IOS ARCHITECTURE
TR-IA
Software Training Manual
Cray Research, Inc.

July, 1987

This manual is not for further distribution without written approval from the nearest CRAY RESEARCH, INC., regional or country sales office.

Each time this workbook is revised and reprinted, all changes issued against the previous revision in the form of handouts and/or change packets are incorporated into the new version and the new version is assigned a numeric level.

Requests for copies of Cray Research,Inc. software training materials, and comments about these publications should be directed to:

Cray Research, Inc.
2520 Pilot Knob Road
Suite 300
Mendota Heights MN 55120

A March, 1984 -- Original Printing

B September, 1986 -- Reprint with reorganization and revision. Changed to include the new features available on the Model C IOS; reorganized to follow more closely the information as it is discussed in the HR-0081, IOS Model C Hardware Reference Manual.

C July, 1987 -- Reprint with minor revisions. Changed previous references to publication as TNG-IOSAW to TR-IA. Clarified information on interprocessor communication channels on p. 3-20. Updated Buffer Memory access time from 40 cp to 26 cp on pp. 5-2, 5-6. Also, removed all references to this book being "Part I" of some series of IOS self study texts. Added trademark information, and designated this to be a Cray Private document.

CONTENTS

I/O Subsystem Architecture

LIST OF FIGURES

1. INTRODUCTION

Welcome to Cray Research Software Training. This training module is a self-contained package of learning materials that may be used in a self-paced self-instruction learning mode or as support material for a conventional classroom training environment. The module contains a technical presentation, review questions and related reading assignments.

2. PURPOSE

The purpose of this module is to familiarize the learner with the main functions, structure and components of the I/O Subsystem.

3. PREREQUISITES

The following prerequisites are highly recommended for the successful completion of this training module.

1. General knowledge of assembly language and computer architecture.

2. Working knowledge of JCL, CAL, and Cray Systems.

4. MATERIALS

HR-0081 IOS Model C Hardware Reference Manual
HR-0077 Disk Systems Hardware Reference Manual
SQ-0059 IOS Instruction Reference Card

5. OBJECTIVES

Upon completion of this module, a learner with the aid of reference material should be capable of:

1. Listing the basic functions of the I/O Subsystem
2. Listing the basic components of the I/O Subsystem and their specific functions
3. Listing an I/O Processor's components and their functions at the block diagram level
4. Listing the basic differences between the two types of I/O channels - accumulator and DMA

CHAPTER 1
I/O SUBSYSTEM ARCHITECTURE OVERVIEW

ARCHITECTURE OVERVIEW

The I/O Subsystem handles all the physical I/O for the Cray.

1.1 FUNCTIONS

The main functions of the I/O Sybsystem are:

1. Increases Cray CPU throughput by reducing its I/O responsibilities, thereby reducing the interrupt load on the mainframe
   - -Drives up to 48 disk drives
   - -Drives front-end channels
   - -Provides CPU with data using 100 MBYTE/S channel(s)
2. Provides access to additional peripherals
   - -Drives IBM plug compatible devices -- on-line tapes
   - -Drives up to 8 million words of fast secondary storage (Buffer Memory)
3. Functions as a Maintenance Control Unit
   - -Deadstarts and deaddumps the Cray CPU
   - -Provides for operator control of CPU
   - -Drives expander peripherals for maintenance purposes

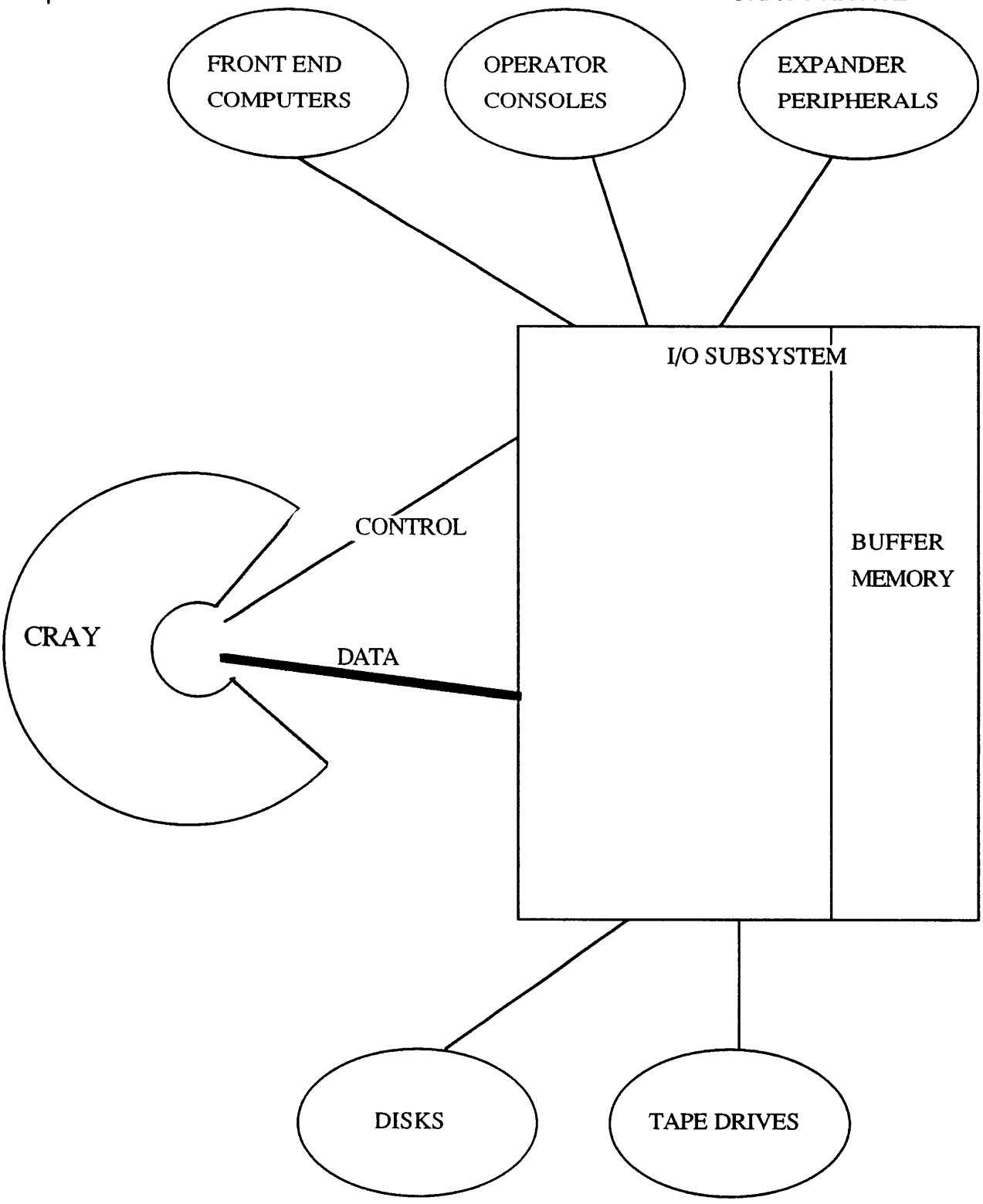


Figure I.1-1: I/O Subsystem Overview

1.2 MAIN COMPONENTS

The I/O Subsystem's four-column chassis contains up to four I/O Processors.* Each I/O Processor is a 16-bit mini-computer. They all share a common memory, Buffer Memory, which has a capacity of 2, 4 or 8 million 64-bit words.

Front-end computers, tapes, disks and expander peripherals are connected directly to the I/O Subsystem.

There are up to four high-speed memory channels between the I/O Subsystem and the Cray for data transmission, with each I/O Processor capable of supporting one high-speed channel. In addition, there is one low-speed communication channel for I/O control and status reporting.

---

* A distinction will be made throughout IOS modules between the terms, "I/O Subsystem" which refers to the entire system, and "I/O Processor" which refers to an individual 16-bit computer contained within the I/O Subsystem.

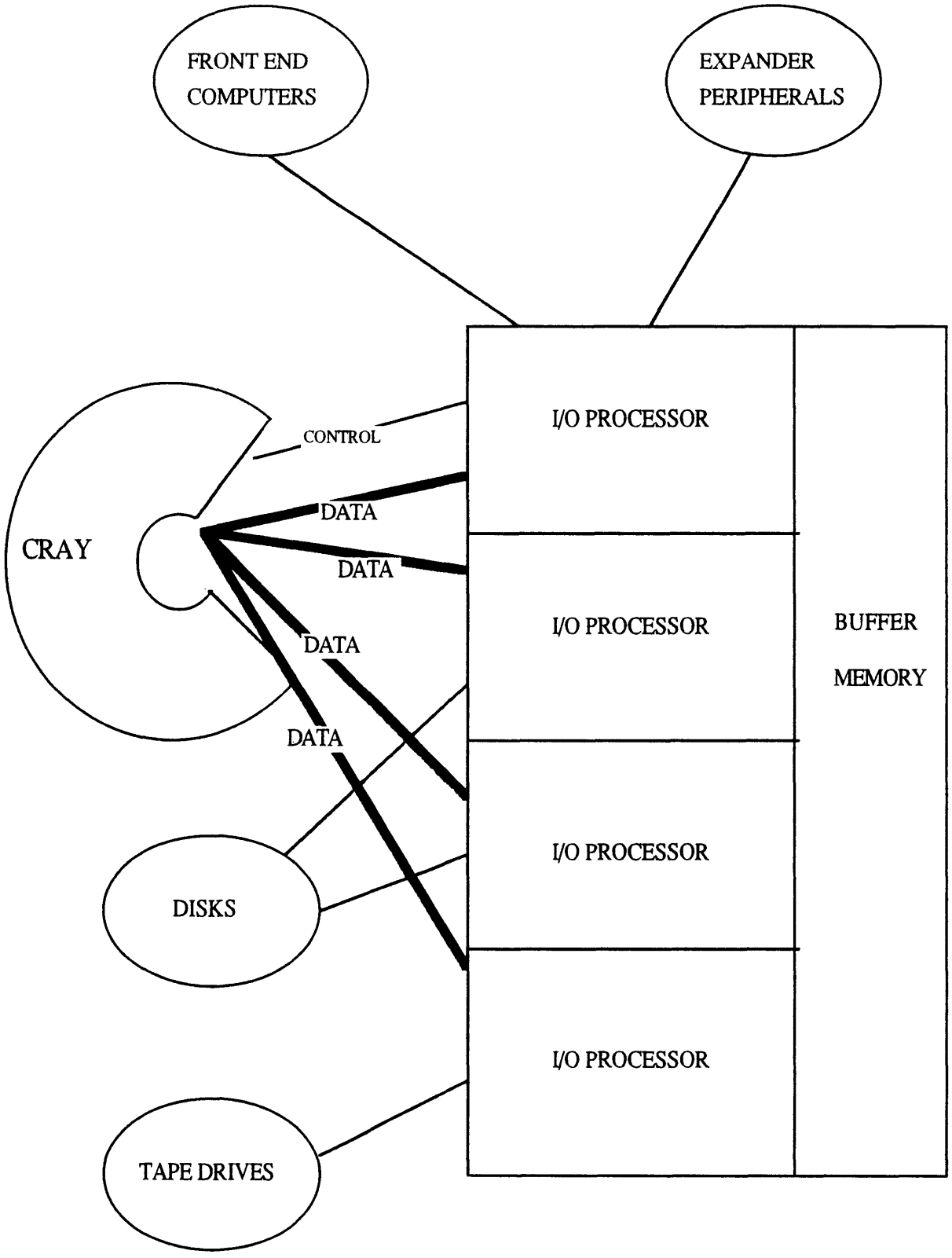


Figure I.1-2: I/O Subsystem Main Components

There are two to four I/O Processors within an I/O Subsystem. They all connect to Buffer Memory through DMA channels. They also communicate with each other over accumulator channels.

TWO PROCESSOR SYSTEM

The minimum configuration within an I/O Subsystem is two I/O Processors. The Master I/O Processor and the Buffer I/O Processor are required in all systems.

1. Master I/O Processor (MIOP or IOP-0)

   The MIOP is the supervisor of all I/O activities. It coordinates the actions of the CPU and other IOP's. All communication with the CPU is over a low-speed channel pair attached to the MIOP. Two to four consoles are connected to the MIOP for operator control. The MIOP supports up to seven front-end interfaces and/or NSC adapaters for multiple front-end computers. On the Expander Chassis (a CDC peripheral expander) there is a printer/plotter used mainly for maintenance, a tri-density tape drive supporting 800, 1600 or 6250 bpi, and an 80 MBYTE disk which is used for deadstarting. MIOP may also support a 100 MBYTE/S channel for transporting data to or from Cray central memory.

2. Buffer I/O Processor (BIOP or IOP-1)

   One of BIOP's main functions is to handle data transfers between the CPU and the I/O Subsystem. The BIOP moves data to or from Cray Central Memory via a 100 MBYTE/S channel. In addition, the BIOP can drive up to 16 disks.

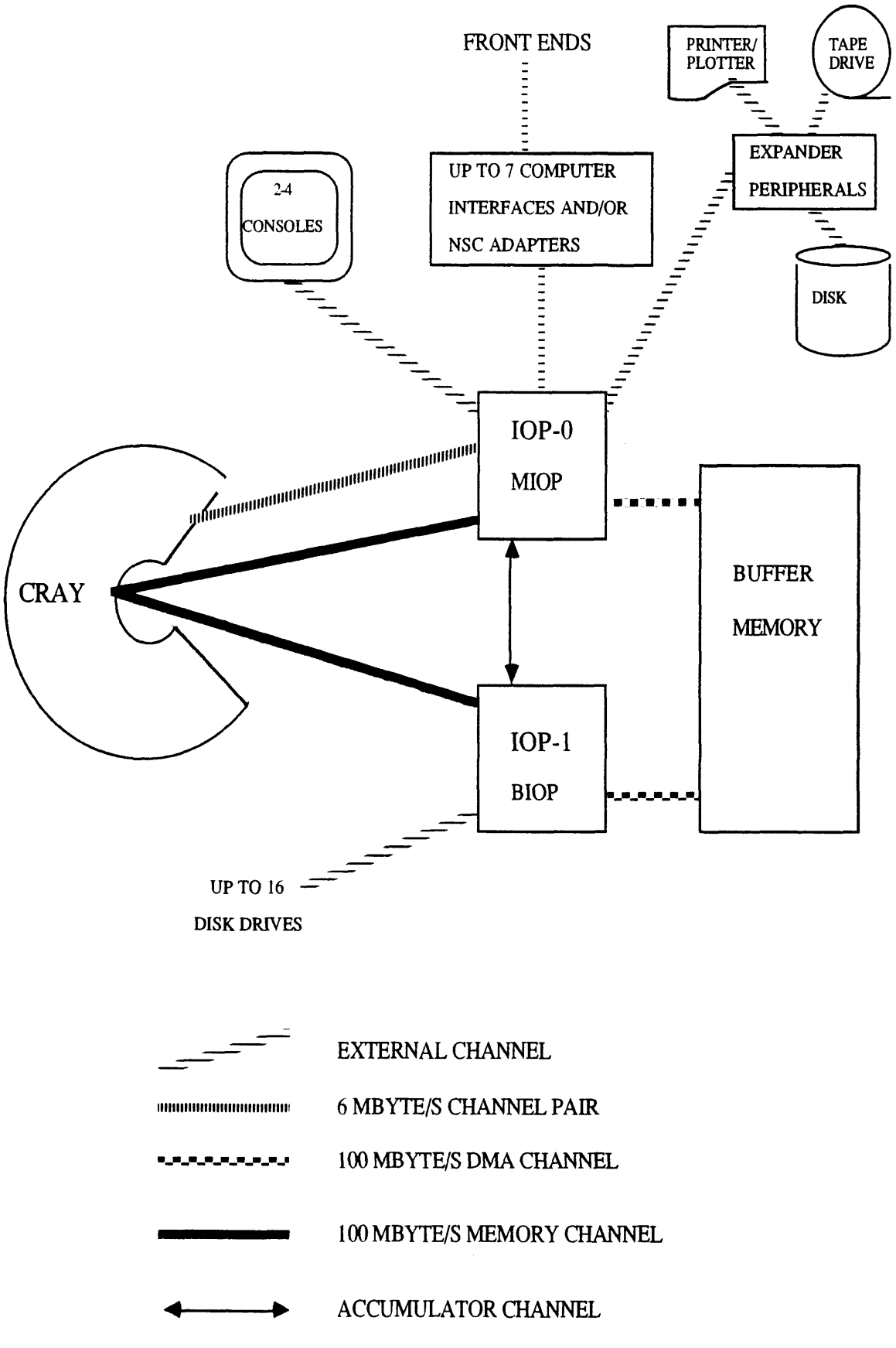


Figure I.1-3: Two Processor System

1.4 THREE PROCESSOR SYSTEM

There are two possible configurations in a three processor system. The third IOP can be either a Disk IOP (for additional disk drives) or an Auxiliary IOP (for on-line tapes).

1.5 FOUR PROCESSOR SYSTEM

In a four processor system, there are three possible configurations. With two DIOP's, up to 48 disk drives are possible. With one DIOP and one XIOP, a maximum of 32 disk drives and 12 channels for tapes are possible. With two XIOP's, up to 24 tape channels, and 16 disk drives are possible.

3. Disk I/O Processor (DIOP, IOP-2, and/or IOP-3)

   The DIOP performs disk I/O to and from disk units attached to the DIOP's channels. Up to 16 disk drives may be attached to the DIOP. DIOP may support a 100 MBYTE/S channel for transporting data to or from Cray Central Memory.

4. Auxiliary I/O Processor (XIOP, IOP-2 and/or IOP-3)

   The XIOP handles 1 to 12 block multiplexor channels for IBM compatible devices. Software is currently available for tape drives on these channels. XIOP may support a 100 MBYTE/S channel for transporting data to or from Cray Central Memory.

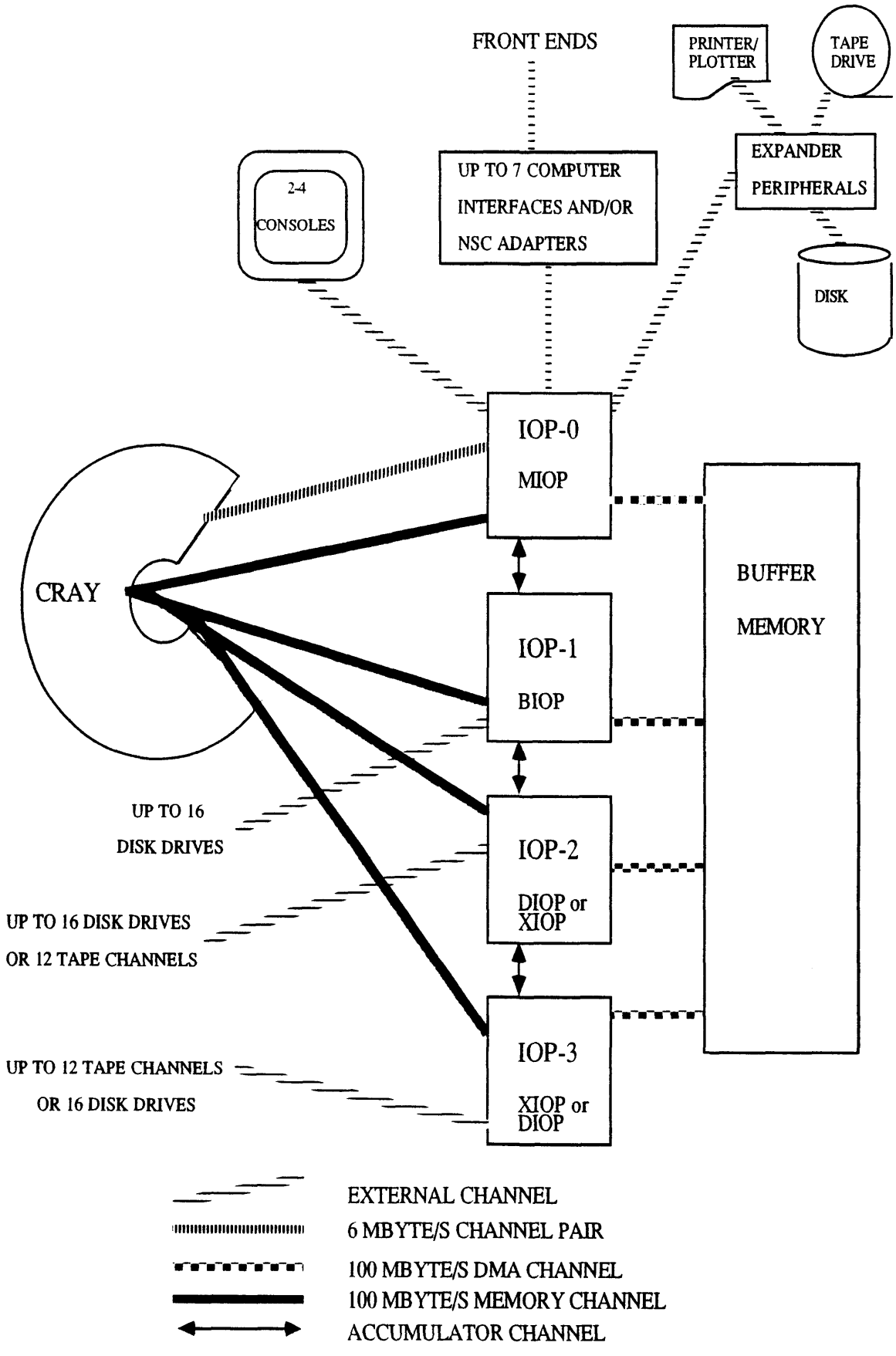


Figure I.1-4: Four Processor System

1.7 INDIVIDUAL I/O PROCESSOR

In general, all individual I/O Processors are identical. They each contain a memory section, computation section and an I/O section.

LOCAL MEMORY

An IOP's memory is referred to as Local Memory to distinguish it from Buffer Memory. Local Memory has a capacity of 65,536 16-bit words. The 16-bit "words" are to be referred to as "parcels" to avoid confusion with the 64-bit words in the CPU and in Buffer Memory.

COMPUTATION SECTION

Within the Computation Section there is an Instruction Control Network for fetching and issuing of all instructions. An IOP can add, subtract, left shift and right shift with the Adder and Shifter functional units. A logical "AND" operation is provided. There are 512 operand registers which are used for temporary data storage and indirect memory addressing. An IOP operates in single-address mode; instructions move data from a source to the accumulator, and from the accumulator to a destination.

I/O SECTION

An IOP has six direct memory access (DMA) ports used for high volume I/O. Several channels may multiplex into one port; e.g., four disk channels can transfer data through one DMA port. The maximum transfer rate per port is 853 MBITS/S.

