15
  840, 850, and 860 8-15
  845 and 855 5-15, 16
  990 11-15
  990E, 995E, and 994 14-15
Address formation
  835 2-2, 15
  840, 850, and 860 8-2, 15
  845 and 855 5-2, 15
  990 11-2, 15
  990E, 995E, and 994 14-2, 15
Addressing by PPs 17-29
Banks
  835 1-2, 4, 8; 2-15
  840, 850, and 860 7-3, 5, 9; 8-15
  845 and 855 4-4, 6, 10; 5-15
  990 10-3, 5, 9; 11-15
Block copy instructions 17-28
Bounds register
  835 1-4; 2-22
  840, 850, and 860 7-5; 8-22
  845 and 855 4-6; 5-24
  990 10-5; 11-22
  990E, 995E, and 994 13-6; 14-22
Configuration switches
  835 3-6
  840, 850, and 860 9-6
  845 and 855 6-7
  990 12-4
  990E, 995E, and 994 15-4
Controls
  835 3-1
  840, 850, and 860 9-1
  845 and 855 6-1
  990 12-1
  990E, 995E, and 994 15-1
Cycle times
  835 2-16
  840, 850, and 860 8-16
  845 and 855 5-17
  990 11-16
  990E, 995E, and 994 14-16
Description
  835 1-8
  840, 850, and 860 7-9
  845 and 855 4-10
  990 10-9
  990E, 995E, and 994 13-10
Distributor
  840, 850, and 860 8-14
  845 and 855 5-14
  990E, 995E, and 994 13-10; 14-14
Extended, see UEM
Field length register, see FLC register

Functional characteristics
835 1-2
840, 850, and 860 7-3
845 and 855 4-4
990 10-3
990E, 995E, and 994 13-4
Functional description
835 2-15
840, 850, and 860 8-15
845 and 855 5-15
990 11-15
990E, 995E, and 994 14-15
Layout
835 2-22
840, 850, and 860 8-22
845 and 855 5-24
990 11-22
990E, 995E, and 994 14-22
Map 17-27
MCH function word 17-57
Port priority
835 2-16
840, 850, and 860 8-16
845 and 855 5-17
990 11-16
990E, 995E, and 994 14-16
Ports
835 1-4; 2-16
840, 850, and 860 7-5; 8-16
845 and 855 4-6; 5-17
990 10-5; 11-16
990E, 995E, and 994 13-6; 14-16
Programming 17-26
Read word from 16-61; 17-28
Reconfiguration
835 2-22; 3-2, 7, 8
840, 850, and 860 8-22; 9-2, 7, 8
845 and 855 5-24; 6-2, 8, 9, 10, 11
990 11-22; 12-5
990E, 995E, and 994 14-22; 15-5
Reconfiguration switches
835 3-2
840, 850, and 860 9-2
845 and 855 6-2
Reference address register, see RAC register
Word 16-1
Write word to 16-61; 17-28
990E, 995E, and 994 13-4, 6, 10; 14-15
CMC
840, 850, and 860 7-8; 8-14
845 and 855 4-9; 5-14
990 10-8; 11-14
990E, 995E, and 994 13-9; 14-14
CMU error exit 17-18, 21
CMU instructions, see Compare/move instructions
CMU interrupted flag
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Codes, display station 17-45
Collate table for compare/move instruction 16-49
COLUMN DEGRADE SWITCH 15-4
Compare collated instruction 16-48; 17-24
Compare/move arithmetic 17-13
Compare/move instruction sequence
835 2-6
840, 850, and 860 8-6
845 and 855 5-6
990 11-6
990E, 995E, and 994 14-6
Compare/move instructions 16-45
Compare collated 16-48; 17-24
Compare uncollated 16-49; 17-24
Designators within instructions 16-3
Move direct 16-47; 17-13
Move indirect 16-47; 17-13
Support registers
835 2-11
840, 850, and 860 8-11
845 and 855 5-11
990 11-11
990E, 995E, and 994 14-11
835 2-6
840, 850, and 860 8-6
845 and 855 5-6
990 11-6
990E, 995E, and 994 14-6
Compare uncollated instruction 16-49; 17-24
Complement, instructions using 16-23, 24, 25, 26
Conditional software errors 17-25
Configuration switches, CM
835 3-1, 6
840, 850, and 860 9-1, 6
845 and 855 6-1, 7
990 12-1, 4
990E, 995E, and 994 15-1, 4
Configuration, system
835 1-1
840, 850, and 860 7-2
845 and 855 4-2, 3
990 10-2
990E, 995E, and 994 13-2, 3
Constant address instructions 17-29
Control checks
835 3-3, 4, 5
840, 850, and 860 9-3, 4, 5
845 and 855 6-3, 4, 5
990 12-3
990E, 995E, and 994 15-3

Control failure
835 3-5
840, 850, and 860 9-5
845 and 855 6-5
Control flags 17-34
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-2
Control words, maintenance channel 17-61
Controls
835 3-1, 3, 4, 5
840, 850, and 860 9-1, 3, 4, 5
845 and 855 6-1, 3, 4, 5
990 12-1, 3
990E, 995E, and 994 15-1, 3
Conversion
Fixed-point to floating-point 16-6
Floating-point to fixed-point 16-7
Coordinates, display station 17-45, 46
Counter, running 16-69
CP
Addressing section
835 1-6; 2-1, 14
840, 850, and 860 7-7; 8-1, 14
845 and 855 4-8; 5-1, 14
990 10-7; 11-1, 14
990E, 995E, and 994 13-8; 14-1, 14
Cache memory
835 1-7; 2-14
840, 850, and 860 7-8; 8-14
845 and 855 4-9; 5-14
990 10-8; 11-14
990E, 995E, and 994 13-9; 14-14
Description
835 1-6
840, 850, and 860 7-7
845 and 855 4-8
990 10-7
990E, 995E, and 994 13-8
Execution section
835 1-7; 2-14
840, 850, and 860 7-8; 8-14
845 and 855 4-9; 5-14
990 10-8; 11-14
990E, 995E, and 994 13-9; 14-14
Functional characteristics
835 1-3
840, 850, and 860 7-4
845 and 855 4-5
990 10-4
990E, 995E, and 994 13-4.1
Functional description
835 2-1
840, 850, and 860 8-1
845 and 855 5-1
990 11-1
990E, 995E, and 994 14-1
Instruction descriptions 16-1
Instruction designators 16-3
Instruction section
835 1-6; 2-1
840, 850, and 860 7-7; 8-1
845 and 855 4-8; 5-1
990 10-7; 11-1
990E, 995E, and 994 13-8; 14-1
MCH function word 17-57
Operating modes 16-4
Programming 17-1
Registers
835 1-6; 2-9
840, 850, and 860 7-7; 8-9
845 and 855 4-8; 5-9
990 10-7; 11-9
990E, 995E, and 994 13-8; 14-9
CRT 17-42
835 1-10
840, 850, and 860 7-11
845 and 855 4-12
990 10-11
990E, 995E, and 994 13-12
CYBER 170 exchange jump 16-4, 42; 17-1
835 2-7, 9
840, 850, and 860 8-7, 9
845 and 855 5-7, 9
990 11-7, 9
990E, 995E, and 994 14-7, 9
CYBER 170 exchange package 16-42; 17-1
Contents 17-1
835 2-7, 9
840, 850, and 860 8-7, 9
845 and 855 5-7, 9
990 11-7, 9
990E, 995E, and 994 14-7, 9
CYBER 170 exchange sequence
835 2-7
840, 850, and 860 8-7
845 and 855 5-7
990 11-7
990E, 995E, and 994 14-7
CYBER 170 job mode 16-4; 17-1
Error exits 17-20, 21, 23
CYBER 170 monitor flag 16-4; 17-1
835 2-13
840, 850, and 860 8-13
845 and 855 5-13
990 11-13
990E, 995E, and 994 14-13
CYBER 170 monitor mode 16-4; 17-1
Error exits 17-17, 18, 19
Cycle times, CM
835 2-16
840, 850, and 860 8-16
845 and 855 5-17
990 11-16
990E, 995E, and 994 14-16