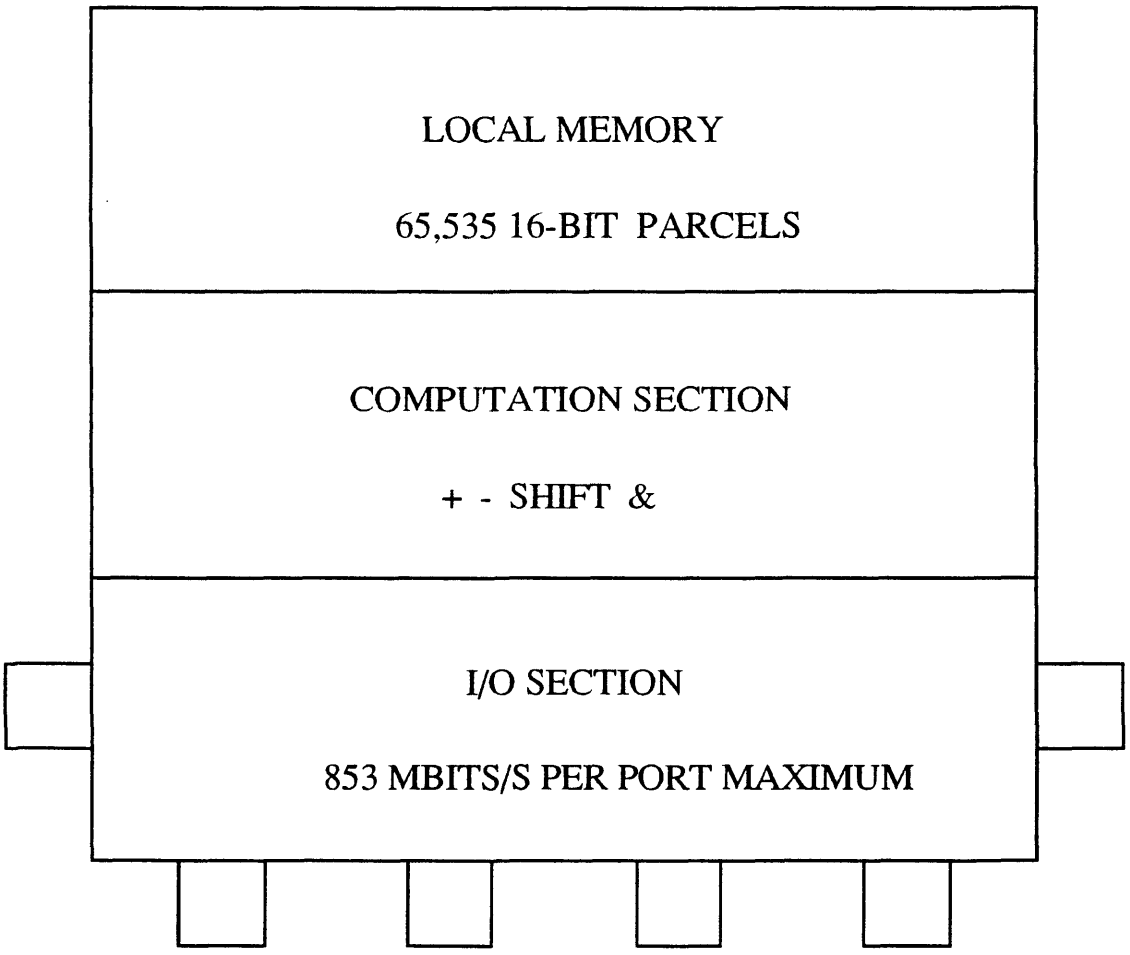


Figure I.1-5: I/O Processor Overview

CHAPTER 1
REVIEW QUESTIONS

1. List two major functions of the I/O Subsystem.
   a. Handles to I/o For a Cray System increasing the CPU throughput by reducing CPU Interrupts
   b. MCU

2. There are 65 536 memory locations in Local Memory and each consists of 16 bits.

3. Name three operations performed by the computation section.
   a. ADD & SUBTRACT
   b. RIGHT & LEFT SHIFT
   c. AND (LOGICAL Product)

4. A 4 processor IOS has:

   24 Direct Memory Access (DMA) Ports into Local Memory
   2 M, 4 M, or 8 M Words of Buffer Memory

5. List two functions of the Master I/O Processor (IOP-0)
   a. Drives Expander Channel
   b. Handles up to 4 consoles

6. What are the Buffer I/O Processor's (IOP-1) main functions?
   Transfer DATA Between IOS & Cray Memory (100 MBYTE Channel)
   HANDLE UP TO 16 Disk DRIVES

7. What are the Auxiliary I/O Processor's (IOP-3) main functions?
   Handle up to 12 IBM COMPATABLE devices. (ON LINE TAPES)

CHAPTER 1
ANSWERS TO REVIEW QUESTIONS

1. The major functions of the I/O Subsystem are:
   a. Increases Cray CPU throughput by reducing its I/O responsibilities, thereby reducing the interrupt load on the mainframe
   b. Provides access to additional peripherals
   c. Functions as a maintenance control unit

2. There are 65,536 memory locations in Local Memory and each consists of 16 bits.

3. Three operations performed by the computation section are:
   a. Add/Subtract
   b. Shift
   c. Logical product ("AND")

4. A 4 processor IOS has:

   24 Direct Memory Access (DMA) Ports into Local Memory (6 per I/O Processor)
   2M, 4M or 8M Words of Buffer Memory

5. The Master I/O Processor has several functions:
   - Coordinates actions of the CPU and other IOPs
   - Communicates with CPU
   - Has up to 4 display consoles for operator control
   - Drives front-end channels
   - Drives expander channel

6. The Buffer I/O Processor's main functions are to transfer data between Central Memory and the IOS system over the 100 MBYTE channel, and to support up to 16 disk drives.

7. The Auxiliary I/O Processor's main functions are to drive up to 12 Block Multiplexor channels for IBM-compatible devices, and to transfer data to the Cray. On-line tapes are currently the only IBM-compatible devices supported by Cray software on these channels.

CHAPTER 2
COMPUTATION SECTION

COMPUTATION SECTION

2.1 BASIC COMPONENTS

The computation section of an I/O Processor has the following components.

1. An Instruction Control Network for all fetching and issuing of instructions.
2. 512 Operand Registers which are used to hold data and addresses.
3. Two Functional Units; an Adder and a Shifter.
4. A Logical "AND" operation.
5. One programmer-visible Accumulator; operates in single address mode.
6. 128 Instruction Codes.

BASIC COMPONENTS

* INSTRUCTION CONTROL NETWORK

* 512 OPERAND REGISTERS

* 2 FUNCTIONAL UNITS

* LOGICAL PRODUCT

* ACCUMULATOR

* 128 INSTRUCTION CODES

Figure I.2-1: Computation Section Basic Components

2.2 INSTRUCTION CONTROL NETWORK

The Instruction Control Network is responsible for controlling issue and execution instructions. The following components make up the Instruction Control Network.

1. Instruction Stack

   The Instruction Stack holds instructions before issuing. Instructions are fetched in 4-parcel bursts from Local Memory to the Instruction Stack.

2. Instruction Issue (II) Register

   Instructions are brought from the Instruction Stack to the Instruction Issue Register one parcel at a time for issue.

3. B Register

   The B Register holds program-modifiable data.

4. P Register

   The Program Address (P) Register holds the address of the current instruction in the II Register.

5. Program Exit Stack

   The Exit Stack holds subroutine return addresses, interrupted program addresses, and the Kernel Interrupt Handler starting address.

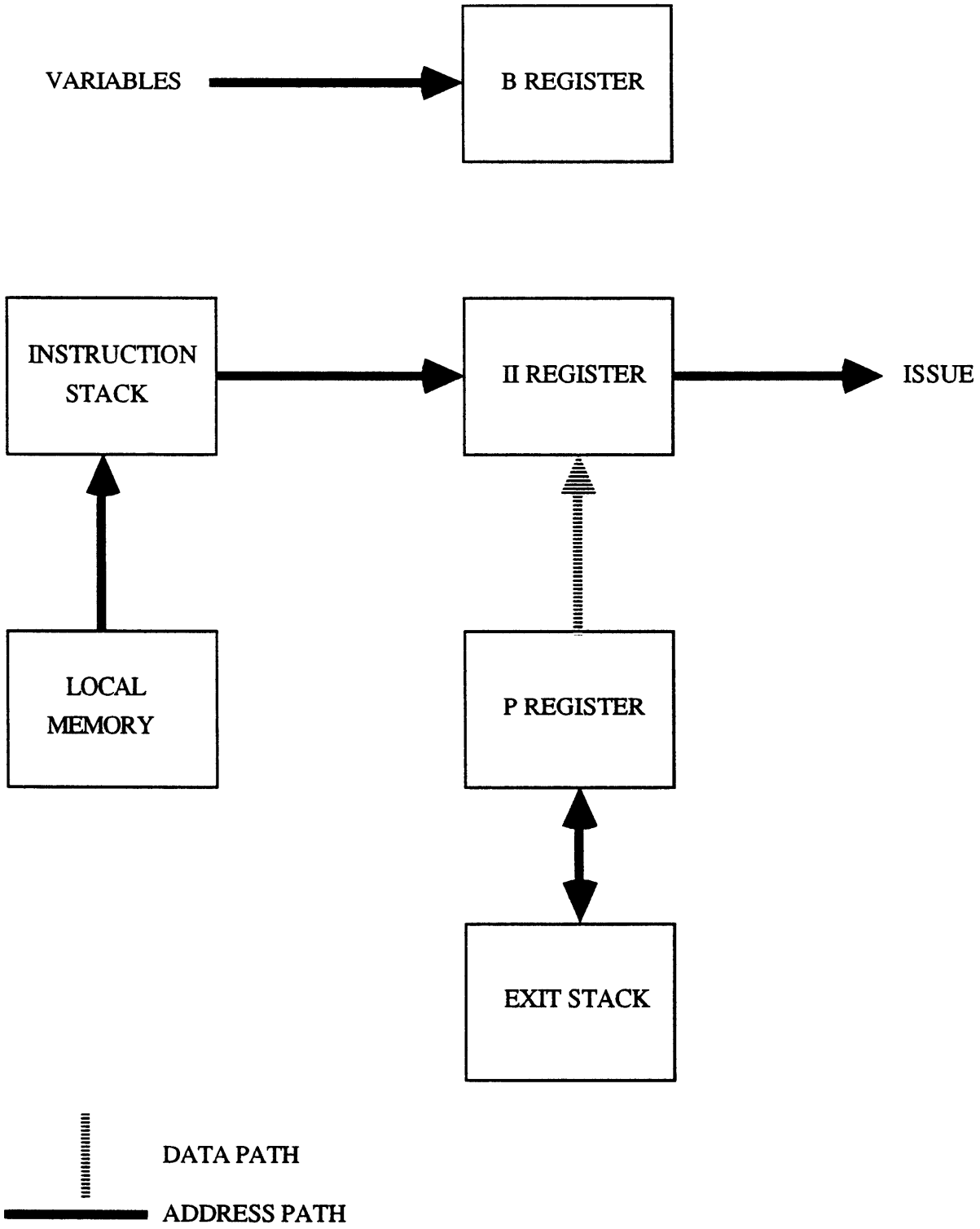


Figure I.2-2: Instruction Control Network Components

2.3 INSTRUCTION STACK

Instruction parcels are fetched from Local Memory and stored in the 32-parcel Instruction Stack prior to issue. This 32-parcel capacity allows for short program loops within the stack thereby reducing memory references. The Instruction Stack has 2 banks of registers, with 16 parcels in each bank. Addresses alternate between the banks, so loading of data interleaves with the readout of instructions for execution. As instructions are leaving the Instruction Stack, 4-parcel background fetches from Local Memory occur so the next instructions are available in the Instruction Stack for issue. The Instruction Stack is implemented as a circular buffer; when the stack is filled, loading continues at zero.

Branch Instructions

All absolute branch instructions are considered out of stack. An absolute branch instruction is one that loads an address into the P register without regard to the current instruction address, and execution continues at the branch address. See IOS Instruction Reference Card, instructions 074-077 and 120-137, for absolute branch instructions.

Relative branch instructions may be in or out of stack. A relative branch instruction is one that loads an address into the P register which is relative to the current program address. The branch address is calculated by adding an offset (+/-) to the current program address. See IOS Instruction Reference Card, instructions 070-073 and 100-177, for relative branch instructions. There are certain hardware restrictions for an in-stack condition:

a. Forward relative branches that are less than or equal to an offset of 9 do not generate an out-of-stack condition;

b. Backward relative branches that are less than or equal to an offset of 11 do not generate an out-of-stack condition;

c. The largest guaranteed loop is 12 parcels.

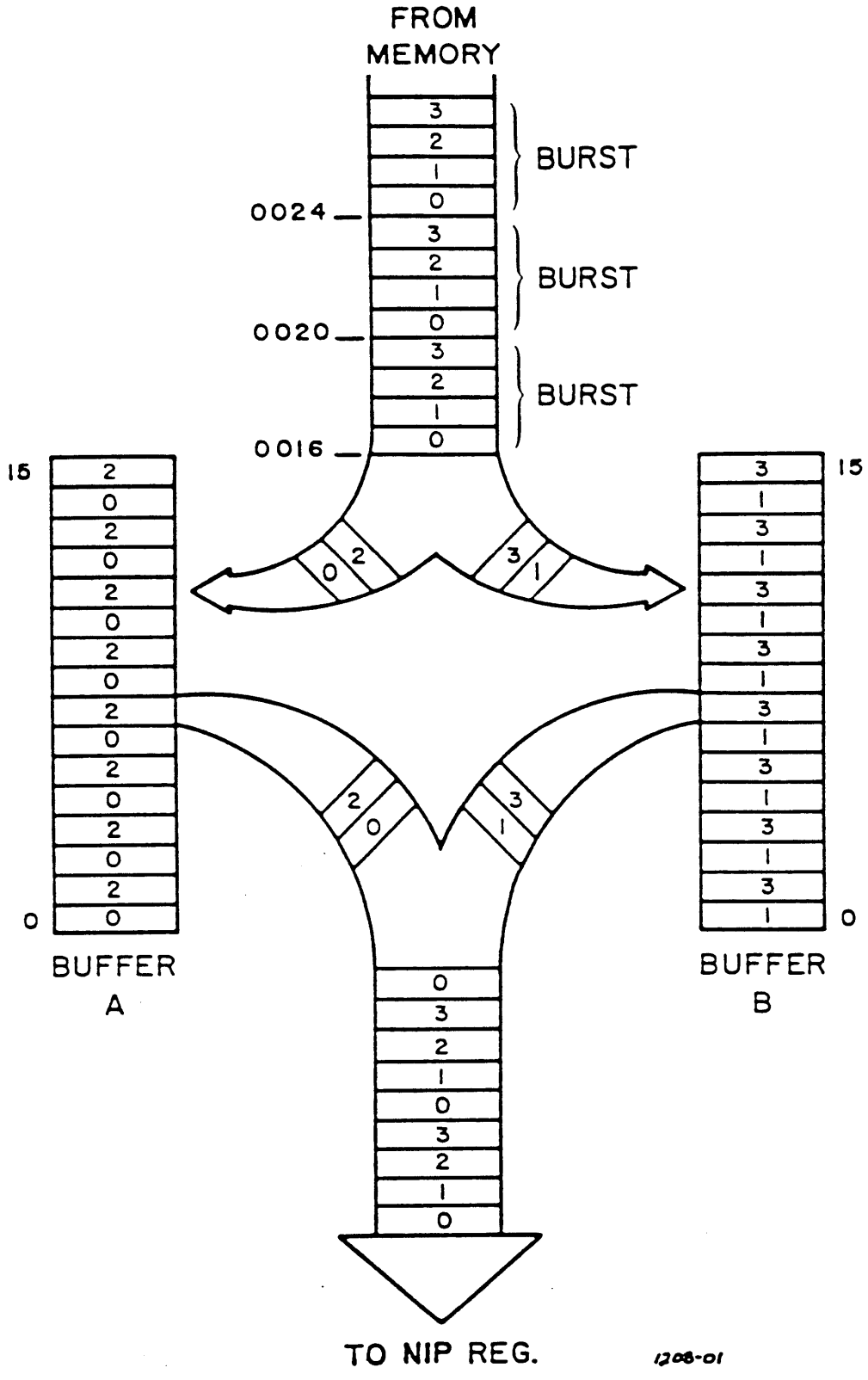


Figure I.2-3: Instruction Stack Operation

2.4 INSTRUCTION ISSUE REGISTER (II)

The II Register is a 16-bit register that receives instructions from the Instruction Stack, one parcel at a time. Instructions are one or two parcels in size.

A 1-parcel instruction consists of a 7-bit function code (f field), and a 9-bit constant (d field). The f field specifies the specific instruction to be executed. The d field contains data, address or shift displacement.

A 2-parcel instruction has a 16-bit constant (k field) as its second parcel.

When the first parcel arrives in the II Register, the f field is decoded to determine the sequence of operations required and whether the instruction is 1 or 2 parcels. In a 2-parcel instruction, the second parcel is issued from the II Register immediately after the f and d fields and is not interpreted as an instruction.

As shown on the next page, the contents of the d field may go to the Addend Register or to the Accumulator; or the d field may address an Operand Register or I/O Channel. Notice the contents of the d field cannot go directly to an Operand Register; it can only address an operand register. The k field contents may go to either the Addend Register or to the Accumulator.

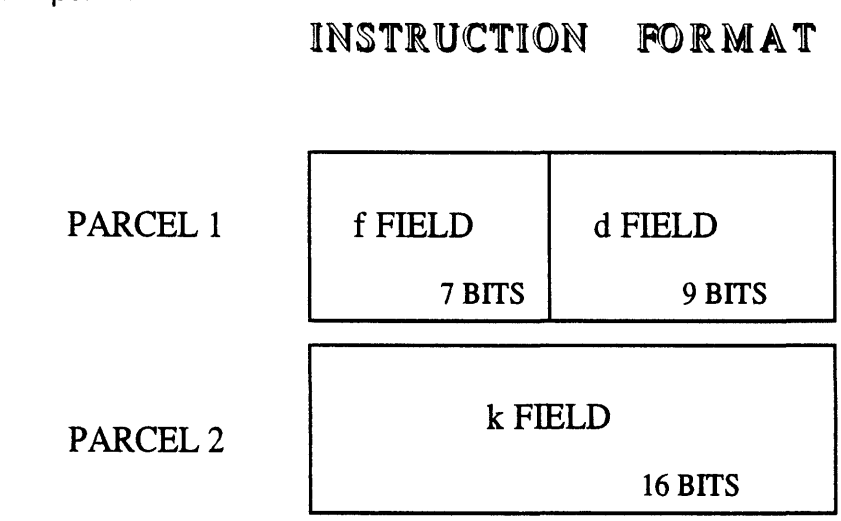


Figure I.2-4: Instruction Parcels

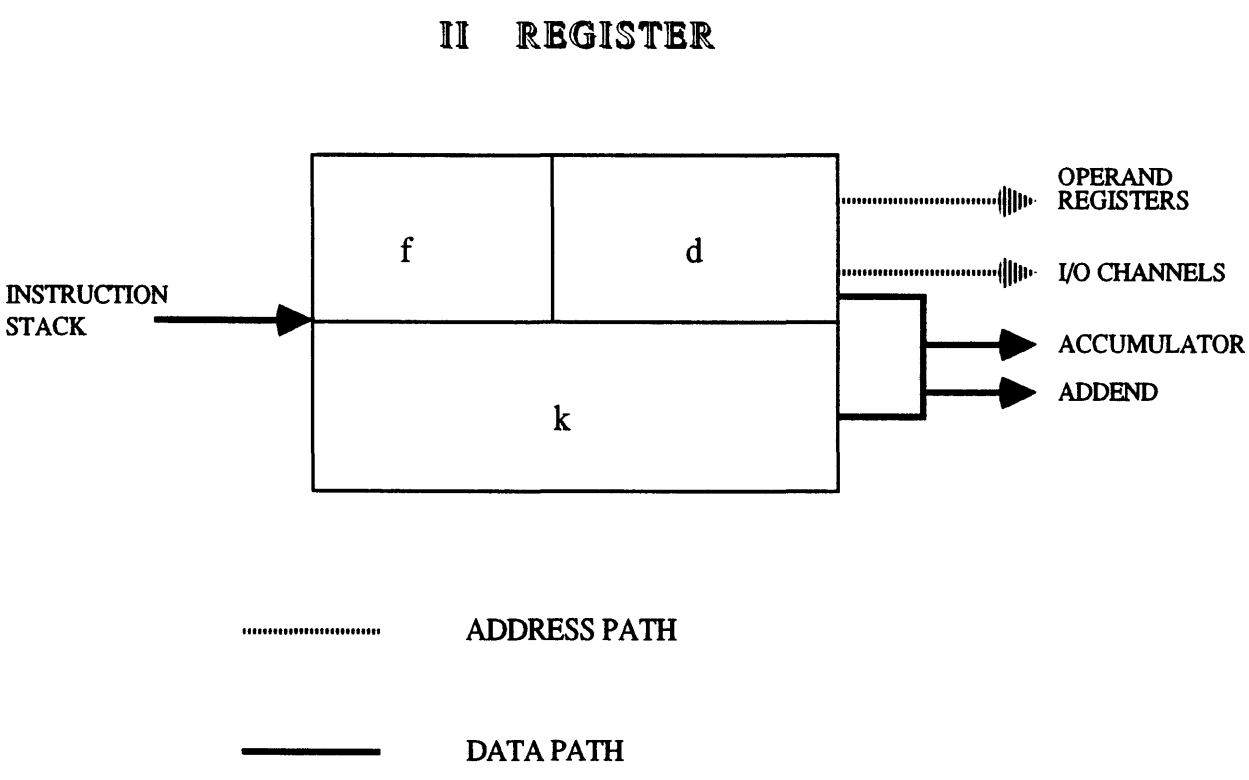


Figure I.2-5: Instruction Issue Register

2.5 B REGISTER

The B Register is 9 bits wide. The B Register contents are program-modifiable in contrast to the d field which is part of the program.

The B Register contents are used to address an I/O Channel or Operand Register, or as an operand value for the Accumulator or Addend Register.

The B Register is loaded from the lower 9 bits of the Accumulator. A value cannot go directly from the d field into the B Register.

An example of an I/O instruction using the B Register to perform a specific function on the channel addressed by the contents of the B Register would be written: IOB:(function code--0-17).

See IOS Reference Card, instructions 160-177, for specific I/O instructions using the B Register.