D

Data carrier detector (DCD) 17-51
Data character 17-52
Data display 17-42
Data formats 16-1
  PP 16-70
Data input sequence 17-38, 39
DATA (octal display) indicator
  835 3-3
  840, 850, and 860 9-3
  845 and 855 6-3
Data output sequence 17-40, 41
Data parity error 17-36
Data set ready (DSR) 17-51
Data terminal ready (DTR) 17-54
DCD, see Data carrier detector
Deactivate channel 16-100, 104
DEAD START ERROR indicator
  835 3-4
  840, 850, and 860 9-4
  845 and 855 6-4
DEAD START switch
  835 2-24; 3-4
  840, 850, and 860 8-24; 9-4
  845 and 855 5-26; 6-4
  990 11-26; 12-1
  990E, 995E, and 994 14-26; 15-1
Deadstart
  Controls
    835 3-1, 3
    840, 850, and 860 9-1, 3
    845 and 855 6-1, 3
    990 21-1
    990E, 995E, and 994 15-1
  Indicators
    835 3-1, 3
    840, 850, and 860 9-1, 3
    845 and 855 6-1, 3
  Options Display
    990 12-1
    990E, 995E, and 994 15-1
  Panel
    835 2-24; 3-2
    840, 850, and 860 8-24; 9-2
    845 and 855 5-26; 6-2
    990 11-26; 12-1
    990E, 995E, and 994 14-26; 15-1
  Program
    835 3-3
    840, 850, and 860 9-3
    845 and 855 6-3
    990 12-3
    990E, 995E, and 994 15-3
  Sequence
    835 2-24; 3-3
    840, 850, and 860 8-24; 9-3
    845 and 855 5-26; 6-3
    990 11-26; 12-6
    990E, 995E, and 994 14-26; 15-6
  Switches
    835 3-1, 3
    840, 850, and 860 9-1, 3
    845 and 855 6-1, 3
    990 12-1
    990E, 995E, and 994 15-1
DEGRADE SWITCH, COLUMN 15-4
Description, system
  835 1-1
  840, 850, and 860 7-1
  845 and 855 4-1
  990 10-1
  990E, 995E, and 994 13-1
Direct address instructions 17-30
Direct read/write instruction sequence
  835 2-7
  840, 850, and 860 8-7
  845 and 855 5-7
  990 11-7
  990E, 995E, and 994 14-7
Direct read/write instructions,
  CM 17-28
    835 2-7
    840, 850, and 860 8-7
    845 and 855 5-7
    990 11-7
    990E, 995E, and 994 14-7
  UEM 17-28
    835 2-7
    840, 850, and 860 8-7
    845 and 855 5-7
    990 11-7
    990E, 995E, and 994 14-7
Direct 12-bit address 17-30
Direct 18-bit operand 17-29
Direct 6-bit address 17-30
Direct 6-bit operand 17-29
Display character codes 17-45
Display controller interface
  835 1-5
  840, 850, and 860 7-6
  845 and 855 4-7
  990 10-10
  990E, 995E, and 994 13-11
Display station
  Character codes 17-42
  Character sizes 17-45
  Codes 17-45
  Controller
    835 2-29
    840, 850, and 860 8-29
    845 and 855 5-31
    990 11-31
    990E, 995E, and 994 14-31
  Coordinates 17-46
  Keyboard 17-42
  Mode 17-42, 45
  Output function code 17-45
  Presentation 17-45, 46
  Programming 44; 17-43
  Timing considerations 17-47

835 1-10
840, 850, and 860 7-11
845 and 855 4-12
990 10-11
990E, 995E, and 994 13-12
Dot mode 17-45
Double-bit errors 17-19, 23, 25
835 2-17
840, 850, and 860 8-17
845 and 855 5-19
990 11-17
990E, 995E, and 994 14-7
Double-precision results 17-10
DSC, see Display station controller
DSR, see Data set ready
DTR, see Data terminal ready

E

ECL, see Emitter-coupled logic
EM register
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
EMC, see Exit mode conditions
EMERGENCY OFF switch
835 3-9
840, 850, and 860 9-10
845 and 855 6-14
990 12-6
990E, 995E, and 994 15-6
Emitter-coupled logic (ECL)
835 1-2
840, 850, and 860 7-3
845 and 855 4-4
990 10-3
990E, 995E, and 994 13-4
Empty register 17-35
EMS bits, see Exit mode selection bits
Environment control register IOU
835 3-3
840, 850, and 860 9-3
845 and 855 6-3
Error conditions
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-2
Error correction code bits
835 2-17
840, 850, and 860 8-17
845 and 855 5-19
990 11-17
990E, 995E, and 994 14-17
Error exit 16-4
Error exit condition codes 17-16
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Error exit to MA instruction 16-66
Error exits 17-16
In CYBER 170 job mode 17-20
In CYBER 170 monitor mode 17-17
Error flag 17-34
Error flag instruction 17-36
Error handling 17-16
Error processing 17-16
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Error response 17-36
Exchange jump instruction 16-42
Exchange jump, see CYBER 170 exchange jump
Exchange package, see CYBER 170 exchange package
Exchange sequence, see CYBER 170 exchange sequence
Execution interval 17-4
835 2-9
840, 850, and 860 8-9
845 and 855 5-9
990 11-9
990E, 995E, and 994 14-9
Execution section
835 1-7; 2-14
840, 850, and 860 7-8; 8-14
845 and 855 4-9; 5-14
990 10-8; 11-14
990E, 995E, and 994 13-9; 14-14
Execution times, PP 16-109
Exit condition codes 17-15
Exit condition register 17-15
Exit mode conditions 17-15
Exit mode register, see EM register
Exit mode selection bits 17-15
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Expanded addressing mode 17-3, 28
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Expanded addressing select flag
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12

990E, 995E, and 994 14-12
Extended purge control, see Instruction lookahead purge control
External interface, IOU
835 1-9; 2-28
840, 850, and 860 7-10; 8-29
845 and 855 4-11; 5-31
990 10-10; 11-31
990E, 995E, and 994 13-11; 14-13

F

Fast deadstart
835 3-4
840, 850, and 860 9-4
845 and 855 6-4
Field length for CM register, see FLC register
Field length for UEM register, see FLE register
Firmware or control failure
835 3-5
840, 850, and 860 9-5
845 and 855 6-5
Fixed-point arithmetic 17-4
Normalized numbers 17-10
Overflow 17-7
Fixed-point to floating-point conversion 16-6
Flag register
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
FLC register
835 2-11
840, 850, and 860 8-11
845 and 855 5-11
990 11-11
990E, 995E, and 994 14-11
FLE register
835 2-13
840, 850, and 860 8-13
845 and 855 5-13
990 11-13
990E, 995E, and 994 14-13
Floating-add instruction sequence
835 2-4
840, 850, and 860 8-4
845 and 855 5-4
990 11-4
990E, 995E, and 994 14-4
Floating divide instruction 16-37
Floating-divide instruction sequence
835 2-4
840, 850, and 860 8-4
845 and 855 5-4
990 11-4
990E, 995E, and 994 14-4
Floating double-precision difference instruction 16-32
Floating double-precision product instruction 16-36
Floating double-precision sum instruction 16-29
Floating-multiply instruction sequence
835 2-4
840, 850, and 860 8-4
845 and 855 5-4
990 11-4
990E, 995E, and 994 14-4
Floating-point arithmetic 17-4
Double-precision 17-10
Format 17-4
Indefinite 17-7
Nonstandard operands 17-8
Normalized numbers 17-10
Overflow 17-7
Packing 17-5
Rounding 17-10
Underflow 17-7
Floating-point difference instruction 16-31
Floating-point sum instruction 16-28
Floating-point to fixed-point conversion 16-6
Floating product instruction 16-34
Form mask instruction 16-68
Framing error 17-52
Free running counter 16-69
FREQ MARGIN switch
835 3-3
840, 850, and 860 9-3
845 and 855 6-3
Frequency margin
835 3-3
840, 850, and 860 9-3
845 and 855 6-3
FS register, see Fault status register
Full register flag 17-35
Function instruction 17-35
Function on channel instruction 16-105
Function words, maintenance channel 17-57
Functional characteristics
835 1-2
840, 850, and 860 7-3
845 and 855 4-4
990 10-3
990E, 995E, and 994 13-4
Functional description
CM
835 2-15
840, 850, and 860 8-15
845 and 855 5-15
990 11-15
990E, 995E, and 994 14-15

CP
835 2-1
840, 850, and 860 8-1
845 and 855 5-1
990 11-1
990E, 995E, and 994 14-1
IOU
835 2-23
840, 850, and 860 8-23
845 and 855 5-25
990 11-23
990E, 995E, and 994 14-23

G

Glossary A-1
Graphic mode 17-45

H

Halt CP 17-61

I

I/O channel failure
835 3-5
840, 850, and 860 9-5
845 and 855 6-5
I/O channels
Activate 17-35
Active/inactive 17-35
Communications 17-36
External interface
835 1-9; 2-28
840, 850, and 860 7-10; 8-29
845 and 855 4-11; 5-31
990 10-10; 11-31
990E, 995E, and 994 13-11; 14-31
Internal interface
835 1-9; 2-28
840, 850, and 860 7-10; 8-29
845 and 855 4-11; 5-31
990 10-10; 11-31
990E, 995E, and 994 13-11; 14-31
Operation 17-34
Parity 17-19, 23, 32, 36, 53
Reservation 17-32
835 2-23, 28
840, 850, and 860 8-23, 29
845 and 855 5-25, 31
990 11-23, 31
990E, 995E, and 994 14-23, 31
I/O transfers 17-36, 38
ILH, see Instruction lookahead
Illegal instructions 16-66; 17-24
Inactive channel 17-35
Increment instruction sequence
835 2-4
840, 850, and 860 8-4
845 and 855 5-4
990 11-6
990E, 995E, and 994 14-4
Indefinite condition 17-19, 23
Indefinite, floating-point 17-7
Indefinite operand error
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Index registers, see B registers
Indexed direct address instructions 17-30
Indexed 12-bit address 17-30
Indicators
835 3-1
840, 850, and 860 9-1
845 and 855 6-1
990 12-1
990E, 995E, and 994 15-1
Indirect address instructions 17-30
Indirect 6-bit address 17-30
Infinite operand
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Input data parity error 17-36
Input from channel instruction 16-102
Input/output channel, see I/O channel
Input/output unit, see IOU
Input sequence 17-38
Instruction control sequences
Block copy sequence
835 2-7
840, 850, and 860 8-7
845 and 855 5-7
990 11-7
990E, 995E, and 994 14-7
Boolean sequence
835 2-2
840, 850, and 860 8-2
845 and 855 5-2
990 11-2
990E, 995E, and 994 14-2
Compare/move sequence
835 2-6
840, 850, and 860 8-6
845 and 855 5-6
990 11-6
990E, 995E, and 994 14-6
CYBER 170 exchange sequence
835 2-7
840, 850, and 860 8-7
845 and 855 5-7
990 11-7
990E, 995E, and 994 14-7
Direct read/write sequence
835 2-7
840, 850, and 860 8-7
845 and 855 5-7

990 11-7
990E, 995E, and 994 14-7
Floating-add sequence
835 2-4
840, 850, and 860 8-4
845 and 855 5-4
990 11-4
990E, 995E, and 994 14-4
Floating-divide sequence
835 2-4
840, 850, and 860 8-4
845 and 855 5-4
990 11-4
990E, 995E, and 994 14-4
Foating-multiply sequence
835 2-4
840, 850, and 860 8-4
845 and 855 5-4
990 11-4
990E, 995E, and 994 14-4
Increment sequence
Normal jump sequence
835 2-8
840, 850, and 860 8-8
845 and 855 5-8
990 11-8
990E, 995E, and 994 14-8
Return jump sequence
835 2-8
840, 850, and 860 8-8
845 and 855 5-8
990 11-8
990E, 995E, and 994 14-8
Shift sequence
835 2-3
840, 850, and 860 8-3
845 and 855 5-3
990 11-3
990E, 995E, and 994 14-3
835 2-2, 4
840, 850, and 860 8-4
845 and 855 5-4
990 11-4
990E, 995E, and 994 14-4
840, 850, and 860 8-2
845 and 855 5-2
990 11-2
990E, 995E, and 994 14-2
Instruction descriptions
CP 16-4
PP 16-70
Instruction designators
CP 16-3
PP 16-71
Instruction execution times, PP 16-109
Instruction lookahead
Purge control 17-14
835 2-1
840, 850, and 860 8-1
845 and 855 5-1
990 11-1
990E, 995E, and 994 14-1
Instruction lookahead purge control 17-14
Instruction lookahead purge flag
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Instruction prefetch, see Instruction lookahead
Instruction purge flag
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Instruction section
835 1-6; 2-1
840, 850, and 860 7-7; 8-1
845 and 855 4-8; 5-1
990 10-7; 11-1
990E, 995E, and 994 13-8; 14-1
Instruction word designators
CP 16-3
PP 16-71
Instruction word format
CP 16-1
Instructions, CP (see also inside front cover)
Block copy from CM 16-17; 17-28
Block copy from UEM 16-18; 17-28
Branch 9; 10; 11; 12; 13; 14; 15; 16
Central exchange jump 16-42
Compare collated 16-48; 17-13
Compare/move 16-47, 48
Compare uncollated 16-49; 17-13
Direct read/write of CM 17-28
Direct read/write of UEM 17-28
Error exit to MA 16-66
Floating divide 16-37
Floating double-precision difference 16-32
Floating double-precision product 16-36
Floating double-precision sum 16-29
Floating-point difference 16-31
Floating-point sum 16-28
Floating product 16-34
Form mask 16-68
Illegal 16-66, 67; 17-24
Integer difference 16-8
Integer sum 16-8
Interrupt to executive mode 16-66
Jump 16-40
Left shift 16-19, 20
Logical difference 16-25
Logical product 16-26
Logical sum 16-24
Monitor exchange jump 16-43
Move direct 16-47; 17-13