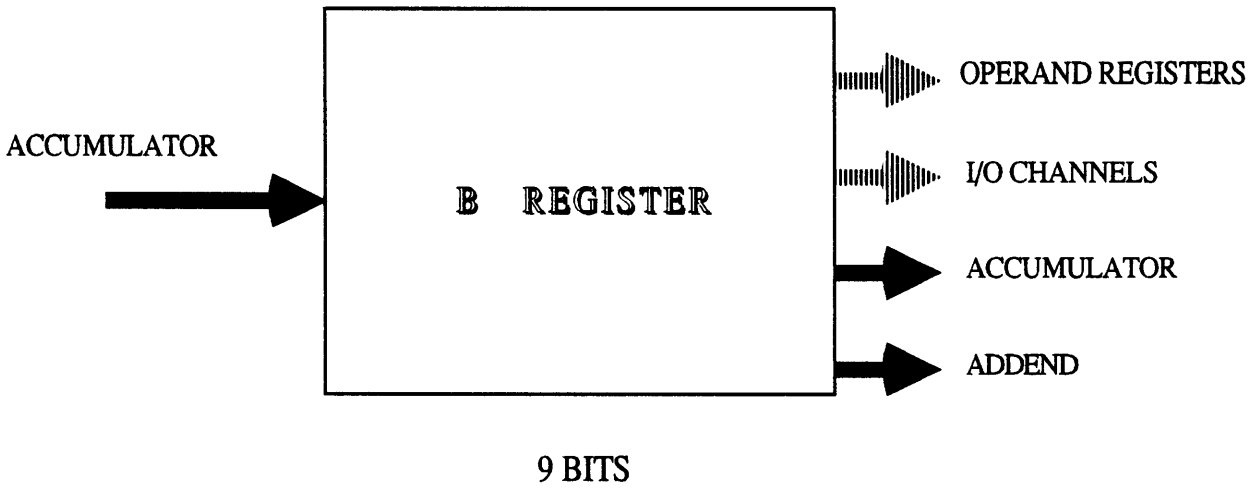


Figure I.2-6: B Register

2.6 P REGISTER

The Program Address (P) Register is 16 bits wide and holds the Local Memory address of the instruction in the II Register. The contents of the P Register automatically increment by 1 as instructions are sequentially issued.

Branch instructions utilize the P register contents by incrementing or decrementing the contents (relative branch), or by loading a new value (absolute branch).

For a relative branch address, the contents of the P register go to a background accumulator as an operand for the new address calculation. The new branch address is then returned to the P Register from the ADD functional unit.

The contents of the P Register may also go to the Exit Stack to store an interrupted program address or a subroutine return address. Conversely, the Exit Stack will load the P Register with the stored interrupted or return address to resume execution at that address.

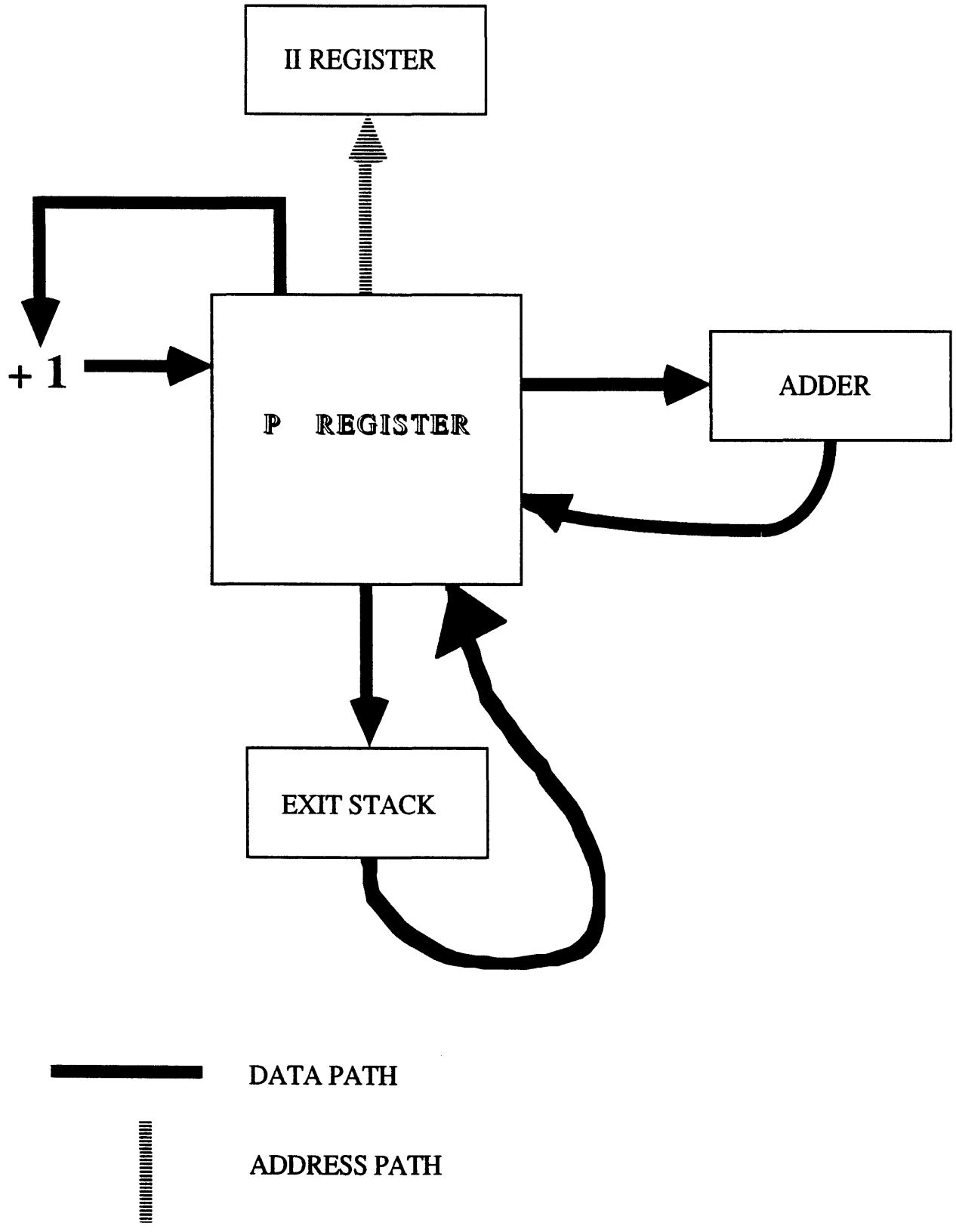


Figure I.2-7: Program Address Register

2.7 PROGRAM EXIT STACK

The Program Exit Stack is used to store:

1. Return Address on subroutine call
2. Interrupted program address
3. Interrupt Handler starting address

The Exit Stack consists of 16 16-bit registers. There is a 4-bit register, the E pointer, that addresses the register number of the most current entry on the stack.

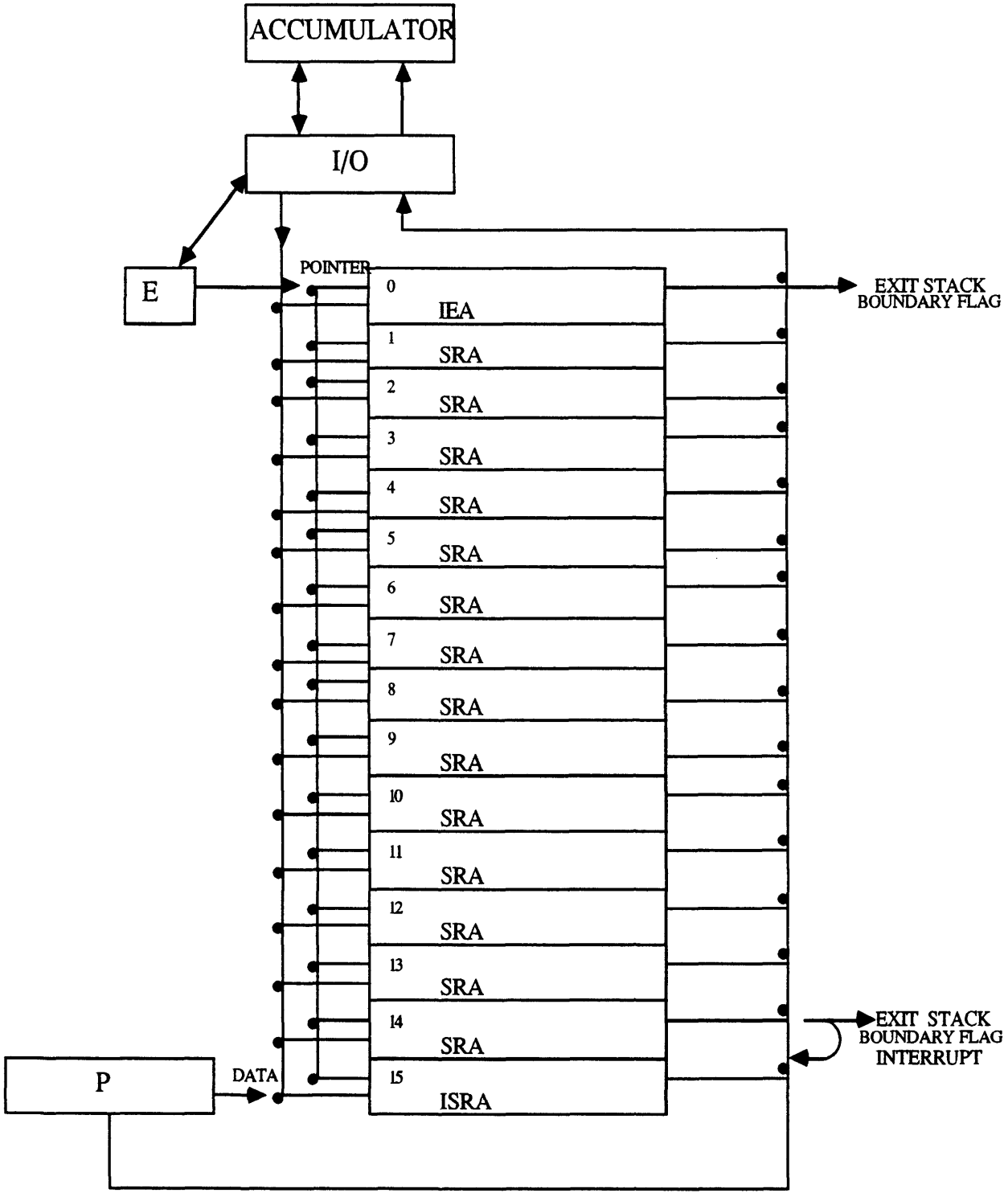
Exit Stack location 0 is reserved for the Software Interrupt Handler starting address. When an interrupt occurs, the E pointer is incremented by 1 and the next sequential program address is entered on the stack at the location addressed by the E pointer. The hardware then goes to location 0, without reference to the E pointer, and enters the Interrupt Handler address into the P register and begins execution at that address.

The Exit Stack locations 1-14 are used for subroutine return addresses and interrupted program addresses. A subroutine call from a program or an interrupt advances the E pointer by 1; stores the program's next sequential address on the stack; and loads the subroutine's address into the P Register and execution proceeds at the P Register address.

To return control to a calling routine, a subroutine issues an EXIT instruction as its last physical statement. An EXIT instruction references the E pointer for the Exit Stack location that holds the current calling routine's return address. The return address is entered into the P Register and E is decremented by 1. Execution continues at the address in the P Register.

The Exit Stack is considered both an I/O Channel and a part of the Instruction Control Network. The Exit Stack and the E Register contents can be modified by the Accumulator through I/O Channel 2. Since an I/O channel access takes several clock periods, and return jumps, EXIT and interrupts use the Exit Stack and E pointer contents immediately, a 5 clock period delay is necessary after any modification before using the stack for a RETURN, EXIT or interrupt.

There is a hardware mechanism which allows the Exit Stack to generate an interrupt so the software may reconfigure the stack when full. Boundary flags are set (interrupt condition) on either end of the stack (positions 0 and 14). If the E pointer reaches 14 due to a subroutine call, the return address is entered into location 14 causing the boundary flag to set. The calling routine's return address is entered into position 14 and the subroutine address is entered into the P Register. The P Register loads position 15 with the subroutine address (interrupted address). The Interrupted Handler would then be entered to store part or all of the stack into Local Memory which would allow for any number of nested subroutines. Conversely, if the E pointer is at 0 and an EXIT instruction is issued, an interrupt is generated and the Interrupt Handler entered to restore the stack from Local Memory. ***Note: The operating system software does not utilize this feature.***



IEA = Interrupt Entrance Address
SRA = Subroutine Return Address
ISRA = Interrupted Subroutine Return Address

Figure I.2-8: Program Exit Stack

2.8 OPERAND REGISTERS

There are 512 Operand Registers, numbered 0-777 (octal), which are used for temporary data storage, as index registers and for indirect memory addressing.

The Operand Registers are addressed by the d field or the B Register, and are loaded from the Accumulator.

The contents of the Operand Registers may go to the Accumulator or to the Addend Register only. For example, data cannot go directly from one Operand Register to another; data must go through the Accumulator.

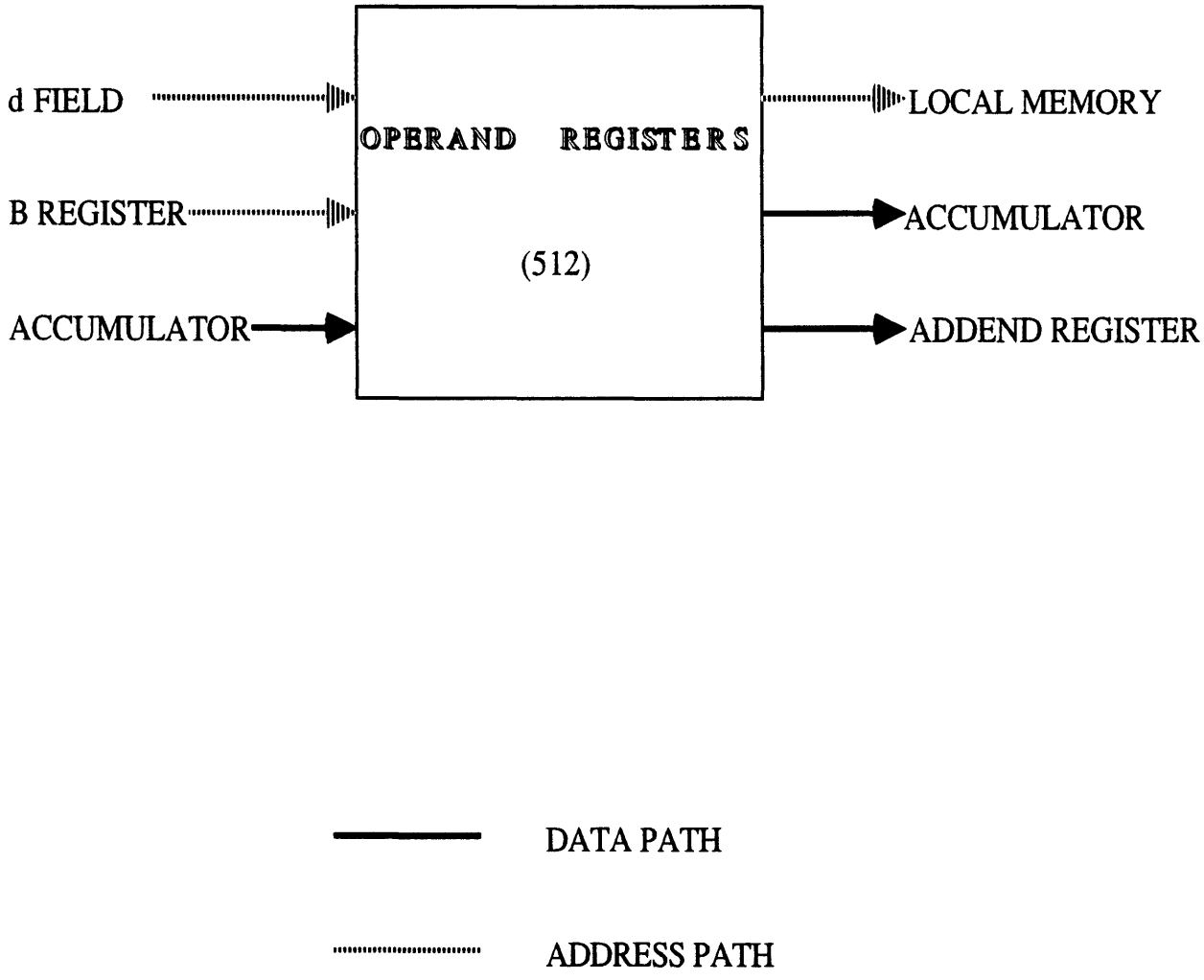


Figure I.2-9: Operand Registers

2.9 ACCUMULATOR

The 16-bit Accumulator is the main register of an I/O Processor. Most instructions involve the Accumulator for temporary storage of operands and results.

The logical product function is at input to the Accumulator; it does not require its own functional unit. Operands are "anded" with the contents of the Accumulator (no extra time is taken for this function). The result is then contained in the Accumulator.

The Carry Bit is a 1-bit register which can be considered the 17th bit position of the Accumulator; it is included in all adds, subtracts and shifts. It holds a carry generated from the adder or shifter. The Carry Bit is cleared when an operand is loaded into the Accumulator.

Branch instructions do not alter the contents of the Accumulator. There is a separate background Accumulator used for branch address calculation.

Data from several sources can be routed to the Accumulator; many destinations for Accumulator contents are available. See the next page for Accumulator data sources and destinations.

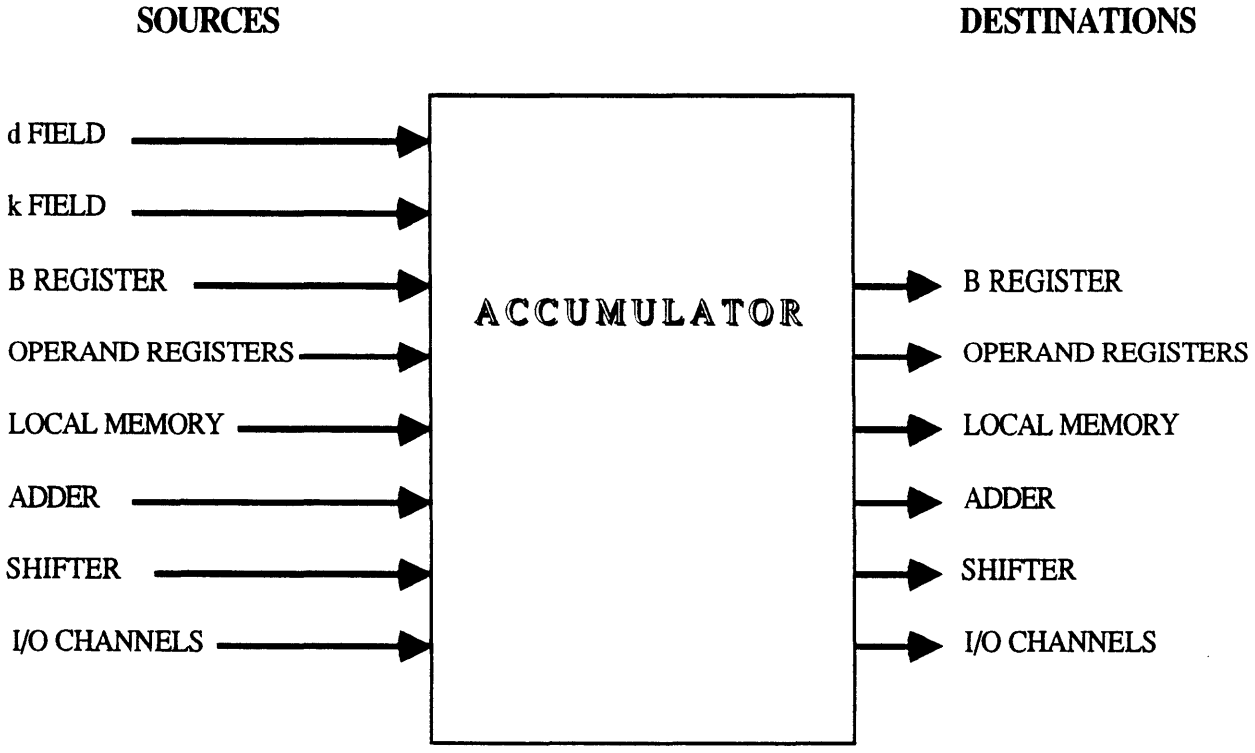


Figure I.2-10: Accumulator

2.10 ADDEND REGISTER

The 16-bit Addend Register supplies operands to the Adder and shift count to the Shifter functional units. One operand is always supplied by the Accumulator; the second operand by the Addend Register.

The Addend Register receives data from the B Register, the instruction d and k fields, the operand registers and Local Memory.

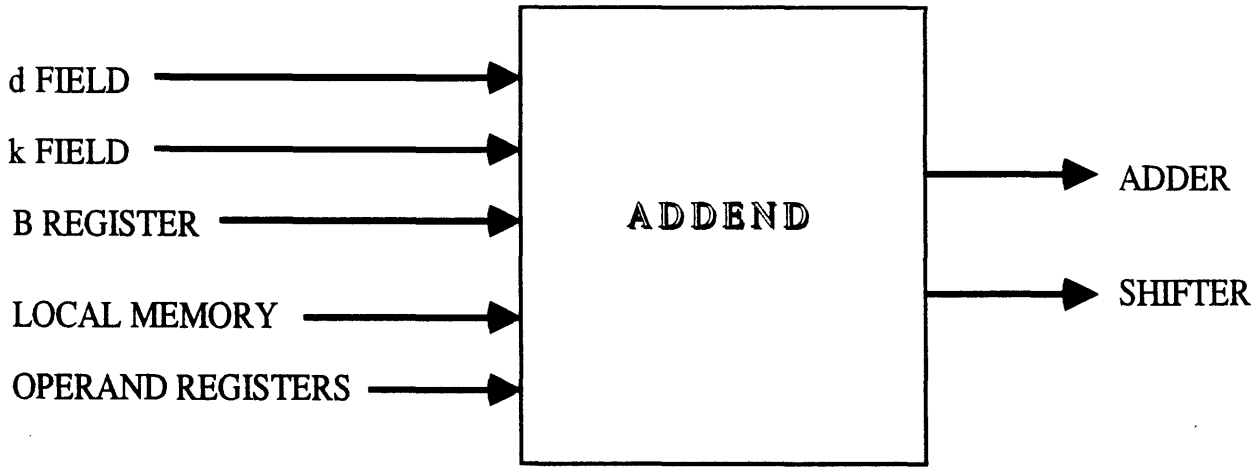


Figure I.2-11 Addend Register

2.11 FUNCTIONAL UNITS

Adder

The Adder functional unit provides for integer addition and subtraction in two's complement. The operands are supplied by the Addend Register and the Accumulator. The result of an Adder function goes to the Accumulator except for a branch address calculation where the final result goes to the P Register. In 16-bit subtraction, the carry bit is toggled if the result is positive or zero. Refer to the example on the opposite page for the Carry Bit value in a series of subtractions. Note in the subtraction process the value to be subtracted, (in the Addend Register), is complemented; added to the Accumulator operand; and a 1 is added to the intermediate result to produce the final result. The Carry Bit is toggled if either addition process generates a carry. A positive answer toggles the Carry Bit, while the Carry Bit is not affected with a negative result.