Move indirect 16-47; 17-13
Normalize 16-63
Pack 16-6
Pass 16-65
Population count 16-68
Read free running counter 16-69
Read word from CM 16-61; 17-28
Read word from UEM 16-67
Return jump 16-39
Right shift 16-21, 22
Round floating difference 16-33
Round floating divide 16-38
Round floating product 16-35
Round floating sum 16-30
Round normalize 16-64
Set Ai 16-51, 52, 53, 54
Set Bi 16-55, 56, 57
Set Xi 16-58, 59, 60
Transfer word from X register 16-44
Transmit complement 16-44
Transmit word 16-44
Unconditional jump 16-39
Unpack 16-7
Write one word to UEM 16-67
Write word to CM 16-61; 17-28
Instructions, PP (see also inside back cover) 16-70
Activate channel 16-104
Add 16-78, 79
Central read 16-96, 97
Central write 16-98, 99
Channel flag 17-34
Clear channel error flag 16-95
Clear channel flag 16-101
CM read 17-31
CM write 17-31
Deactivate channel 16-104
Error flag 17-36
Exchange jump 16-107
Function on channel 16-105
Input from channel 16-102
Jump if channel active 16-94
Jump if channel empty 16-95
Jump if channel error flag clear 16-95
Jump if channel error flag set 16-95
Jump if channel full 16-94
Jump if channel inactive 16-94
Load 16-74, 75, 76
Load complement 16-74
Load/store R register 16-75, 76; 17-29
Logical difference 16-83, 84
Logical product 16-85
Long jump 16-90
Minus jump 16-93
Monitor exchange jump 16-107
Monitor exchange jump to MA 16-107
Nonzero jump 16-92
Output on channel 16-103
Pass 16-106
Plus jump 16-93
Replace add 16-86, 87, 88
Replace add one 16-87, 88
Replace subtract one 16-88
Return jump 16-91
Selective clear 16-82
Shift 16-82
Store 16-76, 77
Store R register 16-76; 17-29
Subtract 16-80, 81
Test and set channel flag 16-101
Unconditional jump 16-91
Zero jump 16-92
Integer arithmetic 17-13
Integer difference instruction 16-5, 8
Integer sum instruction 16-5, 8
Inter-PP communications 17-34
Internal addresses, maintenance registers 17-61, 62, 63
Internal interface, IOU
835 1-9; 2-28
840, 850, and 860 7-10; 8-29
845 and 855 4-11; 5-31
990 10-10; 11-31
990E, 995E, and 994 13-11; 14-31
Interrupt to executive mode instruction 16-66
IOU
Barrel and slot
835 2-24
840, 850, and 860 8-24
845 and 855 5-26
990 11-26
990E, 995E, and 994 14-26
Deadstart
835 2-24
840, 850, and 860 8-24
845 and 855 5-26
990 11-26
990E, 995E, and 994 14-26
Description
835 1-9
840, 850, and 860 7-10
845 and 855 4-11
990 10-10
990E, 995E, and 994 13-11
Environmental control register
835 3-3
840, 850, and 860 9-3
845 and 855 6-3
External interface
835 1-9
840, 850, and 860 7-10
845 and 855 4-11
990 10-10
990E, 995E, and 994 13-11
Functional characteristics
835 1-5
840, 850, and 860 7-6
845 and 855 4-7
990 10-6
990E, 995E, and 994 13-7

Functional description
835 2-23
840, 850, and 860 8-23
845 and 855 5-25
990 11-23
990E, 995E, and 994 14-23
I/O channels
835 1-9
840, 850, and 860 7-10
845 and 855 4-11
990 10-10
990E, 995E, and 994 13-11
Initialization
835 2-23
840, 850, and 860 8-23
845 and 855 5-25
990 11-23
990E, 995E, and 994 14-23
Internal interface
835 1-9
840, 850, and 860 7-10
845 and 855 4-11
990 10-10
990E, 995E, and 994 13-11
Maintenance registers 17-65
MCH function word 17-57
Peripheral processors, see PPs
PPs
835 1-9; 2-23
840, 850, and 860 7-10; 8-23
845 and 855 4-11; 5-25
990 10-10; 11-24
990E, 995E, and 994 13-11; 14-24
Reconfiguration
835 3-11
840, 850, and 860 9-12
845 and 855 6-16
990 12-7
990E, 995E, and 994 15-7

J

Job mode, see CYBER 170 job mode
Jump if channel active instruction 16-94
Jump if channel empty instruction 16-95
Jump if channel error flag clear instruction 16-95
Jump if channel error flag set instruction 16-95
Jump if channel full instruction 16-94
Jump if channel inactive 16-94
Jump instructions 16-40

K

K register, PP
835 2-27
840, 850, and 860 8-27
845 and 855 5-29
990 11-29
990E, 995E, and 994 14-29
Keyboard character codes 17-43

L

L. D. S. ERROR-A indicator
835 3-4
840, 850, and 860 9-4
845 and 855 6-4
L. D. S. ERROR-B indicator
835 3-4
840, 850, and 860 9-4
845 and 855 6-4
LAMP TEST switch
835 3-3
840, 850, and 860 9-3
845 and 855 6-3
Large scale integration (LSI) logic
840, 850, and 860 7-3
845 and 855 4-4
990 10-3
990E, 995E, and 994 13-4
Least significant bit 16-1
Left circular shift 16-19
Left shift instruction 16-19
Load complement instruction 16-74
Load instruction 16-74, 75, 76
Load R register 16-75, 76; 17-29
Lockout time, maximum request
835 2-16
840, 850, and 860 8-16
845 and 855 5-17
990 11-16
990E, 995E, and 994 14-16
Logical difference instruction 16-83, 84
Logical product instruction 16-85
Logical sum instruction 16-24
Long-add instructions
835 2-5
840, 850, and 860 8-5
845 and 855 5-5
990 11-5
990E, 995E, and 994 14-5
Long jump instruction 16-90
LONG/SHORT DEAD START SEQUENCE switch
835 3-4
840, 850, and 860 9-4
845 and 855 6-4
Long/short deadstart sequence
835 3-4
840, 850, and 860 9-4
845 and 855 6-4
990 12-3
990E, 995E, and 994 15-3
Lookahead, instruction
835 2-1
840, 850, and 860 8-1
845 and 855 5-1
990 11-1

990E, 995E, and 994 14-1
Lookahead purge control
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Lookahead, Purge control 17-14
Lookahead purge flag
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
LSI logic
840, 850, and 860 7-3
845 and 855 4-4
990 10-3
990E, 995E, and 994 13-4

M

MA register
835 2-13
840, 850, and 860 8-13
845 and 855 5-13
990 11-13
990E, 995E, and 994 14-13
MAC, see Maintenance access control
Mainframe configuration, see Configuration, system
Maintenance access control
835 2-1, 29
840, 850, and 860 8-1, 30
845 and 855 5-1, 32
990 11-1, 32
990E, 995E, and 994 14-1, 32
Maintenance channel 17-56
Master clear 17-63
Programming 17-56
835 2-29
840, 850, and 860 8-30
845 and 855 5-32
990 11-32
990E, 995E, and 994 14-32
Maintenance channel interface 17-56
Control words 17-61
Function words 17-57
835 1-9; 2-29
840, 850, and 860 7-10; 8-30
845 and 855 4-11; 5-32
990 10-10; 11-32
990E, 995E, and 994 13-11; 14-32
Marker flag, channel 17-36
Mask designator 16-3
Masking word 16-68
Master clear
Maintenance channel 17-58
Two-port multiplexer 17-54
835 3-10
840, 850, and 860 9-11
845 and 855 6-15
990 12-6
990E, 995E, and 994 15-6
MCH, see Maintenance channel
CM reconfiguration
835 2-22; 3-2, 7, 8
840, 850, and 860 8-22; 9-2, 7, 8
845 and 855 5-24; 6-2, 8, 9, 10, 11
990 11-22; 12-5
990E, 995E, and 994 14-22; 15-5
CM reconfiguration switches
835 3-2
840, 850, and 860 9-2
845 and 855 6-2
PP reconfiguration
835 3-11
840, 850, and 860 9-12
845 and 855 6-16
990 12-7
990E, 995E, and 994 15-7
PP reconfiguration switches
835 3-11
840, 850, and 860 9-12
845 and 855 6-16
990 12-8, 12
990E, 995E, and 994 15-8, 12
Programming 17-26
Reconfiguration switches
835 2-22
840, 850, and 860 8-14
845 and 855 5-14
990 11-22
990E, 995E, and 994 14-14
Memory, see CM and PP
Metal oxide semiconductor memory, see MOS
MF, see CYBER 170 monitor flag
Microcode 17-4
Minus jump instruction 16-93
Mode selection bits 17-15
Modes of operation 16-4
Monitor address register, see MA register
Monitor Condition Register Errors 17-25
Monitor exchange jump instruction 16-107
Monitor flag, see CYBER 170 monitor flag
Monitor mode, see CYBER 170 monitor mode
MOS
835 1-4
840, 850, and 860 7-5
845 and 855 4-6
990 10-5
990E, 995E, and 994 13-6
Most insignificant bit 16-1
Move direct instruction 16-47; 17-13
Move indirect instruction 16-47; 17-13

N

No-address instructions 17-30
Nonstandard operands 17-8
Nonzero jump instruction 16-92
Normal jump instruction sequence
  835 2-8
  840, 850, and 860 8-8
  845 and 855 5-8
  990 11-8
  990E, 995E, and 994 14-8
Normalize instruction 16-63
Normalize operations 16-63; 17-10
  835 2-3
  840, 850, and 860 8-3
  845 and 855 5-3
  990 11-3
  990E, 995E, and 994 14-3
Normalized numbers
  Fixed-point 17-12
  Floating-point 17-10

O

Offset designator for
 compare/move 16-45
Ones complement addition/subtraction
  835 2-4
  840, 850, and 860 8-4
  845 and 855 5-4
  990 11-4
  990E, 995E, and 994 14-4
Opcode designator 16-3
Operand designator 16-3
Operand registers, see X registers
Operating instructions/procedures
  835 3-9
  840, 850, and 860 9-10
  845 and 855 6-14
  990 12-6
  990E, 995E, and 994 15-6
Operating modes, CP 16-4
Operating registers
  835 1-6; 2-10
  840, 850, and 860 7-7; 8-10
  845 and 855 4-8; 5-10
  990 10-7; 11-10
  990E, 995E, and 994 13-8; 14-10
Operation code 16-3
Output data parity error 17-36
Output on channel instruction 16-103
Output sequence 17-40
Overflow
  Fixed-point 17-12
  Floating-point 17-7
Overrun 17-51

P

P or Q barrel failure
  835 3-5
  840, 850, and 860 9-5
  845 and 855 6-5
P register, CP
  835 2-11
  840, 850, and 860 8-11
  845 and 855 5-11
  990 11-11
  990E, 995E, and 994 14-11
P register, PP
  835 2-27
  840, 850, and 860 8-27
  845 and 855 5-29
  990 11-29
  990E, 995E, and 994 14-29
Pack instruction 16-6
Pack/unpack instructions 16-6, 7
  835 2-3
  840, 850, and 860 8-3
  845 and 855 5-3
  990 11-3
  990E, 995E, and 994 14-3
Packing numbers 17-5
Panel, deadstart
  835 3-2
  840, 850, and 860 9-2
  845 and 855 6-2
  990 12-1
  990E, 995E, and 994 15-1
Parcels 16-1
  835 2-2
  840, 850, and 860 8-2
  845 and 855 5-2
  990 11-2
  990E, 995E, and 994 14-2
Parity, channel 17-32
Parity errors 17-19, 23
Partial overflow 17-7
Partial underflow 17-7
Pass instruction 16-106
Peripheral processor, see PP
Physical characteristics
  835 1-1
  840, 850, and 860 7-1
  845 and 855 4-1
  990 10-1
  990E, 995E, and 994 13-1
Plus jump instruction 16-93
Population-count instructions 16-68
  835 2-4
  840, 850, and 860 8-4
  845 and 855 5-4
  990 11-4
  990E, 995E, and 994 14-4
Port bounds register, see CM bounds
 register

Port priority
835 2-16
840, 850, and 860 8-16
845 and 855 5-17
990 11-16
990E, 995E, and 994 14-16
Ports, CM
835 1-4; 2-16
840, 850, and 860 7-5; 8-16
845 and 855 4-6; 5-17
990 10-5; 11-16
990E, 995E, and 994 13-6; 14-16
Power-off procedures
835 3-9
840, 850, and 860 9-10
845 and 855 6-14
990 12-6
990E, 995E, and 994 15-6
Power-on procedures
835 3-9
840, 850, and 860 9-10
845 and 855 6-14
990 12-6
990E, 995E, and 994 15-6
PP NO switch
835 3-3
840, 850, and 860 9-3
845 and 855 6-3
PP SELECT switch
835 3-3
840, 850, and 860 9-3
845 and 855 6-3
PPM
835 2-25
840, 850, and 860 8-25
845 and 855 5-27
990 11-27
990E, 995E, and 994 14-27
PPs
Access to CM
835 2-29
840, 850, and 860 8-30
845 and 855 5-32
990 11-34
990E, 995E, and 994 14-35
Addressing 17-29
Barrels
835 1-9; 2-23
840, 850, and 860 7-10; 8-23
845 and 855 4-11; 5-25
990 11-24
990E, 995E, and 994 14-24
CM read instructions 17-31
CM write instructions 17-31
Communications 17-34
835 2-23
840, 850, and 860 8-23
845 and 855 5-25
990 11-24
990E, 995E, and 994 14-24
Data format 16-70
Description
835 2-23
840, 850, and 860 8-23
845 and 855 5-25
990 11-24
990E, 995E, and 994 14-24
Initialization
835 2-23
840, 850, and 860 8-23
845 and 855 5-25
990 11-23
990E, 995E, and 994 14-23
Instruction designators 16-71
Instruction execution times 16-108
Instruction word format 16-70
Instructions 16-70
Memory failure
835 3-5
840, 850, and 860 9-5
845 and 855 6-5
Memory (PPM)
835 2-23
840, 850, and 860 8-23
845 and 855 5-25
990 11-24
990E, 995E, and 994 14-24
Numbering
835 2-28
840, 850, and 860 8-28
845 and 855 5-30
990 11-30
990E, 995E, and 994 14-30
Program timing considerations 17-47
Programming 17-47
Reconfiguration
835 2-28; 3-11
840, 850, and 860 8-28; 9-12
845 and 855 5-30; 6-16
990 11-30; 12-8, 12
990E, 995E, and 994 14-30; 15-8, 12
Reconfiguration switches
835 3-11
840, 850, and 860 9-12
845 and 855 6-16
990 12-8, 12
990E, 995E, and 994 15-8, 12
Registers
835 2-26
840, 850, and 860 8-26
845 and 855 5-28
990 11-28
990E, 995E, and 994 14-28
Relocation register format 16-73
To PP communications 17-34
Write output buffer function 17-52
835 1-9
840, 850, and 860 7-10
845 and 855 4-11
990 10-10
990E, 995E, and 994 13-11