Shifter

The Shifter functional unit shifts the content of the Accumulator and the Carry Bit. The 5-bit shift count is supplied by the Addend Register. The Shifter provides for left, right circular or end-off zero fill shift operations.

"AND"

The "AND" operation forms the logical product of the Accumulator and an operand at input to the Accumulator. Operands are supplied from the d and k fields, operand registers, or a Local Memory location. No additional time is taken for this function.

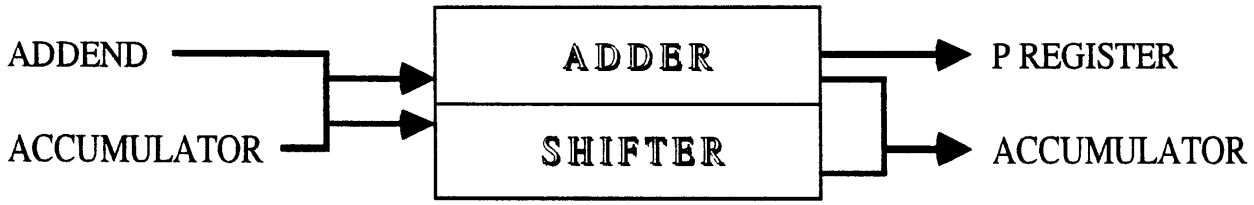


Figure I.2-12: Functional Units

1) 4-2:

```
                0000000000000100
              + 1111111111111101
C=1           ------------------
                0000000000000001
                               +1
              ------------------
                0000000000000010   = 2
```

2) 2-2:

```
                0000000000000010
              + 1111111111111101
C=0           ------------------
                1111111111111111
                               +1
              ------------------
                0000000000000000   = 0
```

3) 2-4:

```
                0000000000000010
              + 1111111111111011
C=0           ------------------
                1111111111111101
                               +1
              ------------------
                1111111111111110   = -2
```

FIGURE I.2-13: Carry Bit Change in X-Y Operations

2.12 INSTRUCTIONS

An instruction is 1 or 2 parcels long. The upper 7 bits of the first parcel is the function code (f field). The lower 9 bits is a positive designator (d field). The second parcel of an instruction is a 16-bit positive constant (k field).

The d field is used as:

- an operand register or I/O channel designator
- shift count (lower 5 bits)
- displacement for a branch instruction
- an operand value

The k field is used as:

- displacement for a branch instruction
- an operand for the Adder and logical "and" operation

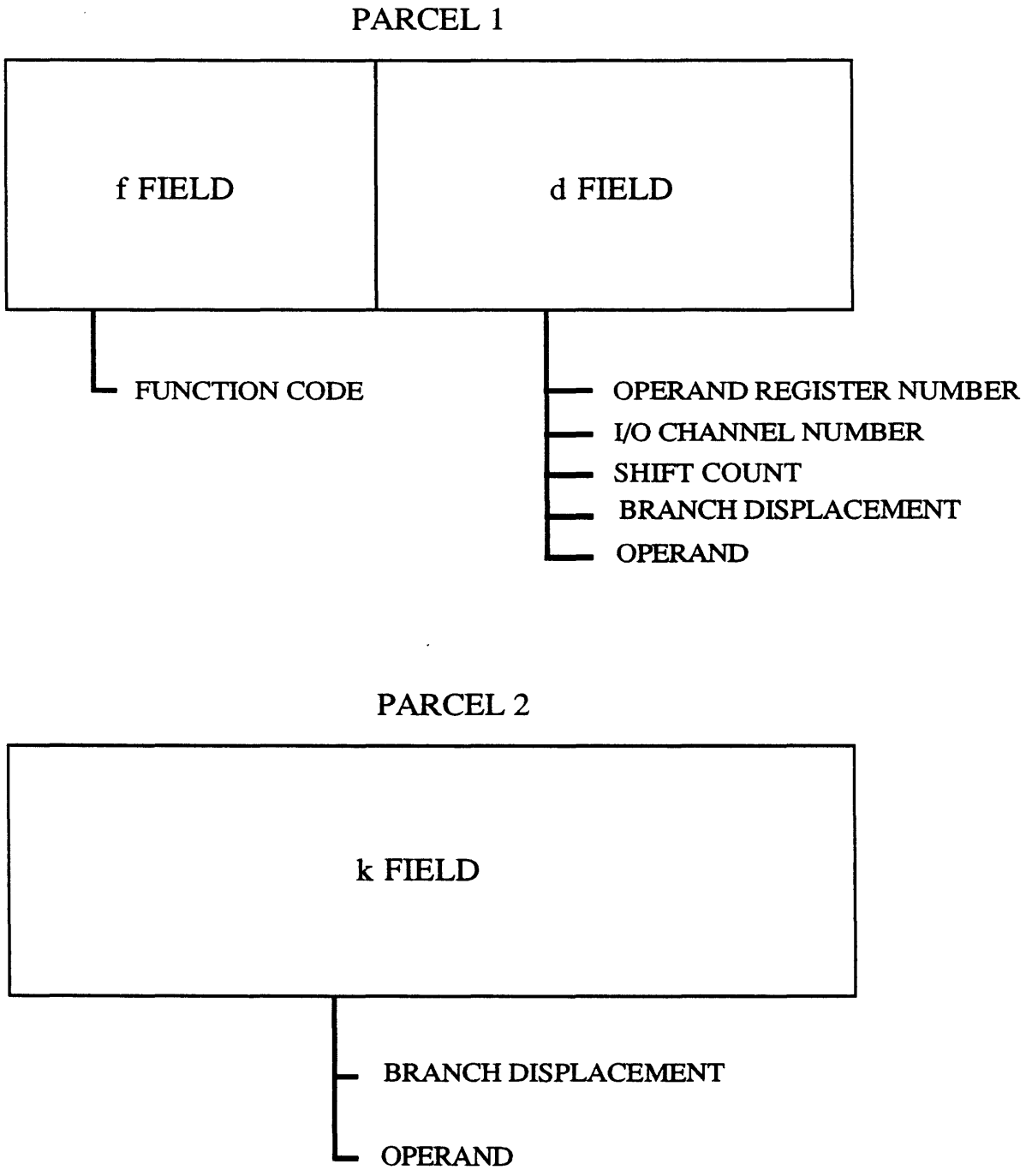


Figure I.2-14: Instructions

2.13 NOTATION

In a symbolic notation for I/O Processor instructions, certain symbols are reserved to represent IOP registers and memory.

A	This symbol indicates the contents of the Accumulator. When used in a shift operation, includes the Carry Bit
B	Indicates the contents of Register B
(B)	Parentheses in this notation indicate the contents of the operand register addressed by contents of the B Register
C	Indicates the value of the Carry Bit (1 or 0). Can be used as criterion for conditional jump instructions. Can be set to test channel states
d	Indicates the contents of the d field. A value is substituted for this notation
k	Indicates the contents of the k field. A value is substituted for this notation
dd	Indicates the contents of an operand register which is addressed by the d field. A symbol with value is substituted for this notation
(dd)	Indicates the contents of the memory location whose address is contained in the operand register addressed by the d field. A symbol with value is substituted for this notation
,	A Comma literally means "if". Makes an assignment statement conditional

The following reserved names are the notation for I/O channel commands:

IOB	I/O Channel referenced by B; the channel number is contained in the B Register
iod	Conventionally, a 3-character mnemonic to address an I/O channel. The value of the 3-character symbol (iod) is the channel designator, which is contained in the d field.

An instruction can use either notation. For example, to perform a function 0 on channel 5:

```
Using the B Register      - IOB:0
Using the iod mnemonic    - MOS:0
```

In this example, the assumption is made that MOS has been previously equated to channel 5, and the value 5 has previously been loaded into the B Register.

A	Accumulator
B	B Register
(B)	Content of Operand Register addressed by B
C	Carry Bit
d	Instruction Issue, d Field
k	Instruction Issue, k Field
dd	Content of Operand Register addressed by ***d***
(dd)	Content of Memory addressed by Operand Register ***dd***
,	If
iod	3-Character Mnemonic for Channel addressed by **B**
IOB	Channel addressed by **B**

Figure I.2-15: Instruction Notation

2.14 SYSTEM CONTROL INSTRUCTIONS

System control instructions are used to manage interrupts and monitor channel states.

The following instructions are for program and interrupt control.

000-PASS - This instruction acts as a null operation for 1 clock period.

001-EXIT -The EXIT instruction terminates the program sequence to return to the sequence that was suspended in calling a subroutine. It is the last statement in a called subroutine which pops the caller's return address off the Exit Stack and loads the address into the P Register.

002-I=0 -This instruction disables the system interrupt mechanism. It must be followed with a PASS instruction which allows the hardware time to disable interrupts before proceeding to the next instruction. NOTE: At the APML level, the assembler automatically issues a PASS instruction making this instruction a 2-parcel instruction.

003-I=1 -This instruction reenables the system interrupt mechanism. It is the last instruction in the Interrupt Handler before the EXIT which returns control to the interrupted program. Therefore, interrupts are not reenabled with this instruction until the completion of the next non-branch or non-I/O instruction which allows time to return to the interrupted program without an interrupt.

The following instructions set the Carry Bit depending upon the specified channel's state. Channel states are monitored by busy (BZ) and done (DN) flags. At the completion of I/O, the DN flag is set which creates an interrupt condition.

040-C=1,iod=DN - This instruction sets the Carry Bit if the channel designated by the value of the mnemonic in the d field has its done flag set.

041-C=1,iod=BZ - This instruction sets the Carry Bit if the channel designated by the value of the mnemonic in the d field has its busy flag set.

042-C=1,IOB=DN - This instruction sets the Carry Bit if the channel designed by the B Register has its done flag set.

043-C=1,IOB=BZ - This instruction sets the Carry Bit if the channel designed by the B Register has its busy flag set.

INSTRUCTION	MNEMONIC	DESCRIPTION
000	PASS	Null Operation
001	EXIT	Exit From Subroutine
002	I=0	Disable System Interrupts
003	I=1	Enable System Interrupts

the following instructions force carry bit to same state as specified channel's Done (DN) or Busy (BZ) flag

040	C=1, iod=DN	Set Carry Bit According to State of Flag on Channel Specified in d Field
041	C=1, iod=BZ	
042	C=1, IOB=DN	Set Carry Bit According to State of Flag on Channel Specified in B Register
043	C=1, IOB=BZ	

Figure I.2-16: System Control Instructions

2.15 JUMP INSTRUCTIONS

There are 40 unique instruction codes for jump instructions. There are 8 unconditional jump instructions that can be made conditional by appending the following conditions.

- If the value of the Carry Bit is zero or non-zero
  - ,C=0
  - ,C#0
- If the value of the Accumulator is zero or non-zero
  - ,A=0
  - ,A#0

The 8 basic jump instructions fall into the following categories:

1. Relative Jumps

   a. Relative jumps using the d field value as an offset to the current program address.

   b. Relative return jumps using the d field value as an offset to the current program address. The next sequential address following the return jump instruction is pushed onto the Exit Stack.

2. Absolute Jumps

   a. Absolute jump to an address contained in the operand register designated by the d field.

   b. Absolute jump to an address which is the sum of the address contained in the operand register designed by the d field and the value in the k field.

   c. Absolute return jump to the address contained in the operand register designated by the d field. The next sequential program address following this instruction is pushed onto the Exit Stack.

   d. Absolute return jump to the address which is the sum of the address contained in the operand register designated by the d field and the constant k. The next sequential program address following this instruction is pushed onto the Exit Stack.

1) RELATIVE JUMP INSTRUCTIONS

070	P=P+d
071	P=P-d
072	R=P+d
073	R=P-d

2) ABSOLUTE JUMP INSTRUCTIONS

074	P=dd
075	P=dd+k
076	R=dd
077	R=dd+k

Figure I.2-17: Basic Jump Instructions

2.16 I/O PROCESSOR LOGICAL LAYOUT

The block diagram on the next page shows the interrelationship of the programmer-visible components of an I/O Processor; i.e., the logical layout of an I/O Processor. As in previous diagrams, solid lines indicate data paths, and broken lines are address paths.

See Appendix A of this text for a more detailed IOP diagram which includes various hardware registers not visible to the programmer.

I/O PROCESSOR BLOCK DIAGRAM

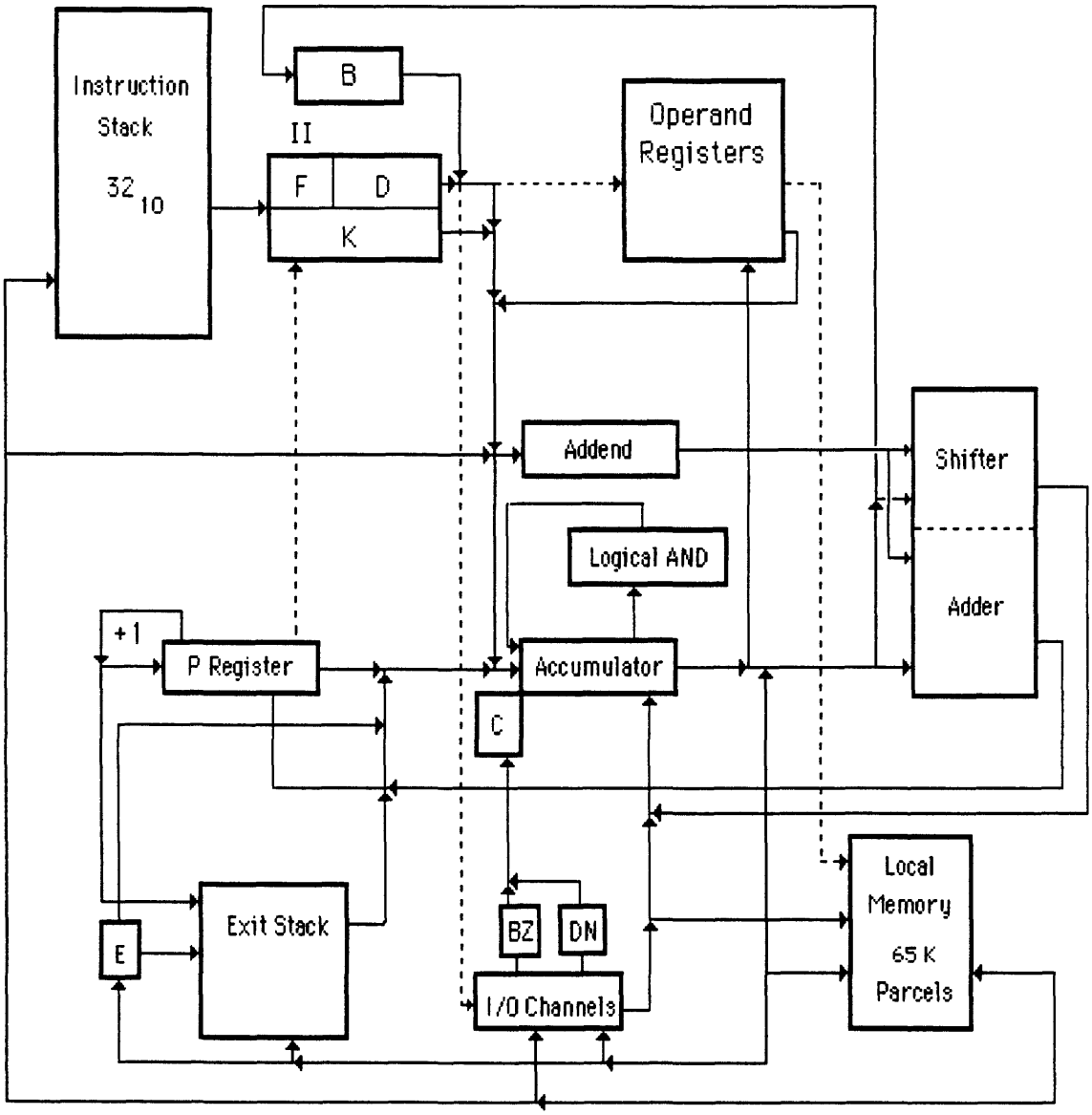


Figure I.2-18: I/O Processor Logical Layout

CHAPTER 2
REVIEW QUESTIONS

1. The B Register is 9 bits wide and may be used to:
   a. Index a OPERAND REGISTER
   b. INDICATE AN I/O CHANNEL
   c. LOAD Accumulator or Address Register

2. What is the Exit Stack used for?
   a. Subroutine Return Addresses
   b. Store Interrupted Program Addresses
   c. Interrupt HANDLER entrance Address

3. Suppose the Exit Stack pointer is 11 and R1 represents an operand register. What is the value of the Exit Stack pointer after the following sequence of instructions executes? What value is stored at this location in the Exit Stack? 1001
   12

LOCATION	*INSTRUCTION*	*MNEMONIC*
776	010300	A=300
777	024ddd	R1=A
1000	076ddd	R=R1

4. There are 512 Operand Registers, each loaded only from the Accumulator.

5. Suppose C=1 and the Accumulator is loaded with 17777 (octal). What is the new value of C? C=0

6. Consider the following sequence of instructions:

0	002000	I=0
1	000000	PASS
2	003000	I=1
3	070001	P=P+1
4	040005	C=1,MOS=DN
5	104001	P=P-1,C=0
6	050000	A=B
7	071007	P=P-7

Is this loop interruptible? If so, where?

Yes, the loop is interruptible after the A=B statement, as this is the first Nav branch, non I/O statement after I=1

CHAPTER 2
ANSWERS TO REVIEW QUESTIONS

1. The B Register is 9 bits wide and may be used to:

   a. Index an Operand Register
   b. Address an I/O Channel
   c. Load Accumulator or Addend Register

2. The Exit Stack is used to hold:

   a. Interrupt Handler entrance address
   b. Return addresses for subroutine calls
   c. Interrupted program addresses

3. After the instruction sequence executes the value of the Exit Stack pointer is incremented by 1 to hold the return address for the subroutine call (R=R1); therefore the Exit Stack pointer is now equal to 12. The value stored in the Exit Stack at this location is the next sequential instruction to execute; therefore, the value stored in the Exit Stack is 1001.

4. There are 512 Operand Registers, each loaded only from the Accumulator.

5. The new value of the Carry Bit is zero. Loading of the Accumulator always clears the Carry Bit (C).

6. This loop is interruptible, but not until completion of the next non-branch or non-I/O instruction after I-1. The first instruction that fits these requirements is A=B; therefore, this loop is interruptible after execution of instruction 6.

RELATED READING

The following reading from HR-0081, IOS Model C Hardware Reference Manual, is highly recommended before proceeding with Chapter 3.

1. Chapter 1; "The I/O Subsystem" and Chapter 2, pp. 2-1 to 2-17; "The I/O Processor" -- Introduction, The IOP Control Section, The IOP Computation Section.

2. Chapter 5; examine the functions and timing of instructions 000 through 137.

CHAPTER 3
I/O SECTION

3.1 CONFIGURATION

There are up to 42 I/O channels on each I/O Processor, numbered 0-51 (octal). Of these 42 channels, 12 (0-13) are standard for each IOP, and up to 30 (14-51) channels may be implemented differently by each IOP. (Refer to IOS Model C Hardware Reference Manual, HR-0081, Appendix C, for typical channel assignments.)

Input channels are even numbered; output channels are odd numbered. Exceptions to this are the channels that are both input and output; i.e., Exit Stack and Buffer Memory (MOS).

Faster devices connect to an I/O Processor via DMA ports. A DMA port may attach one device or several devices. One DMA port is reserved on each IOP for connection to Buffer Memory. Illustrated on the next page are the uses of two DMA ports on the BIOP. One port has Buffer Memory connected as its sole device; the other port multiplexes four disk drives.

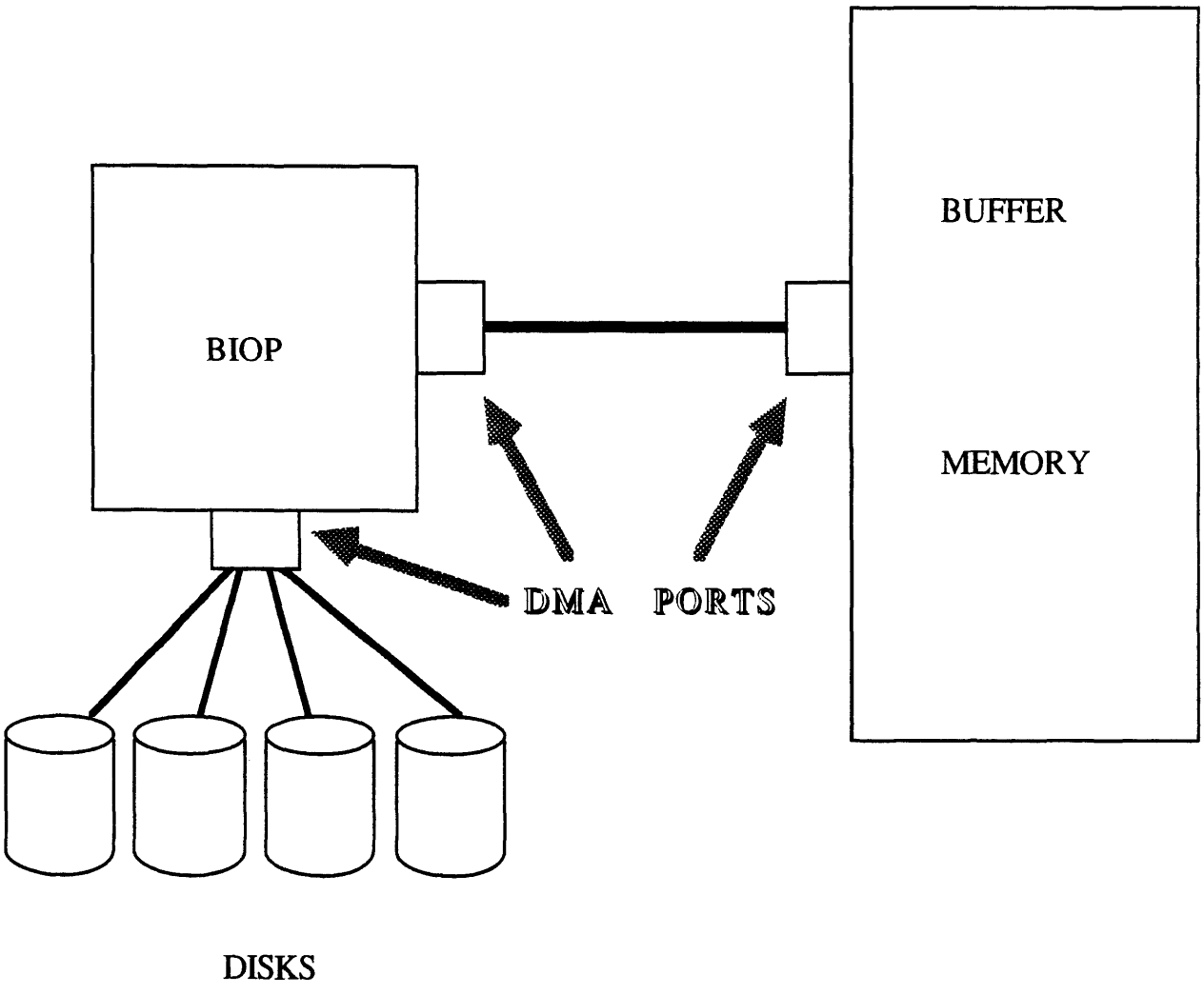


Figure I.3-1: Sample DMA Port Usage

3.2 CHANNEL TYPES

There are two types of channels for an I/O Subsystem: Accumulator and Direct Memory Access (DMA).