Prefetch of instructions, see Instruction lookahead
Processor-Detected malfunctions 17-25
Program address register, see P register
Program mode 16-4
Program timing considerations, PP 17-34
Program word format 16-1
Programming
  CM 17-26
  CP 17-1
  Display station 17-42
  Maintenance channel 17-56
  PP 17-29
  Real-time clock 17-49
  Two-port multiplexer 17-49
Programming information 17-1
Publication index, see System publication index
Purge control 17-14, 15

Q

Q barrel failure
  835 3-5
  840, 850, and 860 9-5
  845 and 855 6-5
Q register, PP
  835 2-27
  840, 850, and 860 8-27
  845 and 855 5-29
  990 11-29
  990E, 995E, and 994 14-29
Quadrant select field
  835 2-15
  840, 850, and 860 8-15
  845 and 855 5-15
Queuing buffers
  835 1-8
  840, 850, and 860 7-9
  845 and 855 4-10
  990 10-9
  990E, 995E, and 994 13-10

R

R register
  Load instruction 17-29
  Store instruction 17-29
  835 2-26
  840, 850, and 860 8-26
  845 and 855 5-28
  990 11-28
  990E, 995E, and 994 14-28
RAC 17-3
RAC register
  835 2-11
  840, 850, and 860 8-11
  845 and 855 5-11
  990 11-11
  990E, 995E, and 994 14-11
RAE register
  835 2-13
  840, 850, and 860 8-13
  845 and 855 5-13
  990 11-13
  990E, 995E, and 994 14-13
Ranks
  835 2-25
  840, 850, and 860 8-25
  845 and 855 5-27
  990 11-27
  990E, 995E, and 994 14-27
Read free running counter instruction 16-69
Read maintenance registers 17-58, 60
Read-only memory
  835 3-3
  840, 850, and 860 9-3
  845 and 855 6-3
  990 12-6
  990E, 995E, and 994 15-6
Read status summary 17-51
Read status summary function 17-51
Read terminal data function 17-51
Read word from CM instruction 16-61; 17-28
Read word from UEM instruction 16-61; 17-28
Real-time clock interface
  Programming 17-49
  835 1-9; 2-29
  840, 850, and 860 7-10; 8-30
  845 and 855 4-11; 5-32
  990 10-10; 11-31
  990E, 995E, and 994 13-11; 14-31
Reconfiguration,
  CM
    835 2-22; 3-2, 7, 8
    840, 850, and 860 8-22; 9-2, 7, 8
    845 and 855 5-24; 6-2, 8, 9, 10, 11
    990 11-22; 12-5
    990E, 995E, and 994 14-22; 15-5
  PPs
    835 2-28; 3-11
    840, 850, and 860 8-28; 9-12
    845 and 855 5-30; 6-16
    990 11-30; 12-8, 12
    990E, 995E, and 994 14-30; 15-8, 12
RECONFIGURATION, BARREL switch
  835 3-4
  840, 850, and 860 9-4
  845 and 855 6-4
RECONFIGURATION, PPM switch
  835 3-4
  840, 850, and 860 9-4
  845 and 855 6-4
Reference address for CM register, see RAC register

Reference address for UEM register, see RAE register
Register designator 16-4
Register display, PP
835 3-3
840, 850, and 860 9-3
845 and 855 6-3
Register empty 17-35
Register formats 16-1
Register full flag 17-35
REGISTER SELECT switch
835 3-3
840, 850, and 860 9-3
845 and 855 6-3
Registers,
A, CP 16-51
835 1-3, 6; 2-10
840, 850, and 860 7-4, 7; 8-10
845 and 855 4-5, 8; 5-10
990 10-4, 7; 11-10
990E, 995E, and 994 13-5, 8; 14-10
A, PP
835 2-26
840, 850, and 860 8-26
845 and 855 5-28
990 11-28
990E, 995E, and 994 14-28
B 16-55
835 1-3, 6; 2-10
840, 850, and 860 7-4, 7; 8-10
845 and 855 4-5, 8; 8-10
990 10-4, 7; 11-10
990E, 995E, and 994 13-5, 8; 14-10
Compare/move support
835 2-11
840, 850, and 860 8-11
845 and 855 5-11
990 11-11
990E, 995E, and 994 14-11
CP
835 1-6; 2-9
840, 850, and 860 7-7; 8-9
845 and 855 4-8; 5-9
990 10-7; 11-9
990E, 995E, and 994 13-8; 14-9
EM
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Exit mode, see EM register
Field length, see FLC and FLE registers
Flag
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
FLC
835 2-11
840, 850, and 860 8-11
845 and 855 5-11
990 11-11
990E, 995E, and 994 14-11
FLE
835 2-13
840, 850, and 860 8-13
845 and 855 5-13
990 11-13
990E, 995E, and 994 14-13
General description
835 1-6
840, 850, and 860 7-7
845 and 855 4-8
990 10-7
990E, 995E, and 994 13-8
K
835 2-27
840, 850, and 860 8-27
845 and 855 5-29
990 11-29
990E, 995E, and 994 14-29
MA
835 2-13
840, 850, and 860 8-13
845 and 855 5-13
990 11-13
990E, 995E, and 994 14-13
Monitor address register, see MA register
Operating
835 1-6; 2-10
840, 850, and 860 7-7; 8-10
845 and 855 4-8; 5-10
990 10-7; 11-10
990E, 995E, and 994 13-8; 14-10
P, CP
835 2-11
840, 850, and 860 8-11
845 and 855 5-11
990 11-11
990E, 995E, and 994 14-11
P, PP
835 2-27
840, 850, and 860 8-27
845 and 855 5-29
990 11-29
990E, 995E, and 994 14-29
PP
835 2-26
840, 850, and 860 8-26
845 and 855 5-28
990 11-28
990E, 995E, and 994 14-28
Program address, see P register
Q
835 2-26
840, 850, and 860 8-26
845 and 855 5-28