ACCUMULATOR

An accumulator channel is a 16-bit path between an interface and the IOP's accumulator used to transfer one parcel of data at a time.

Accumulator channels are utilized for small volume I/O and low speed devices. All control information in an I/O instruction must go through the accumulator. Some examples of accumulator channels are:

- Consoles (display and keyboard)
- Error logging
- Inter-IOP communication

Standard IOP channels 0-4 and 6-13 are all accumulator channels.

DIRECT MEMORY ACCESS (DMA)

A DMA channel consists of an accumulator channel with a data path into Local Memory through a DMA port.

DMA channels are used for high volume I/O; transfers are 4 parcels per read or write request. In a data transfer, the accumulator issues the request, and data is transferred directly to/from Local Memory. Examples of DMA channels are:

- Buffer Memory (standard IOP channel 5)
- CPU high and low speed
- Disk
- Block Multiplexer
- Peripheral Expander

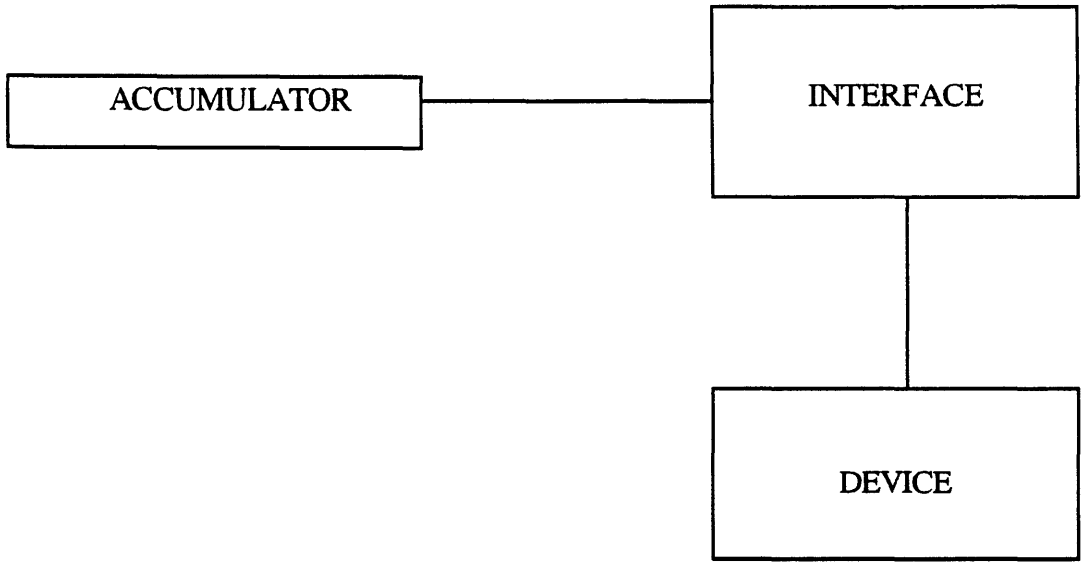


Figure I.3-2: An Accumulator Channel

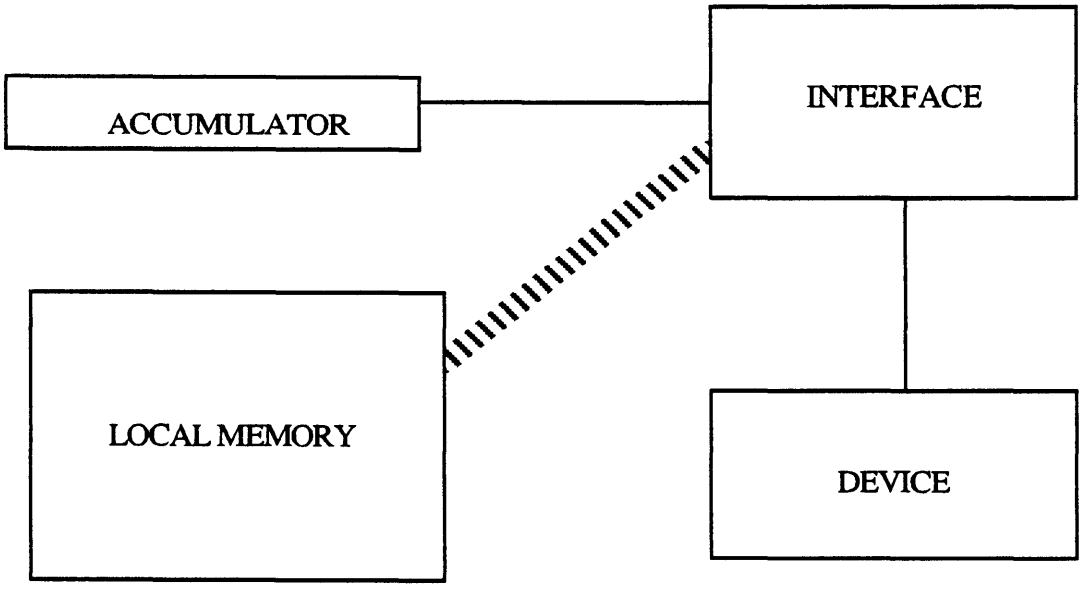


Figure I.3-3: A DMA Channel

3.3 OVERVIEW OF I/O

An I/O channel instruction is sent to the interface which is specified in the d field or the B Register (iod or IOB). The function code is then interpreted by the interface; the operation requested varies with the specific interface. However, three function codes are common to most interfaces; they are functions 0, 6 and 7, as shown on the next page. Note that the interrupt mechanism on a specific channel can be set or cleared, the same as with the system interrupts discussed in Chapter 2.

I/O channel states are monitored through interface Busy (BZ) and Done (DN) flags. When a DN flag sets, an interrupt condition is created. Channel states are also shown on the next page.

iod:0 IOB:0	CLEAR DN and BZ FLAGS
iod:6 IOB:6	DISABLE CHANNEL INTERRUPTS
iod:7 IOB:7	ENABLE CHANNEL INTERRUPTS

Figure I.3-4: Common Channel Instructions

CHANNEL IDLE - BZ=0, DN=0

START I/O - BZ=1, DN=0

FINISH I/O - BZ=0, DN=1

I/O ERROR - BZ=1, DN=1

Figure I.3-5: Channel States

3.4 DEDICATED CHANNELS (0-13)

The dedicated, or standard, channels are found on all I/O Processors.

I/O REQUEST CHANNEL 0 (accumulator channel)

This channel has only one function (10). Instruction IOR:10 reads the highest priority (lowest number) interrupting channel number into the accumulator. The accumulator value is then used to handle the specific interrupt. All interrupts have been handled when a zero value is returned to the accumulator.

In the example on the next page, DN flags are set on channels 3, 12 and 7. The first IOR:10 reads the highest priority channel number (3) into the accumulator. That number is loaded into the B register where the next instruction clears the flags for that channel. The sequence repeats for the next two channel numbers. Finally all interrupts have been handled indicated by the zero value in the accumulator.

INSTRUCTION	ACCUMULATOR VALUE
IOR:10 B=A IOB:0	A=3
IOR:10 B=A IOB:0	A=7
IOR:10 B=A IOB:0	A=12
IOR:10	A=0

Figure I.3-6: I/O Request Channel 0

PROGRAM FETCH REQUEST CHANNEL 1 (accumulator channel)

This channel provides a hardware mechanism to alert the monitor (Kernel) to load in code that is not in Local Memory. ***This feature is not utilized by our Operating System software.***

The Program Fetch Request flag is set (causing an interrupt condition) during execution of an absolute jump instruction when the contents of the operand register, dd, is zero. (See IOS reference card, instructions 074-077 and 120-137). The interrupt would cause the operand register number d, to be loaded into an interface register for access by the Kernel. Each operand register number would correspond to a specific overlay. The overlay indicated would then be loaded into Local Memory by the Kernel.

The hardware instructions for this channel are:

PFR:0 Clear PFR flag. There is no busy flag.

PFR:6 Clear channel interrupt enable flag (IEF)

PFR:7 Set IEF

PFR:10 Load Accumulator with operand register number and clear PFR flag

dd=0

REGISTER NUMBER	OVERLAY
1	OVERLAY 1
2	OVERLAY 2
3	OVERLAY 3
•	•
•	•
•	•

Figure I.3-7: Program Fetch Request Channel 1

PROGRAM EXIT STACK CHANNEL 2 (accumulator channel)

Hardware instructions for Channel 2 provide information necessary to restructure the Exit Stack when the Exit Stack boundary flag is set. The boundary flag is set in two situations:

1. A subroutine return address is loaded into stack location 14. Addresses in part or all of the stack would be stored in Local Memory for later retrieval.

2. An EXIT instruction issued with the E pointer at location 0 of the stack. Addresses which have been stored previously would be restored to the stack from Local Memory.

Note: Our Operating System software does not restructure the Exit Stack - Refer to Chapter 2 of this text.

Instructions for Channel 2 are as follows:

PXS:0 Clear Exit Stack Boundary Flag. No busy flag.

PXS:6 Clear IEF (interrupt enable flag)

PXS:7 Set IEF

PXS:10 Transmit E pointer value to accumulator; clear C

PXS:11 Transmit Exit Stack entry (E) to accumulator; clear C

PXS:13 Read contents of jump history log

PXS:14 Transmit accumulator value to E pointer

PXS:15 Transmit accumulator value to Exit Stack (E)

PXS:16 Enter diagnostic mode

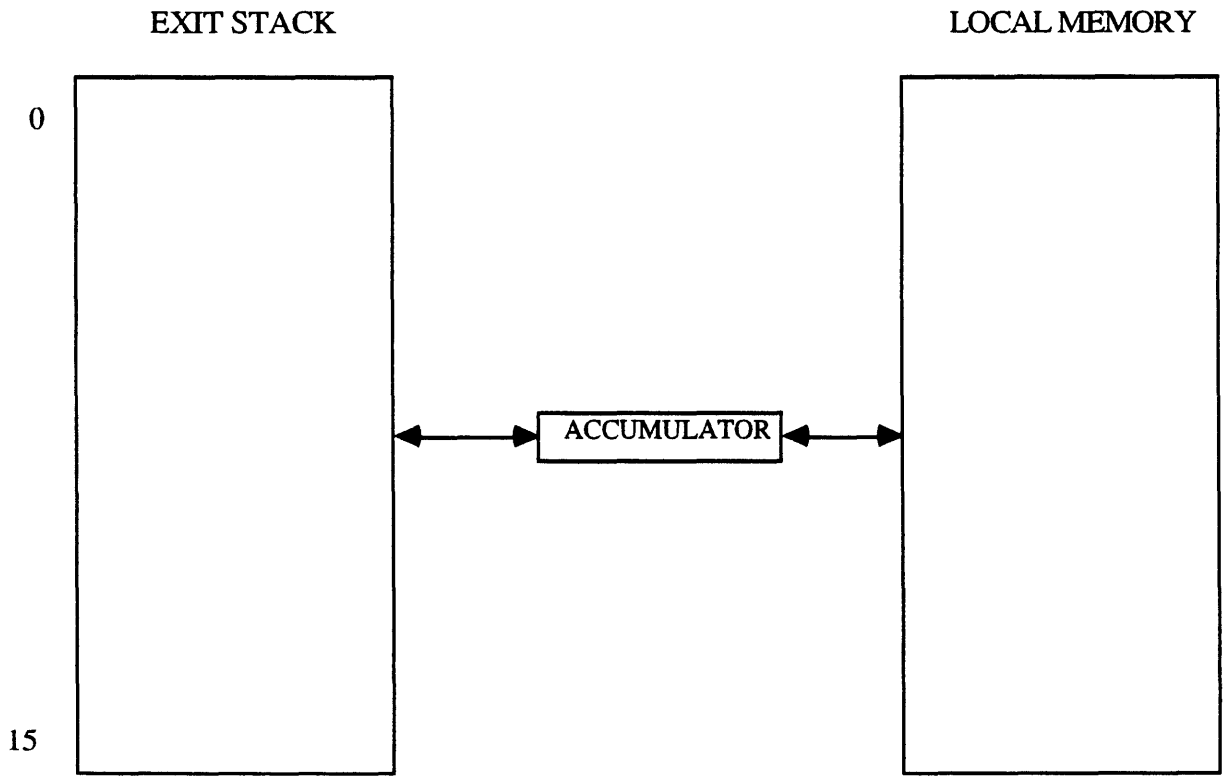


Figure I.3-8: Program Exit Stack Channel 2

LOCAL MEMORY ERROR CHANNEL 3 (accumulator channel)

Channel 3 provides maintenance information on Local Memory errors. The MIOP logs these errors through its Error Logging channel.

Channel 3 instructions are as follows:

LME:0 Clear parity error flag. No busy flag.

LME:6 Clear IEF

LME:7 Set IEF

PROVIDES MAINTENANCE
INFORMATION ON
LOCAL MEMORY ERRORS:

- BANK
- SECTION
- BYTE

Figure I.3-9: Local Memory Error Channel 3

REAL TIME CLOCK CHANNEL 4 (accumulator channel)

There is a 12.5 ns clock on an IOP. The IOP Real Time Clock is a 17-bit counter/timer which interrupts the IOP at 1 millisecond intervals. The Real Time Clock adds 1 each clock period up to an octal count of 234177, which is 1 ms, interrupts and resets to 0. This fixed time interrupt allows the operating system to time out events (such as pending interrupts) as well as to maintain the time of day.

The Real Time Clock is read by the accumulator. Since the accumulator only holds 16 bits, reading of the clock is accurate to 2 cps. See HR-0081, IOS Model C Hardware Reference Manual, for the algorithm for timing tests.

The instructions for Channel 4 are:

RTC:0 Clear DN flag

RTC:6 Clear IEF

RTC:7 Set IEF

RTC:10 Read high order bits of RTC into accumulator

17 BITS

* ADDS 1 EACH CLOCK PERIOD (12.5 ns)

* INTERRUPTS EVERY ms

Figure I.3-10: Real Time Clock Channel 4

BUFFER MEMORY CHANNEL 5 (DMA channel)

On Channel 5, blocks of data are transferred in either direction, but not in both directions simultaneously.

Three interface registers must be loaded from the accumulator before a block transfer can take place. The Buffer Memory address is held in a 24-bit register. The high order 15 bits are entered with a function 2 request; the low order 9 bits with a function 3 request. The Local Memory address is held in a 14-bit register which is entered with a function 1 request. The block length in 64-bit words is held in a 14-bit register which is entered with a function 4 (read) or a function 5 (write) request. Loading of the block length register starts the data transfer.

The following are the Channel 5 instructions:

MOS:0 Clear DN and BZ flags (both set on error). Must be done after every double bit error before next transfer.

MOS:1 Load Local Memory Address Register with upper 14 bits of accumulator.

MOS:2 Load upper 15 bits of Buffer Memory Address Register with lower 15 bits of accumulator.

MOS:3 Load lower 9 bits of Buffer Memory Address Register with lower 9 bits of accumulator.

MOS:4 Load Buffer Memory Block Length Register with lower 14 bits of accumulator. Start Buffer Memory to Local Memory block transfer.

MOS:5 Load Buffer Memory Block Length Register with lower 14 bits of accumulator. Start Local Memory to Buffer Memory block transfer.

MOS:6 Clear IEF

MOS:7 Set IEF

MOS:10 Read errors

MOS:14 Load diagnostic control register with bits 2^1 and 2^2 of accumulator. Used only while in diagnostic mode.

MOS:15 Load a second diagnostic control register with bits 2^1 , 2^2 and 2^3 of the accumulator. Used only while in diagnostic mode.

MOS:16 Enter bypass mode. This allows data to transfer directly between Cray central memory and Buffer Memory, bypassing the local memory of an IOP supporting a 100 MBYTE/S data channel.

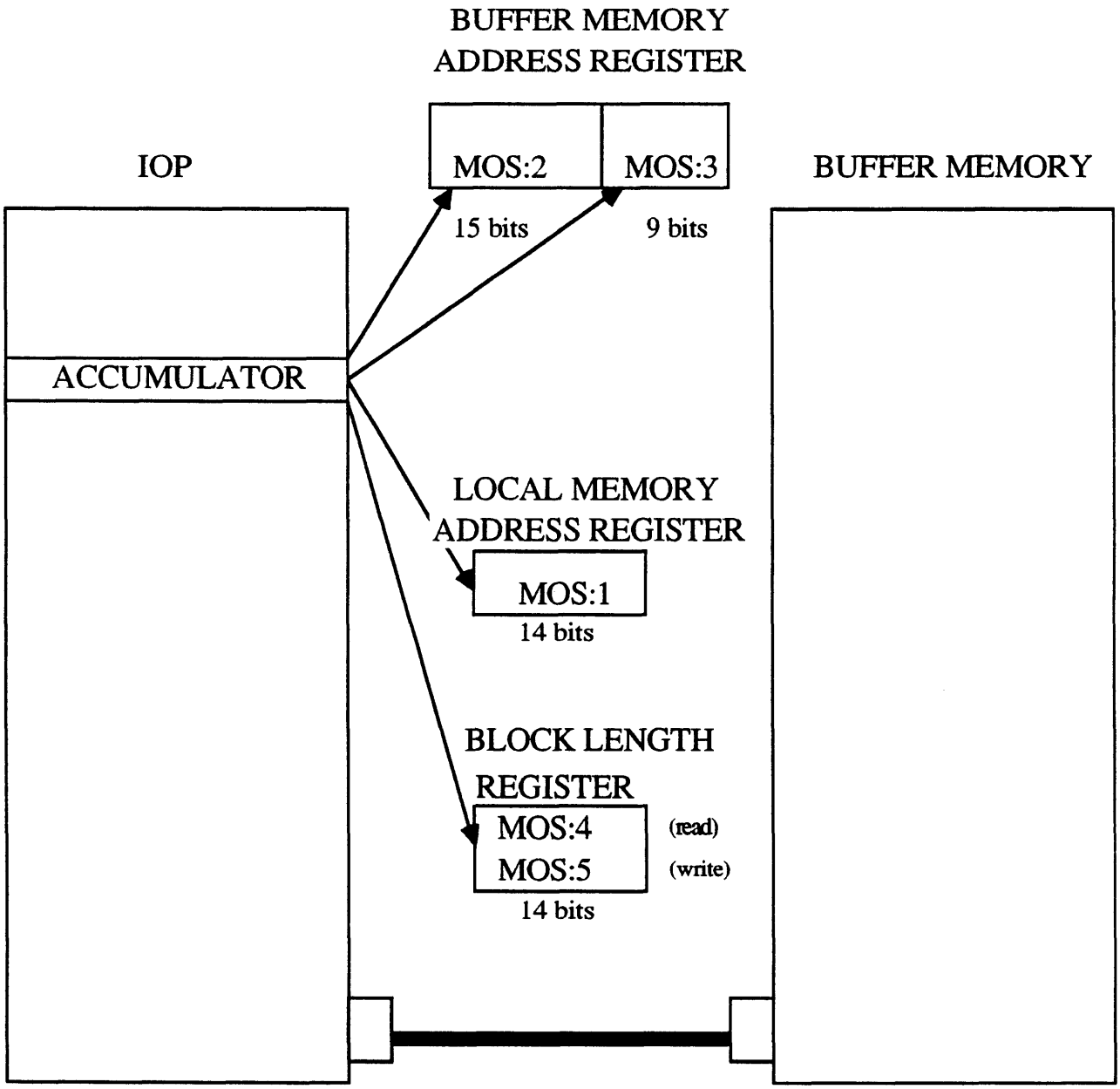


Figure I.3-11: Buffer Memory Channel 5

INTERPROCESSOR COMMUNICATION CHANNELS (accumulator channels)

Every IOP has channels to communicate with every other IOP. Each channel uses a 16-bit interface register to hold the data parcel being transferred.

The interface causes an interrupt in the opposite IOP. When a message is input to the interface register, an interrupt is generated in the receiving IOP; conversely, when the message is taken out by the receiving IOP, an interrupt is generated in the sending IOP.

The interprocessor communication channel numbers correspond to which IOP is controlling; the lower the IOP number, the higher priority channel number. The channel numbers are indicated by the last letter of the 3-character mnemonic as shown on the opposite page.

I/O PROCESSOR INPUT CHANNEL 6, 10, 12

The following are the instructions for an IOP input channel (the '*' corresponds to letters A=6, B=10, C=12):

AI*:0 Clear DN Flag. No busy flag.

AI*:6 Clear IEF

AI*:7 Set IEF

AI*:10 Read interface to accumulator. This clears interface.

I/O PROCESSOR OUTPUT CHANNEL 7, 11, 13

The following are the instructions for an IOP output channel. Notice instruction A0*:1 allows an IOP to Master Clear, Deadstart and Dead Dump another IOP through a 3-bit control register.

AO*:0 Clear BZ and DN flags

AO*:1 Load control register with lower 3 bits of accumulator; bit 0=master clear; bit 1=deadstart, bit 2=dead dump

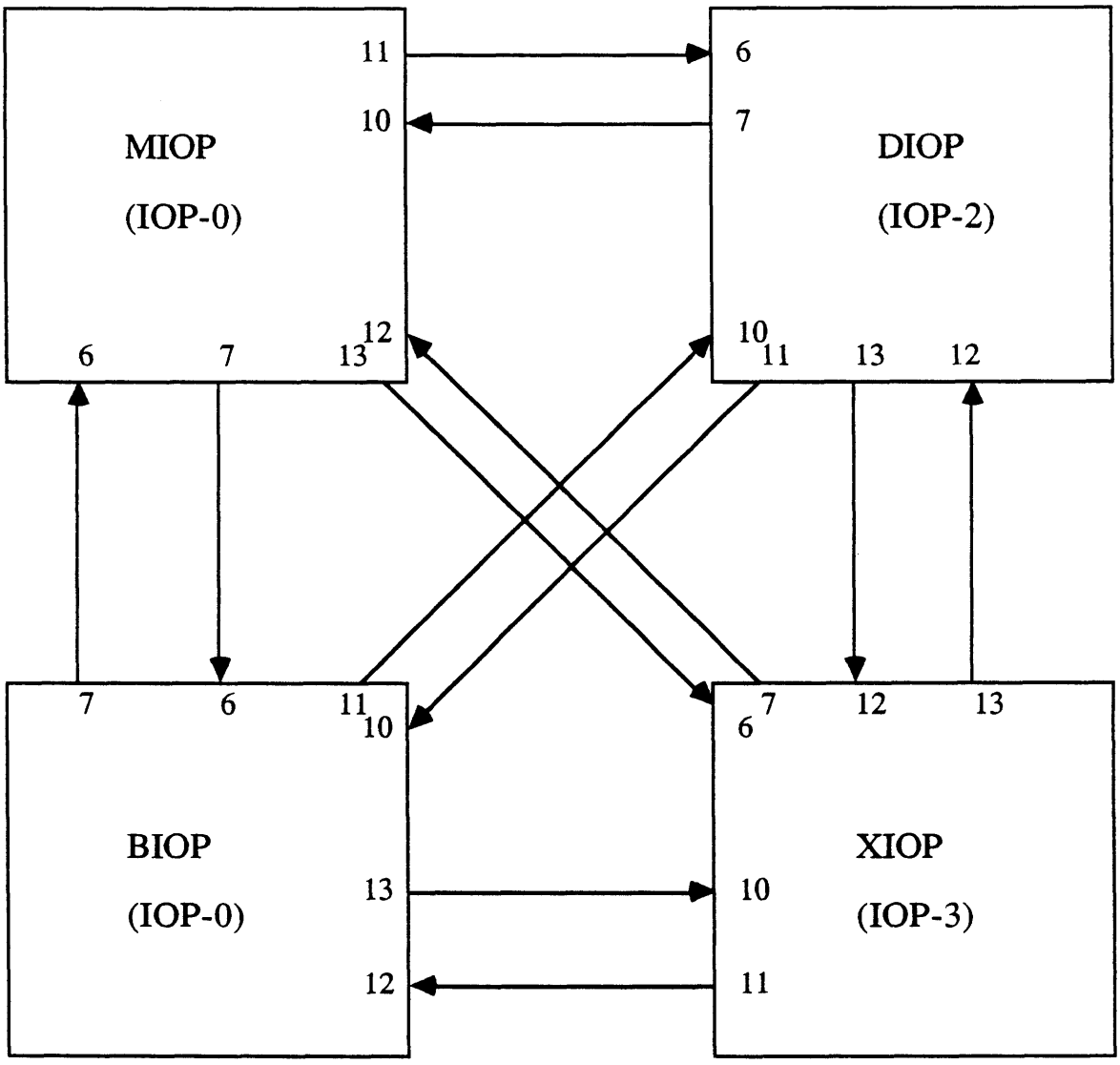
AO*:6 Clear IEF

AO*:7 Set IEF

AO*:14 Load interface register with accumulator. DN flag sets when target IOP performs an AI*:10.

SO in the code of the IOS operating system you may see instructions such as
AIC:7 AIB:10 AOA:14 AOC:1
where A, B, C are used to determine which IOP we're talking to; this distinction is necessary because the channel functions are identical except for which channel they're issued to

NOTE: IOB instructions may also be used for all channels.



AIA = 6

AIB = 10_8

AIC = 12_8

AOA = 7

AOB = 11_8

AOC = 13_8

Figure I.3-12: Interprocessor Communication Channels

3.5 INTERFACE CHANNELS

The rest of this section on I/O Channels will deal with the specialized channels on difference IOPs. The discussion will be limited to channel names and characteristics. Refer to HR-0081 Model C IOS Hardware Reference Manual, for specific channel functions and typical channel assignments. Appendix B of this text shows all the channels on each individual I/O Processor.

CPU I/O CHANNELS (DMA channels)

All control information between the I/O Subsystem and the CPU is communicated over these channel pairs; i.e., an interrupt is generated on these channels when a data transfer is complete.

INPUT FROM CPU I/O CHANNEL CIA->CID

These channels are for input from the CPU (communication channel) and from front end computers. There are up to eight channels on the MIOP.

OUTPUT FROM CPU I/O CHANNEL COA->COD

These channels accept output from Local Memory to the CPU and front end computers. There are up to eight channels on the MIOP. The channel to the CPU provides the I/O Subsystem with CPU deadstart capability.

Each channel pair (CI*-CO*) transfers directly to/from Local Memory via a DMA port.

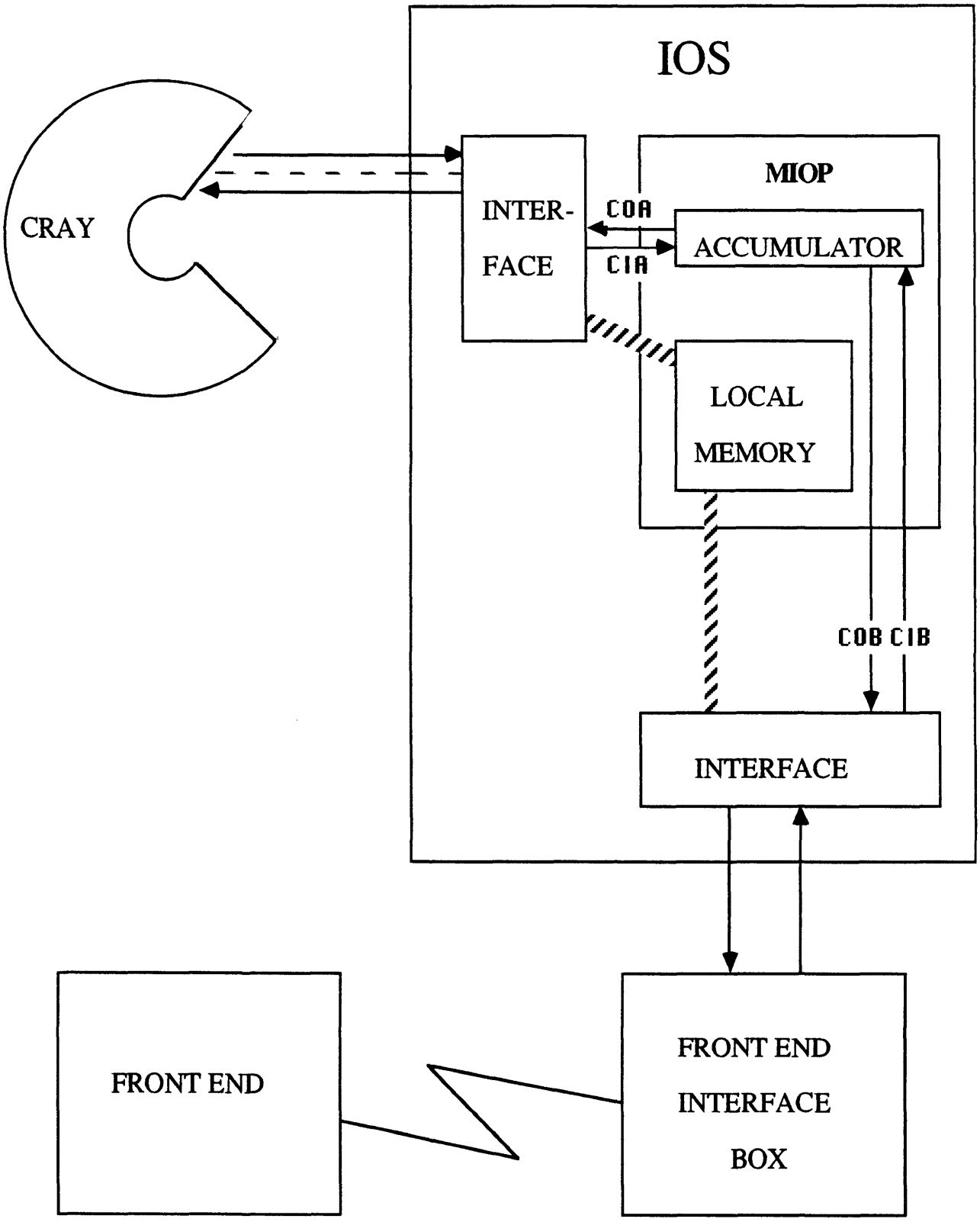


Figure I.3-13: CPU-Type 6 MBYTE Channels

MEMORY (100 MBYTE) CHANNEL (DMA channel)

A memory channel consists of an input channel that carries data from Central Memory to Local Memory, and an output channel that passes data from Local Memory to Central Memory. A transfer of data can be only one way at a time.

As shown on the next page, the CPU double buffers the data; as one buffer is filling, the other is transferring. The interfaces break 64-bit words into 16-bit parcels on input, and group parcels into words on output.

The IOP has total control over this channel. When a block transfer is complete, an interrupt occurs in the IOP. This channel does not generate an interrupt in the CPU.

INPUT FROM CPU MEMORY CHANNEL-HIA

This channel accepts data from Central Memory into IOP Local Memory. The maximum data transfer rate is in excess of 800 MBITS/S.

OUTPUT TO CPU MEMORY CHANNEL-HOA

This channel passes data directly to Central Memory from IOP Local Memory. The maximum data transfer rate is in excess of 800 MBITS/S.

Both HIA and HOA channels may currently be used in bypass mode, such that data moves directly between Central Memory and Buffer Memory. HIA and HOA channels may be connected to an SSD, but in this case it is not possible to execute in bypass mode. The IOS software released with COS 1.16 is the first version of IOS that supports this "backdoor" to SSD, allowing data to move from disk to SSD without going through Cray Central Memory.

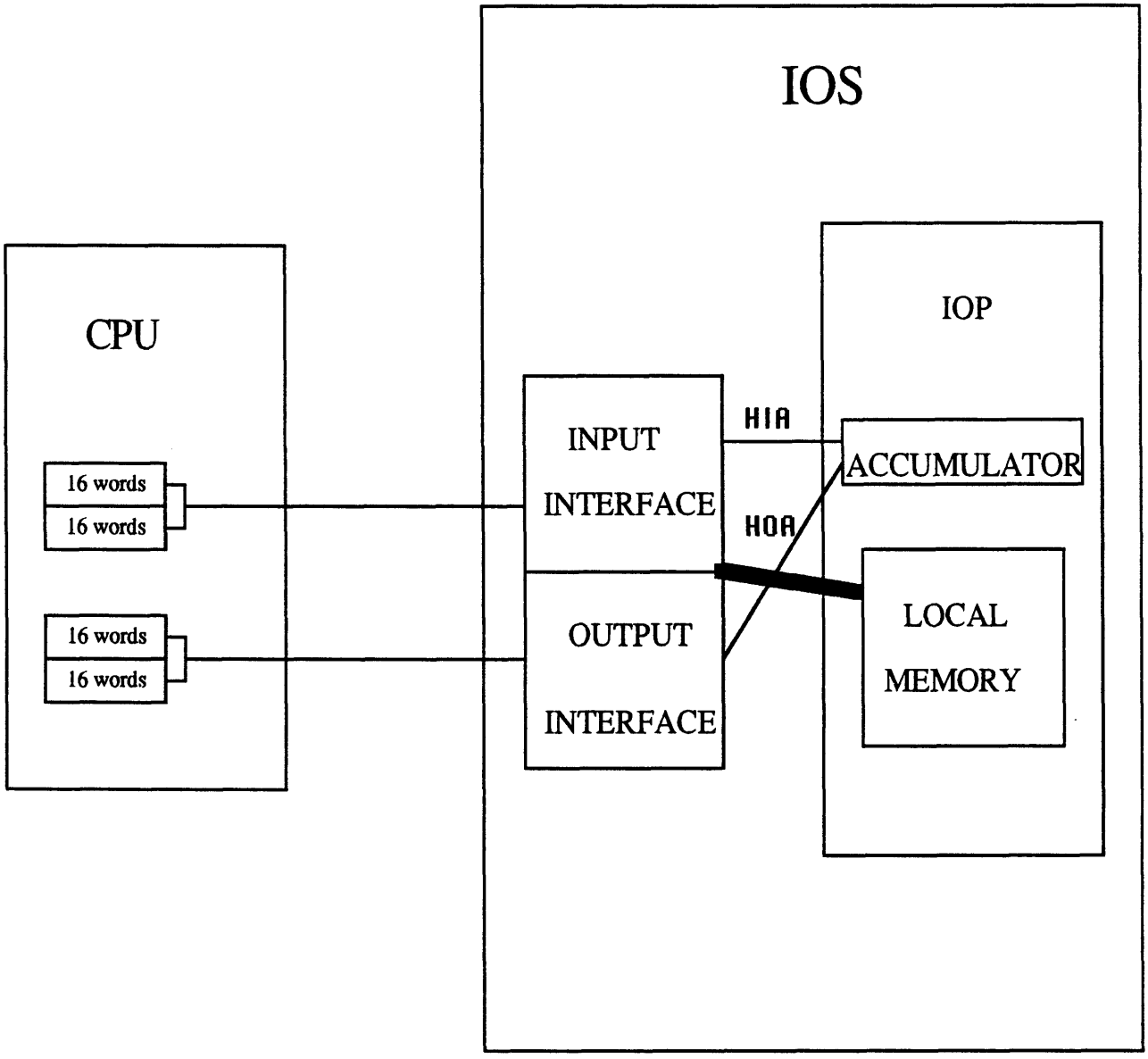


Figure I.3-14: Memory (100 MBYTE) Channel

EXPANDER CHASSIS CHANNEL EXB (DMA channel)

The MIOP has one channel connected to a Peripheral Expander. The peripheral expander can contain up to 16 controllers for peripheral devices. Only one controller can be active at one time, but that controller can be servicing more than one peripheral device. The current function of the EXB channel is to transfer data to/from the tri-density tape drive, 80 MBYTE disk and the printer/plotter. Data transfer speed to the peripheral expander is 16 MBITS/S, while the transfer rate to the MIOP is 14.5 MBITS/S.

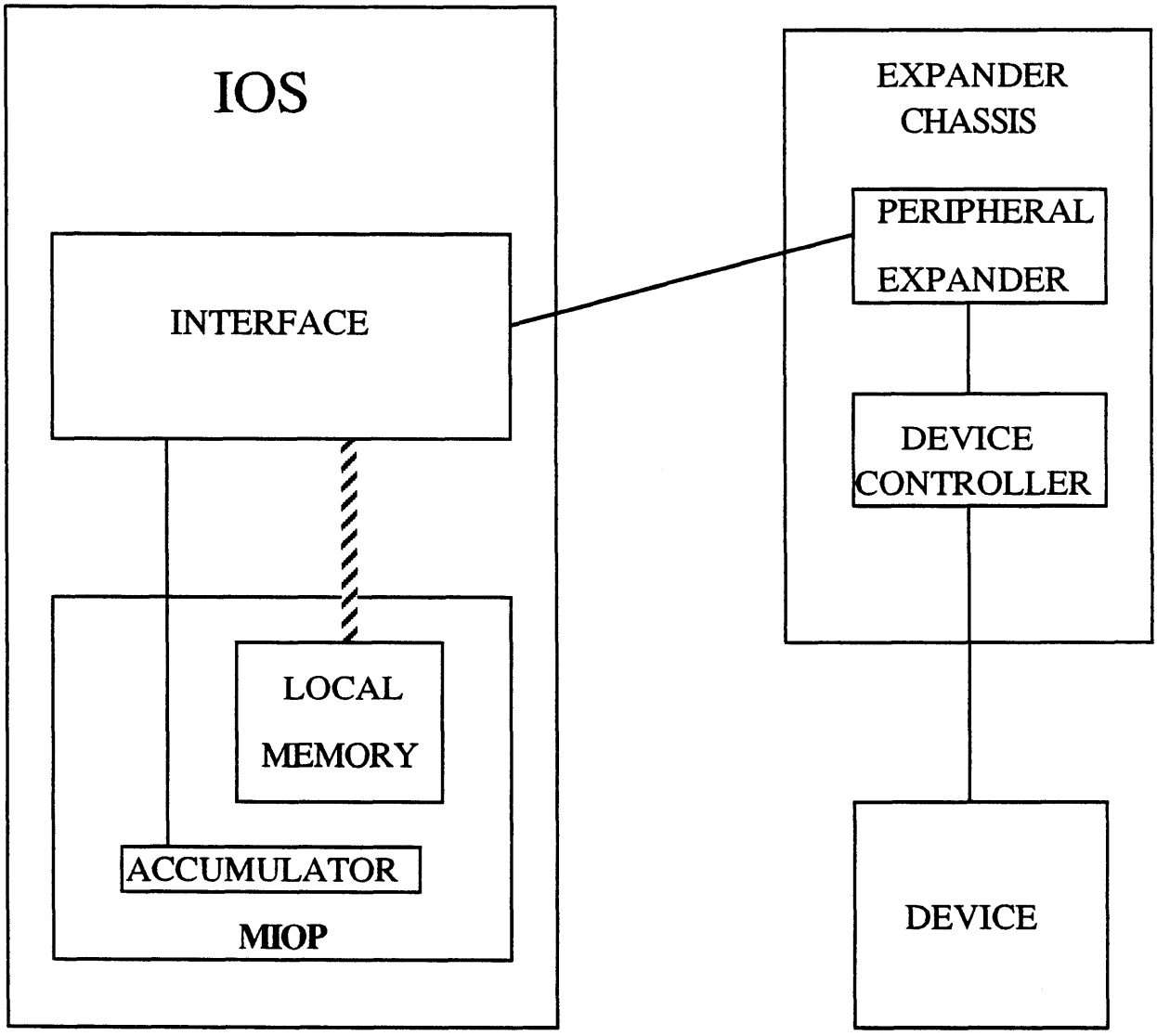


Figure I.3-15: Expander Channel

CONSOLE CHANNELS (accumulator channels)

In general, there are between 2-4 console channel pairs on the MIOP, and 1 on each of the other IOPs. One console on each IOP is the Kernel console for that IOP.

CONSOLE KEYBOARD CHANNEL TIA->TID

Console keyboard channels accept input from the keyboard one character at a time; a key depression generates an interrupt. There is one channel per console.

CONSOLE DISPLAY CHANNEL TOA->TOD

Console display channels send output to a display, one character at a time. There is one channel per console.

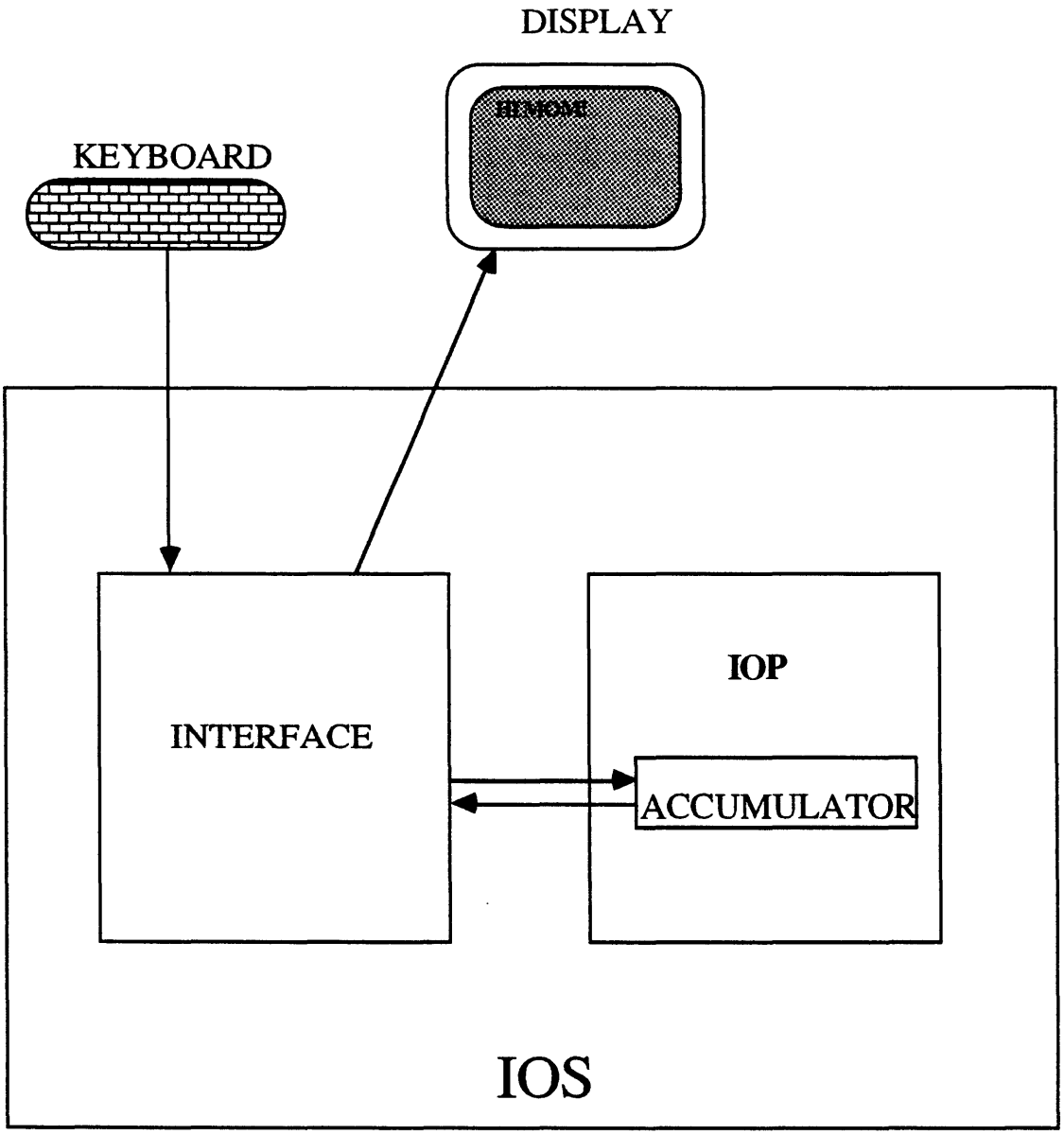


Figure I.3-16: Console Channels

ERROR LOGGING CHANNEL ERA (accumulator channel)

The error logging channel on the MIOP reports errors from the following sources:

- IOP Local Memories
- Buffer Memory
- Central Memory
- CPU memory channels

These sources are shown on the next page.