990 11-28
990E, 995E, and 994 14-28
R
835 2-26
840, 850, and 860 8-26
845 and 855 5-28
990 11-28
990E, 995E, and 994 14-28
RAC
835 2-11
840, 850, and 860 8-11
845 and 855 5-11
990 11-11
990E, 995E, and 994 14-11
RAE
835 2-13
840, 850, and 860 8-13
845 and 855 5-13
990 11-13
990E, 995E, and 994 14-13
Reference address, see RAC and RAE registers
Support
835 1-6; 2-11
840, 850, and 860 7-7; 8-11
845 and 855 4-8; 5-11
990 10-7; 11-11
990E, 995E, and 994 13-8; 14-11
X 16-58
835 1-6; 2-10
840, 850, and 860 7-7; 8-10
845 and 855 4-8; 5-10
990 10-7; 11-10
990E, 995E, and 994 13-8; 14-10
835 2-9
840, 850, and 860 8-9
845 and 855 5-9
990 11-9
990E, 995E, and 994 14-9
Relative address 17-26
Relocation register, see R register
Replace add instruction 16-86, 87, 88
Replace add one instruction 16-87, 88
Replace subtract one instruction 16-89
Request lockout time, maximum
835 2-16
840, 850, and 860 8-16
845 and 855 5-17
990 11-16
990E, 995E, and 994 14-16
Request to send 17-54
Reservation, channel 17-36
Return jump instruction 16-91
Return jump instruction sequence
835 2-8
840, 850, and 860 8-8
845 and 855 5-8
990 11-8
990E, 995E, and 994 14-8
Right shift instruction 16-21, 22
ROM, see Read-only memory
Round floating difference instruction 16-33
Round floating divide instruction 16-38
Round floating product instruction 16-35
Round floating sum instruction 16-30
Round normalize instruction 16-64
Rounding, floating-point 17-10
Rounding operation 16-64; 17-10
Row select field
835 2-15
840, 850, and 860 8-15
845 and 855 5-15, 16
990 11-15
RTS, see Request to send
Running counter 16-4

S

S register, see Status register
SECDED
835 2-17
840, 850, and 860 8-17
845 and 855 5-19
990 11-17
990E, 995E, and 994 14-17
SECDED code bits
Description
835 2-18
840, 850, and 860 8-18
845 and 855 5-20
990 11-18
990E, 995E, and 994 14-18
SECDED errors 17-19, 23
Selective clear instructions 16-82
Selector 17-56
Set A register instructions
835 2-5
840, 850, and 860 8-5
845 and 855 5-5
990 11-5
990E, 995E, and 994 14-5
Set Ai instructions 16-51, 52, 53, 54
Set B register instructions
835 2-5
840, 850, and 860 8-5
845 and 855 5-5
990 11-5
990E, 995E, and 994 14-5
Set Bi instructions 16-55, 56, 57
Set X register instructions
835 2-5
840, 850, and 860 8-5
845 and 855 5-5
990 11-5
990E, 995E, and 994 14-5
Set Xi instructions 16-58, 59, 60
Shift designator 16-3
Shift instruction 16-82

Shift instruction sequence
835 2-3
840, 850, and 860 8-3
845 and 855 5-3
990 11-3
990E, 995E, and 994 14-3
Short deadstart sequence
835 3-4
840, 850, and 860 9-4
845 and 855 6-4
990 12-3
990E, 995E, and 994 15-3
Single-bit errors
835 2-17
840, 850, and 860 8-17
845 and 855 5-19
990 11-17
990E, 995E, and 994 14-17
Single error correction/double error detection, see SECDED
Single-precision 17-11
Slot, see Barrel and slot
Slow deadstart
835 3-4
840, 850, and 860 9-4
845 and 855 6-4
Software errors, 17-25
SS register, see Status summary register
Standard addressing mode 17-3, 8
835 STA
Store instruction 16-76, 77
Store R register 16-76; 17-29
Subtract instruction 16-80, 81
Support registers
Compare/move
835 2-6
840, 850, and 860 8-6
845 and 855 5-6
990 11-6
990E, 995E, and 994 14-6
835 1-6; 2-11
840, 850, and 860 7-7; 8-11
845 and 855 4-8; 5-11
990 10-7; 11-11
990E, 995E, and 994 13-8; 14-11
Switches,
CM reconfiguration
835 3-2
840, 850, and 860 9-2
845 and 855 6-2
990E, 995E, and 994 15-4
PP reconfiguration
835 3-11
840, 850, and 860 9-12
845 and 855 6-16
990 12-8, 12
990E, 995E, and 994 15-8, 12
Syndrome codes
835 2-18
840, 850, and 860 8-18
845 and 855 5-20
990 11-18
990E, 995E, and 994 14-18
System description
835 1-1
840, 850, and 860 7-1
845 and 855 4-1
990 10-1
990E, 995E, and 994 13-1

T

Target address
835 2-1
840, 850, and 860 8-1
845 and 855 5-1
990 11-1
990E, 995E, and 994 14-1
Terminal deselect function 17-50
Terminal operation mode function 17-53
Terminal select function 17-50
Test and set channel flag instruction 16-101
Timing considerations
Channel 17-36
Display station 17-47
PP 17-47
Transfer blocks 17-28
Transfer single words 17-28
Transfer timing 17-36, 37
Transfer word from X register 16-44
Transfers, I/O, see I/O transfers
Transmit complement instruction 16-44
Transmit word instruction 16-44
Two-port multiplexer interface
Input data 17-55
Operation 17-50
Output data 17-55
Programming 17-49
835 1-9; 2-29
840, 850, and 860 7-10; 8-30
845 and 855 4-11; 5-32
990 10-10; 11-31
990E, 995E, and 994 13-11; 14-32

U

UART 17-52
UEL register, see Uncorrected error log register
UEM
Block copy instructions 16-9
Description
835 1-8
840, 850, and 860 7-5
845 and 855 4-10
990 10-9

990E, 995E, and 994 13-10
Direct read/write instructions 17-28
Enable flag
835 2-12
840, 850, and 860 8-12
845 and 855 5-12
990 11-12
990E, 995E, and 994 14-12
Field length register, see FLE register
Read one word from 17-28
Reference address register, see RAE register
Write one word to 17-28
Unconditional jump instruction 16-39
Underflow, floating-point 17-7
Unified extended memory, see UEM
Universal Asynchronous Receiver and Transmitter, see UART
Unpack instruction 16-7
Unpack instructions
835 2-3
840, 850, and 860 8-3
845 and 855 5-3
990 11-3
990E, 995E, and 994 14-3

V

Virtual State 16-4
835 2-22
840, 850, and 860 8-22
845 and 855 5-24
990 11-22
990E, 995E, and 994 14-22

W

Word, Instruction 16-1
Write maintenance registers 17-62
Write output buffer 17-52
Write word to CM instruction 16-61; 17-28
Write word to UEM instruction 16-67

X

X registers
Set instructions 16-58, 59, 60
835 2-5
840, 850, and 860 8-5
845 and 855 5-5
990 11-5
990E, 995E, and 994 14-5
835 1-6
840, 850, and 860 7-7
845 and 855 4-8
990 10-7
990E, 995E, and 994 13-8

Z

Zero jump instruction 16-92

0

00 instruction
Error exit 17-17, 20

1

12/16 conversion failure
835 3-5
840, 850, and 860 9-5
845 and 855 6-5

Comments (continued from other side)

Please fold on dotted line;
seal edges with tape only.