The MIOP logs the errors and sends an error logging packet to the CPU for inclusion in the COS or UNICOS system log. The MIOP also maintains a circular table in Buffer Memory of the last 512 errors reported on the channel.

In the newer I/O Subsystem (SN 21+) this channel is replaced by a central microprocessor-controlled logger. The logger is capable of operating with up to eight Cray devices. The logger software captures, formats and records errors from each of the devices. One of the main advantages of the logger is the ability to report the error that caused a system crash, even if the error channel is busy at the time of the crash.

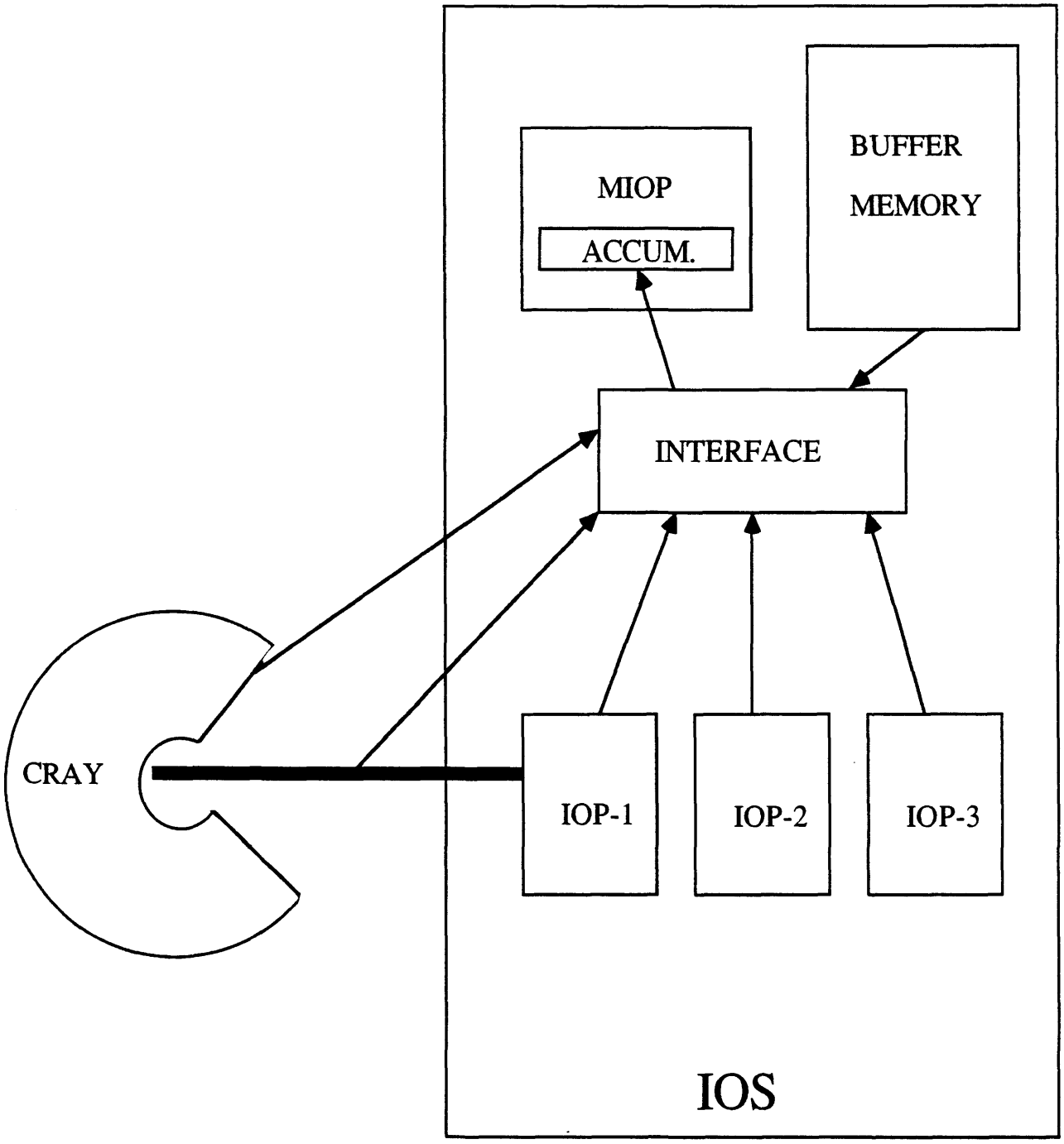


Figure I.3-17: Error Logging Channel

DISK CHANNEL DKA->DKP (DMA channels)

The disk channels transfer data to/from disk storage units. There are four channels per DCU-4 or DCU-5 controller, one disk per channel. Each controller takes up one DMA port.

The diagram on the next page shows one controller. A disk controller can drive four disks at one time. Each controller is separate from the others, all controllers can be transferring simultaneously barring Local Memory conflicts. The data transfer rate between the disk and the controller is 35.4 MBITS/S. Within the controller, data is double buffered for rapid data transfer.

For more information on disk channels, disk channel functions, disk controllers and disk storage units, see the HR-0077, Disk Systems Hardware Reference Manual.

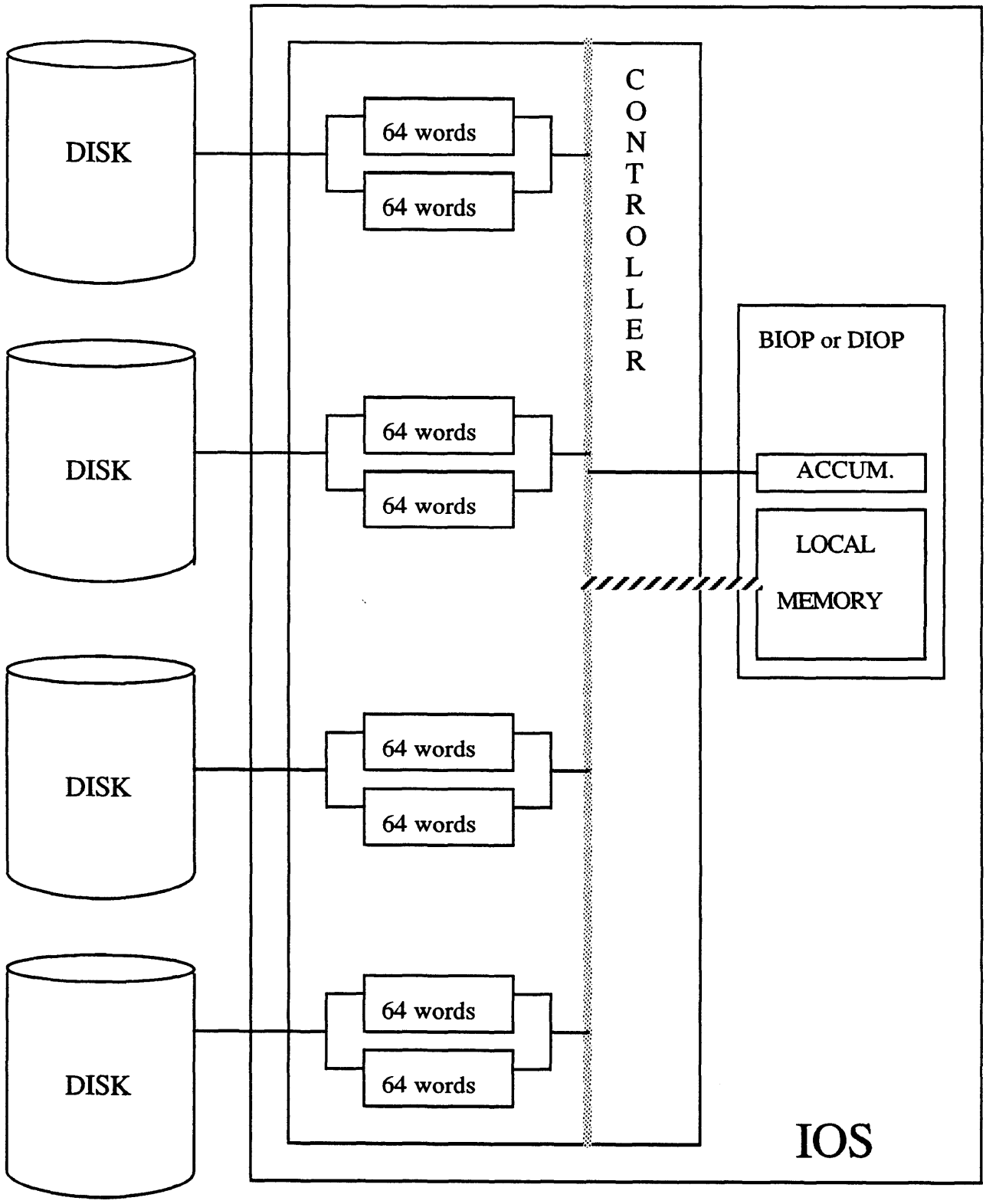


Figure I.3-18: Disk Controller with 4 Channels

BLOCK MULTIPLEXER CHANNEL BMA->BML (DMA channels)

The Block Multiplexer channel provides access to IBM plug-compatible peripherals (our software currently supports tapes). Block multiplexer channels are grouped into sets of four channels which share one BMC-4 block multiplexer controller. It is possible to attach many devices to each channel. Only one tape can be transferring per control unit at once. The transfer rate from tape to the controller is 1.2 MBYTES/S. Within the controller, the data is double buffered. When one buffer is transferring, the other is filling.

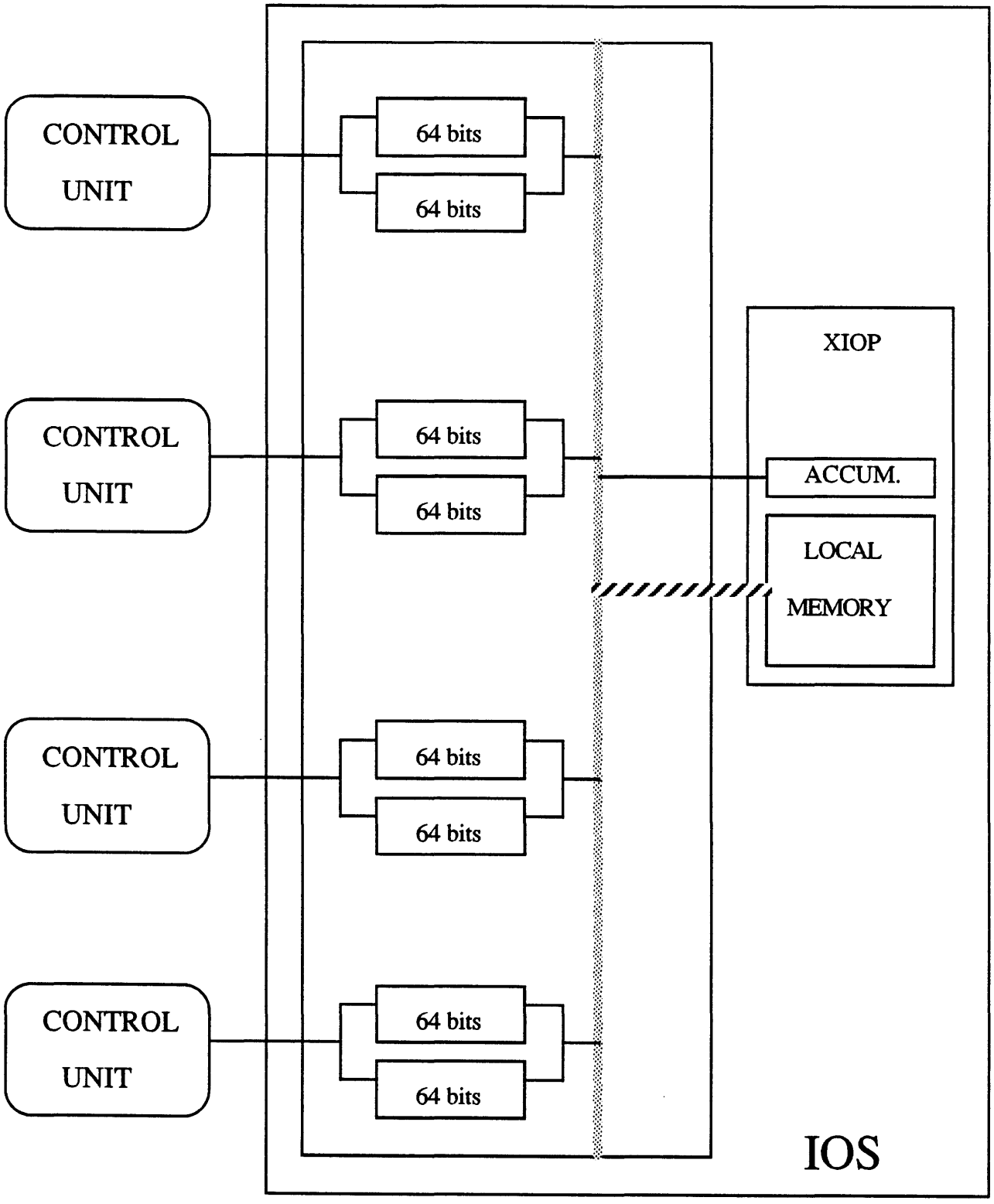


Figure I.3-19: BMC-4 Controller with 4 Channels

4 X 16 TAPE CONFIGURATION

The diagram on the next page shows one possible way to configure tapes. Each control unit can access any tape. As previously noted, only one tape can be transferring through a control unit at one time; but all control units can be transferring at once. The diagram is one possible configuration; there is no limit to the number of tapes configured.

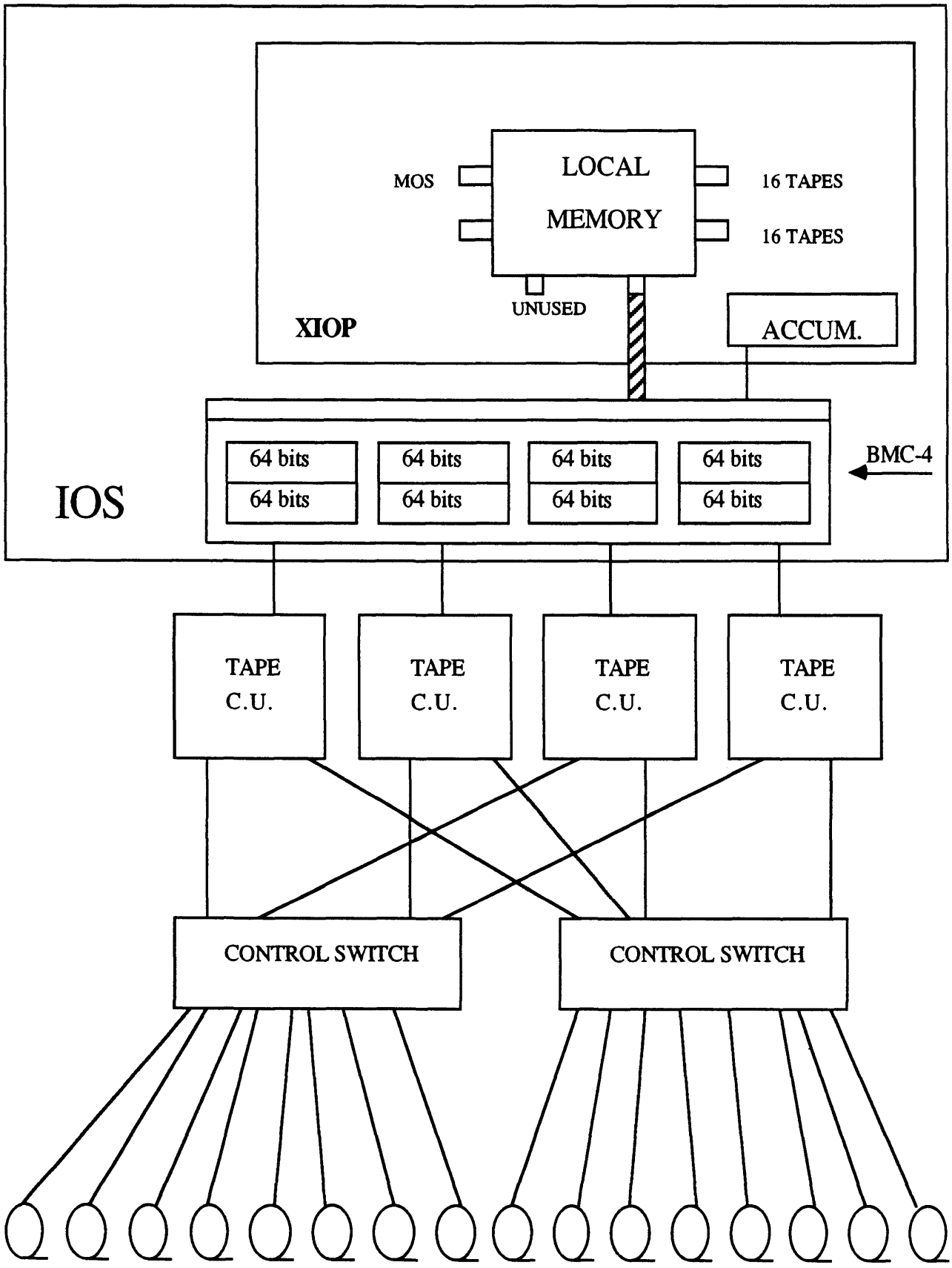


Figure I.3-20: 4 X 16 Tape Configuration

MAINFRAME/MAINTENANCE CHANNELS (DMA channels)

Each IOP has one channel pair for connecting to the Cray mainframe, or for interfacing to the Maintenance Control Unit (MCU).

MAINFRAME/MAINTENANCE INPUT CHANNEL LIA

This channel (channel 50) may interface to the MCU or to a Cray mainframe. When connected to a mainframe data is transferred in block moke directly into IOP Local Memory. Memory references are made in bursts of four parcels.

MAINFRAME/MAINTENANCE OUTPUT CHANNEL LOA

This channel (channel 51) may interface to the MCU or to a Cray mainframe. When connected to a mainframe data is transferred in block mode directly from IOP Local Memory vea a 6 MBYTE/S channel. Memory references are made in bursts of four parcels.

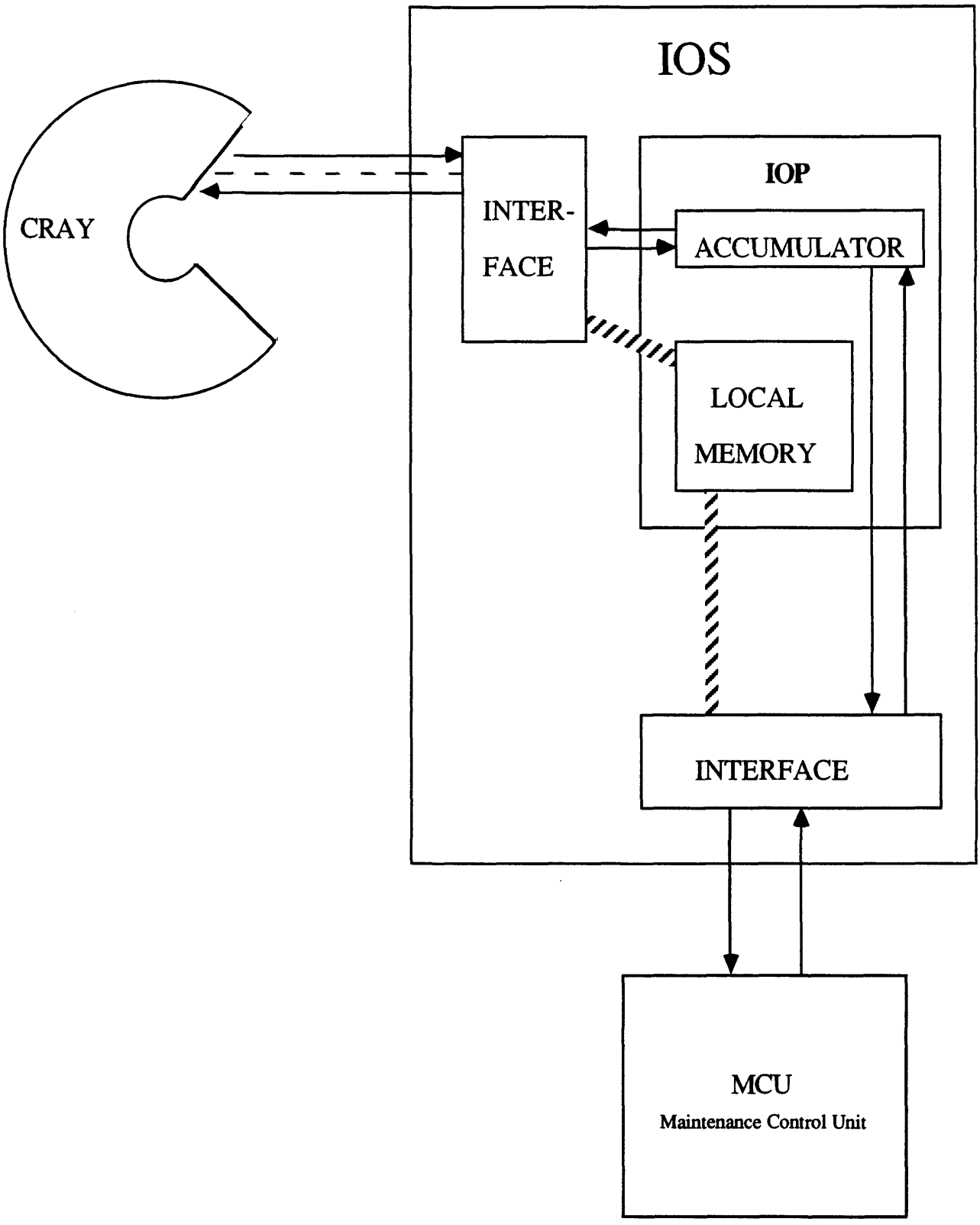


Figure I.3-21: Mainframe/Maintenance Channel

CHAPTER 3
REVIEW QUESTIONS

1. There are 12 standard (dedicated) channels on every IOP.

2. Accumulator channels pass 1 parcel of data per transfer.

3. DMA channels consist of an accumulator channel and Data Paths to Local Memory.

4. Channel states are monitored via Done and Busy flags.

5. Suppose system interrupts are disabled and the done flag sets on channels 17, 5 and 47. Assuming no other done flag sets, what is the value of the accumulator after each IOR:10 instruction in the following sequence of instructions?

```
IOR:10          A = 5
B=A
IOB:0
IOR:10          A = 17
B=A
IOB:0
IOR:10          A = 47
B=A
IOB:0
IOR:10          A = 0
```

6. Which of the following code sequences may result in an error? Why?

a)
```
I = 0
PASS
A = 7
PXS:14
A = 10000
PXS:15
```

b)
```
I = 1
A = 7
PXS:14
A = 10000
PXS:15
```

b. Could get a Error in address placement if a interrupt occurs.