FOL

FOLD

FOLI

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
First-Class Mail Permit No. 8241 Minneapolis, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA
Technical Publications
ARH219
4201 N. Lexington Avenue
Arden Hills, MN 55126-9983

We would like your comments on this manual to help us improve it. Please take a few minutes to fill out this form.

Who are you?

- ☐ Manager
- ☐ Systems analyst or programmer
- ☐ Applications programmer
- ☐ Operator
- ☐ Other _____

How do you use this manual?

- ☐ As an overview
- ☐ To learn the product or system
- ☐ For comprehensive reference
- ☐ For quick look-up
- ☐ Other _____

What programming languages do you use? _____

How do you like this manual? Answer the questions that apply.

Yes	Somewhat	No	
☐	☐	☐	Does it tell you what you need to know about the topic?
☐	☐	☐	Is the technical information accurate?
☐	☐	☐	Is it easy to understand?
☐	☐	☐	Is the order of topics logical?
☐	☐	☐	Can you easily find what you want?
☐	☐	☐	Are there enough examples?
☐	☐	☐	Are the examples helpful? (☐ Too simple? ☐ Too complex?)
☐	☐	☐	Do the illustrations help you?
☐	☐	☐	Is the manual easy to read (print size, page layout, and so on)?
☐	☐	☐	Do you use this manual frequently?

Comments? If applicable, note page and paragraph. Use other side if needed.

Check here if you want a reply: ☐

~~Name~~ _____ ~~Company~~ _____

~~Address~~ _____ ~~Date~~ _____

_____ ~~Phone~~ _____

Please send program listing and output if applicable to your comment.

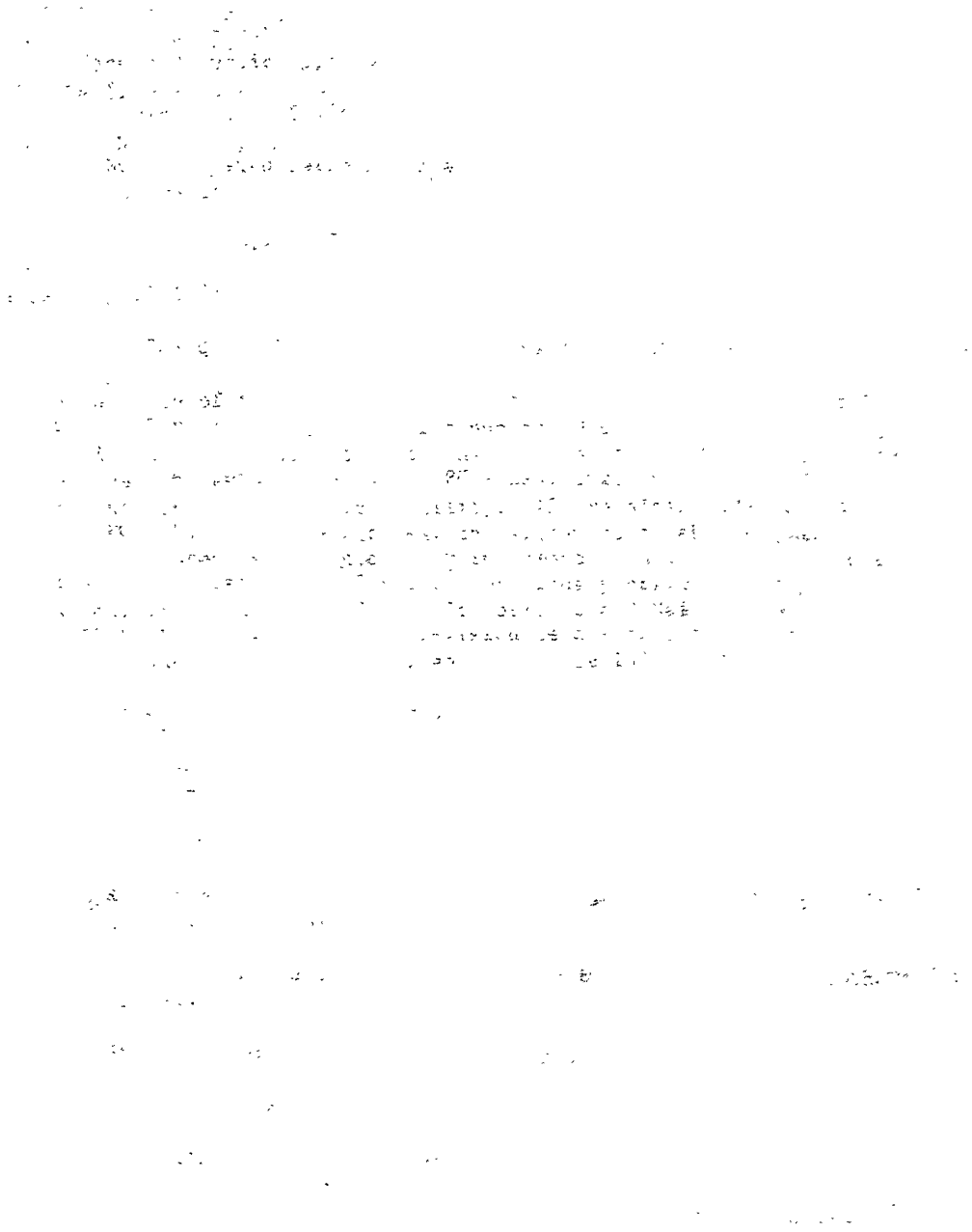
Peripheral Processor Instruction Index

ACN 16-104
ADC 16-78
ADD 16-79
ADI 16-79
ADM 16-79
ADN 16-78
AJM 16-94
AOD 16-87
AOI 16-87
AOM 16-88
CCF 16-101
CFM 16-95
CRD 16-96
CRM 16-97
CWD 16-98
CWM 16-99
DCN 16-104
EJM 16-95
EXN 16-107
FAN 16-105
FJM 16-94
FNC 16-105
IAM 16-102
IAN 16-102
IJM 16-94
KPT 16-106
LCN 16-74
LDC 16-75
LDD 16-75
LDI 16-76
LDM 16-76
LDN 16-74
LJM 16-90
LMC 16-83
LMD 16-83
LMI 16-84
LMM 16-84
LMN 16-83
LPC 16-85
LPN 16-85
LRD 16-75
MAN 16-107
MJN 16-93
MXN 16-107
NJN 16-92
OAM 16-103
OAN 16-103
PJN 16-93
PSN 16-106
RAD 16-86
RAI 16-87
RAM 16-88
RJM 16-91
SBD 16-80
SBI 16-80
SBM 16-81
SBN 16-80
SCF 16-101
SCN 16-82
SFM 16-95
SHN 16-82
SOD 16-89
SOI 16-89
SOM 16-89
SRD 16-76
STD 16-77
STI 16-77
STM 16-77
UJN 16-91
ZJN 16-92

CORPORATE HEADQUARTERS, P.O. BOX O, MINNEAPOLIS, MINN 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A

CONTROL DATA



CYBER 170 Computer Systems Models 835, 845, and 855
CYBER 180 Computer Systems Models 835, 840, 845, 850, 855, 860, and 990
CYBER 990E, 995E, and 994
CYBER 170 State
Hardware Reference Manual
60469290 H