7. Suppose interrupts are enabled in all I/O Processors and the Master I/O Processor (MIOP) executes an AOA:14 instruction. Does this instruction generate an interrupt in an I/O processor? If so, which one? yes. To the BIOP

8. False (True or False) A transfer from BIOP Local Memory to Central Memory results in an interrupt to the Cray on this channel when done.

9. True (True or False) The channel pair linking MIOP with the CPU is not normally used to transfer large amounts of data.

CHAPTER 3
ANSWERS TO REVIEW QUESTIONS

1. There are up to 12 standard (dedicated) channels in every IOP.

2. Accumulator channels pass 16 bits or 1 parcel of data per transfer.

3. DMA channels consist of an accumulator channel and data paths to Local Memory.

4. Channel states are monitored via Busy (BZ) and Done (DN) flags. The setting of a done flag creates an interrupt condition.

5. The IOR:10 instruction reads the highest priority (lowest number) interrupting channel number into the accumulator. The three channel numbers, by priority are 5, 17, 47. The first IOR:10 reads channel number 5 into the accumulator; that value is then loaded into the B register; so the next instruction clears its DN flag using the IOB:0 instruction. The sequence continues for the other channels until the last IOR:10 puts a zero value into the accumulator indicating all interrupts have been handled.

6. The code sequence in item (b) could be result in an error. With the interrupts enabled, the occurrence of an interrupt could put the address in the wrong place. Interrupts must be disabled when modifying the Exit Stack.

7. The MIOP is issuing a message to the BIOP, which is the highest priority IOP to the MIOP, with the instruction AOA:14. The message goes to the interface register which interrupts the BIOP to take the message out. (The BIOP would then issue an AIA:10).

8. False. The high speed memory channel does not interrupt the CPU when done. The CPU is only interrupted over the low speed control channel connected to the MIOP.

9. True. The low speed channel connecting the MIOP with the CPU is used for communication purposes only. Large data transfers occur over the 100 MBYTE/S channel(s) between an IOP and the CPU.

RELATED READING

The following related reading from HR-0081, IOS Model C Hardware Reference Manual, is highly recommended before proceeding with Chapter 4.

1. Chapter 2, pp. 2-17 to 2-23; "The IOP Input/Output Section"; and Chapter 4; "I/O Processor Channel Interfaces."

2. Appendix B, I/O Processor Programming Considerations;
   Appendix C, System Channel Assignments;
   Appendix F, Channel Functions; and
   Appendix G, Buffer Memory/Central Memory Bypass Mode.

3. Disk Systems Hardware Reference Manual, HR-0077

CHAPTER 4
I/O PROCESSOR LOCAL MEMORY

I/O PROCESSOR LOCAL MEMORY

Each I/O Processor has its own memory which is called Local Memory. Local Memory is separate from Buffer Memory which all IOPs share.

4.1 LOCAL MEMORY FUNCTIONS

The main functions of Local Memory are:

1. Holds nucleus of operating system.

   At the lower end of Local Memory in each I/O Processor is a copy of the Kernel. The Kernel controls all activities running in an I/O Processor.

2. Provides space for execution of IOS Overlay code.

   Because of the limited size of Local Memory, the operating system makes use of overlays. All overlays are stored in Buffer Memory; and are brought into Local Memory when needed to perform a specific function.

3. Provides I/O Buffers.

   Large volume I/O goes through an I/O Processor's Local Memory if that I/O Processor does not have a 100 MBYTE data channel to the Cray. For example, data on its way to Cray central memory from a tape attached to an XIOP that doesn't have a 100 MBYTE/S channel would take the following path:

   Tape --> XIOP LM --> Buffer Memory --> Cray Memory

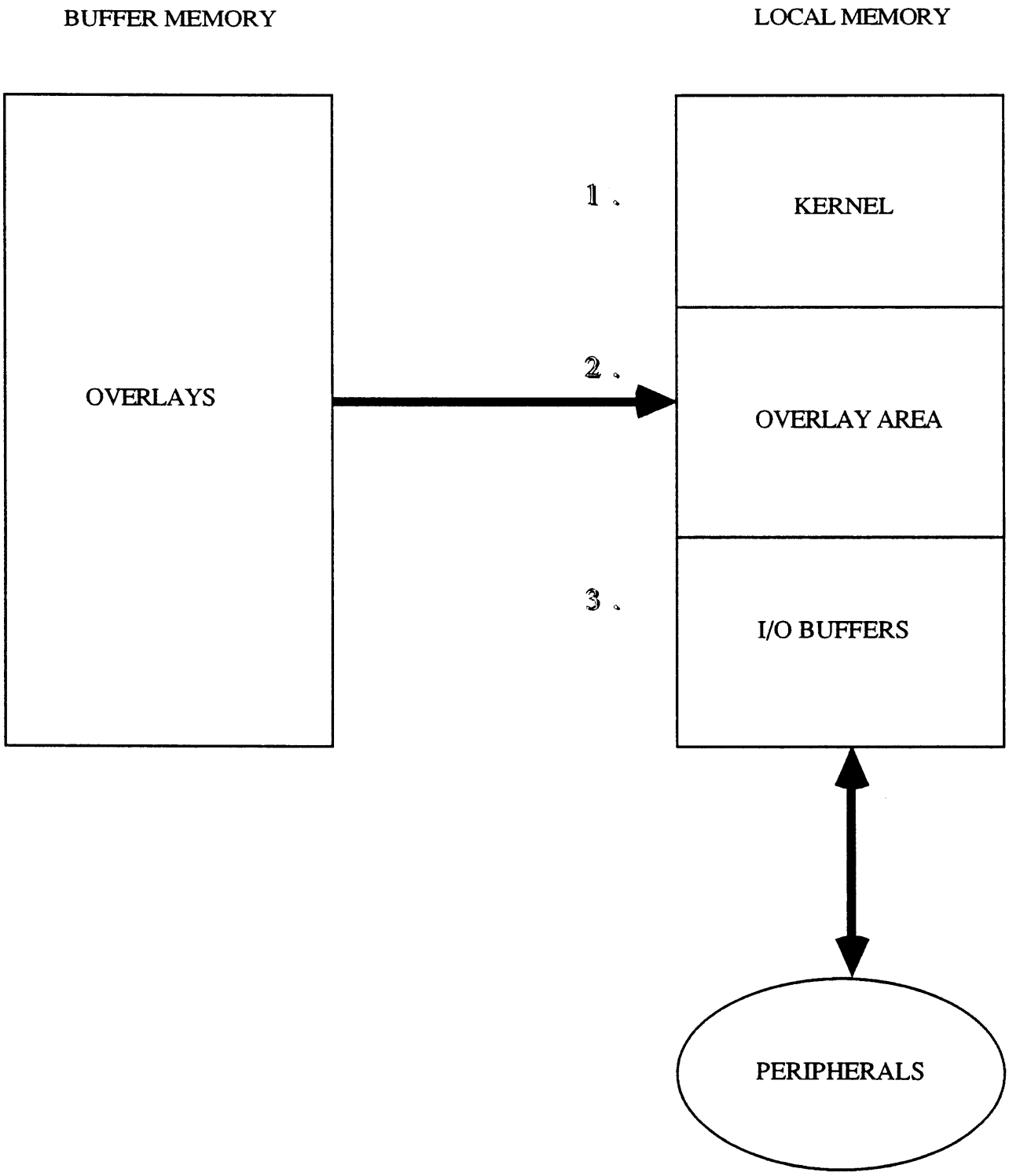


Figure I.4-1: Local Memory Functions

4.2 CHARACTERISTICS

Local Memory has a capacity of 65,536 16-bit units (parcels).

Local Memory is organized in 16 banks, 4 banks per section. A reference to any bank causes the entire section to be busy. There is a 4 CP busy time on a read, and a 6 CP busy time on a write operation.

Memory access is through six Direct Memory Access (DMA) ports. Access time to fetch an operand from Local Memory to the accumulator is 7 CPs. Instructions are fetched from Local Memory in 4 CP bursts, 1 parcel per clock period. An I/O reference moves 4 sequential parcels to/from an I/O channel.

Error Detection

The I/O Subsystem takes advantage of SECDED error correction; that is, single-bit error correction double-bit error detection. A 5-bit check byte is generated for each 8-bit data byte before the data is written to memory, and when the data is read from memory a new check byte is generated and compared to the original 5 bits.

There is a local memory refresh every 15 minutes to clear single bit errors. This is done by the software, where the Kernel resident area of Local Memory is written to Buffer Memory and read back in; and overlays are reloaded into Local Memory before being activated.

UPPER BYTE SECTION 0				SECTION	SECTION	SECTION
BANK 0	BANK 1	BANK 2	BANK 3	1	2	3
PARCEL 0	PARCEL 1	PARCEL 2	PARCEL 3			
				UPPER BYTE	UPPER BYTE	UPPER BYTE
LOWER BYTE SECTION 0				SECTION	SECTION	SECTION
BANK 0	BANK 1	BANK 2	BANK 3	1	2	3
PARCEL 0	PARCEL 1	PARCEL 2	PARCEL 3			
				LOWER BYTE	LOWER BYTE	LOWER BYTE

Figure I.4-2: Local Memory Layout

4.3 ADDRESSING AND ACCESS

Address Format

A parcel in Local Memory is addressed by 16 bits:

- the lower 2 bits select the bank
- the next 2 bits select the section
- the next 10 bits select the address within the chip
- the upper 2 bits select the chip

Address Paths

There are three address paths to each Local Memory section:

1. One from the I/O Section for the 6 DMA ports
2. One from the Computation Section for operand values
3. One from the Control Section (Fetch Register) to fetch instructions before executing

Access Paths

There are three read paths, and two write paths per Local Memory section:

1. One read and one write for the Accumulator
2. One read and one write for the I/O Section (6 DMA Ports)
3. One read to the Instruction Stack for instruction fetch

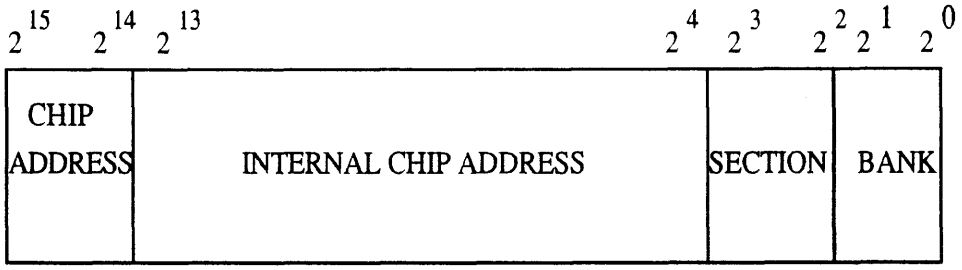


Figure I.4-3: Local Memory Address Format

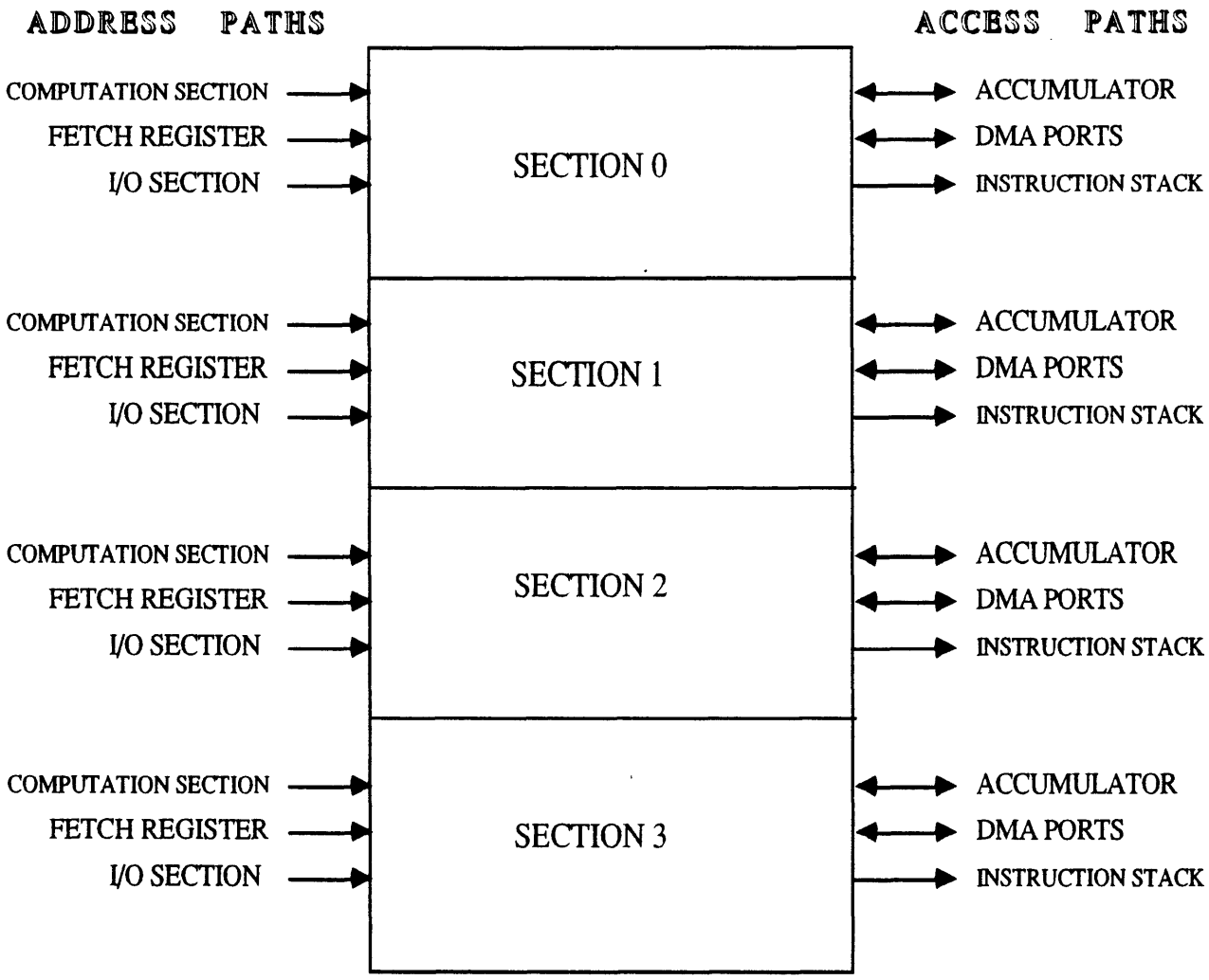


Figure I.4-4: Local Memory Address and Access Paths

CHAPTER 4
REVIEW QUESTIONS

1. List three functions of Local Memory
   a. Hold Nucleus of Operating System
   b. Space for Overlays
   c. I/O Buffers

2. There is/are 1 Buffer Memory(ies) and 3 Local Memory(ies) in a 3 processor IOS.

3. Up to 16 banks and 4 sections can be busy simultaneously.

4. True (True or False) Local Memory is protected by SECDED logic.

5. In what section and bank is Local Memory location 17234 (octal)?
   BANK 00
   Section 11

6. How many read and write paths are there from Local Memory to the six DMA ports? 4 read
   4 write

CHAPTER 4
ANSWERS TO REVIEW QUESTIONS

1. Three functions of Local Memory are:

   a. Provides I/O Buffers for data transfers
   b. Provides space for overlays to execute
   c. Holds a copy of the Kernel, the core of the Operating System

2. There is 1 Buffer Memory and 2 Local Memories in a 3 processor IOS. Each IOP has its own Local Memory; while all IOPs share Buffer Memory.

3. Up to 16 banks and 4 sections can be busy simultaneously. A reference to 1 bank causes all banks in that section to be busy. Each section has separate access paths; therefore, all four sections can be referenced at the same time.

4. True.

5. Section 3, Bank 0. The lower 4 bits select the bank and section of Local Memory.

6. There are 4 read and 4 write paths from Local Memory to the 6 DMA ports; 1 read and 1 write path per section.

RELATED READING

The following reading from HR-0081, IOS Model C Hardware Reference Manual, is highly recommended before proceeding with Chapter 5:

1. Chapter 2, pp. 2-24 to 2-27; "The IOP Memory Section (Local Memory)."

CHAPTER 5
BUFFER MEMORY

BUFFER MEMORY

5.1 FUNCTIONS

The main functions of Buffer Memory are:

1. Provides space for passing large messages between I/O Processors
2. Provides space for overlays used by the I/O Subsystem
3. Buffers data to/from disks, tapes and front end computers
4. Provides space for user job scratch datasets, called Buffer Memory Resident datasets

5.2 CHARACTERISTICS

Buffer Memory resides in the I/O Subsystem chassis. Buffer Memory's capacity is two million, four million or eight million words of 64 bits each. Data is refreshed every 4 milliseconds. Data protection is by single error correction/double error detection (SECDED). There is a 12 cp access time and a 26 cp bank busy time. There are four access ports into Buffer Memory, one for each I/O Processor.

Refer to Chapter 3 for the specific instructions for Buffer Memory access.

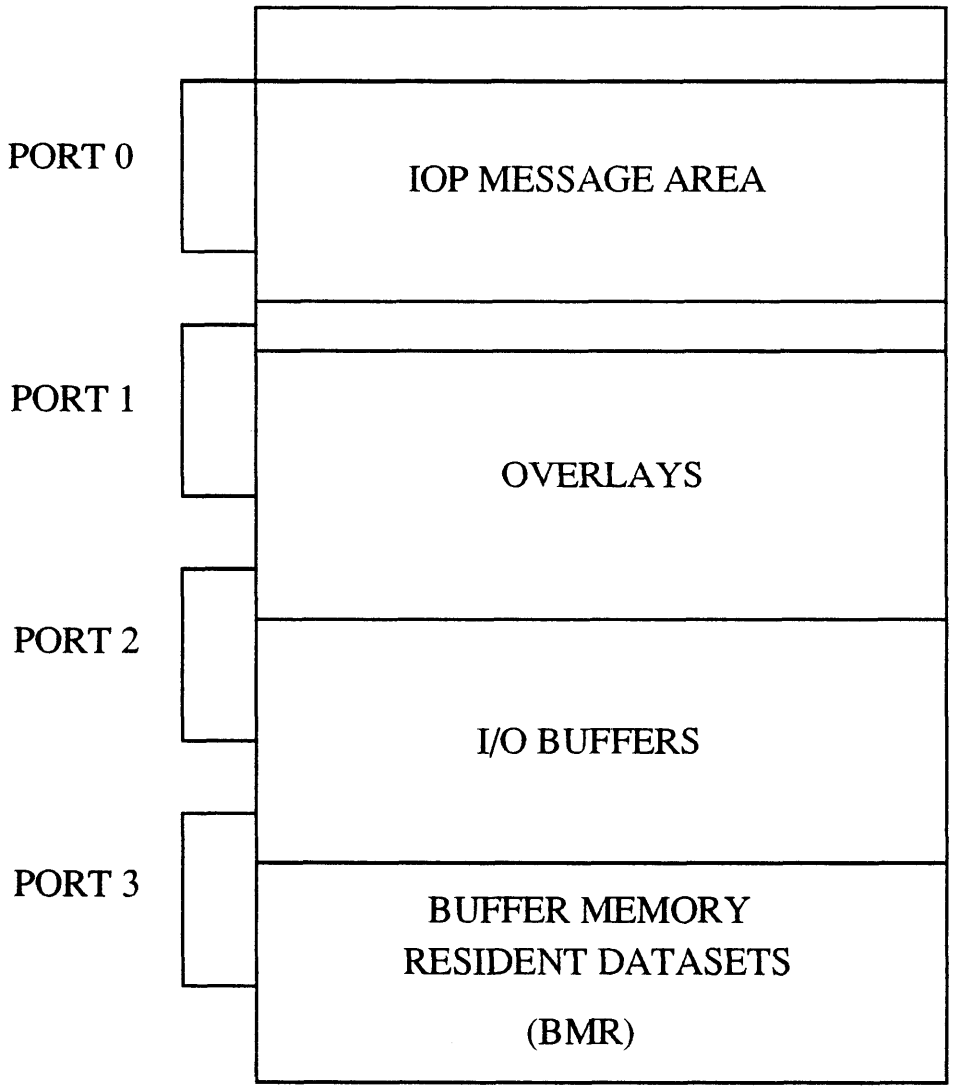


Figure I.5-1: Buffer Memory

CHAPTER 5
REVIEW QUESTIONS

1. List three functions of Buffer Memory.

   a. Provides space for passing large messages between IOP's

   b. SPACE FOR THE IOS OVERLAYS.

   c. Buffer data to/from disks, tapes, & front-end computers.

2. There are 4 ports into Buffer Memory, one for each IOP.

3. There is a 12 cp access time into Buffer Memory and a 26 cp bank busy time.

4. True (True or False) Buffer Memory is protected by SECDED logic.

CHAPTER 5
ANSWERS TO REVIEW QUESTIONS

1. The functions of Buffer Memory are:

   - provides space for large IOP messages
   - provides space for overlay storage
   - provides I/O buffers for data transfers
   - provides space for Buffer Memory Resident (BMR) datasets

2. There are 4 ports into Buffer Memory, one for each I/O Processor.

3. There is a 12 cp access time and a 26 cp bank busy time for Buffer Memory.

4. True. Buffer Memory is protected by the same logic as on the Cray - SECDED.

RELATED READING

The following reading from HR-0081 IOS Model C Hardware Reference Manual, is highly recommended for the successful completion of Chapter 5.

1. All of Chapter 3; "Buffer Memory."

APPENDIX A
I/O PROCESSOR DETAILED BLOCK DIAGRAM

I/O PROCESSOR DETAILED BLOCK DIAGRAM

The following hardware registers are not visible to the programmer, but are a part of the Instruction Control Network (refer to HR-0081, IOS Model C Hardware Reference Manual, Chapter 2 for a complete description of these registers):

- RP (Register Pointer) Register
- DP (Destination Pointer) Register
- Fetch Register
- Branch Accumulator
- Branch Addend Register
- MA (Memory Address) Register

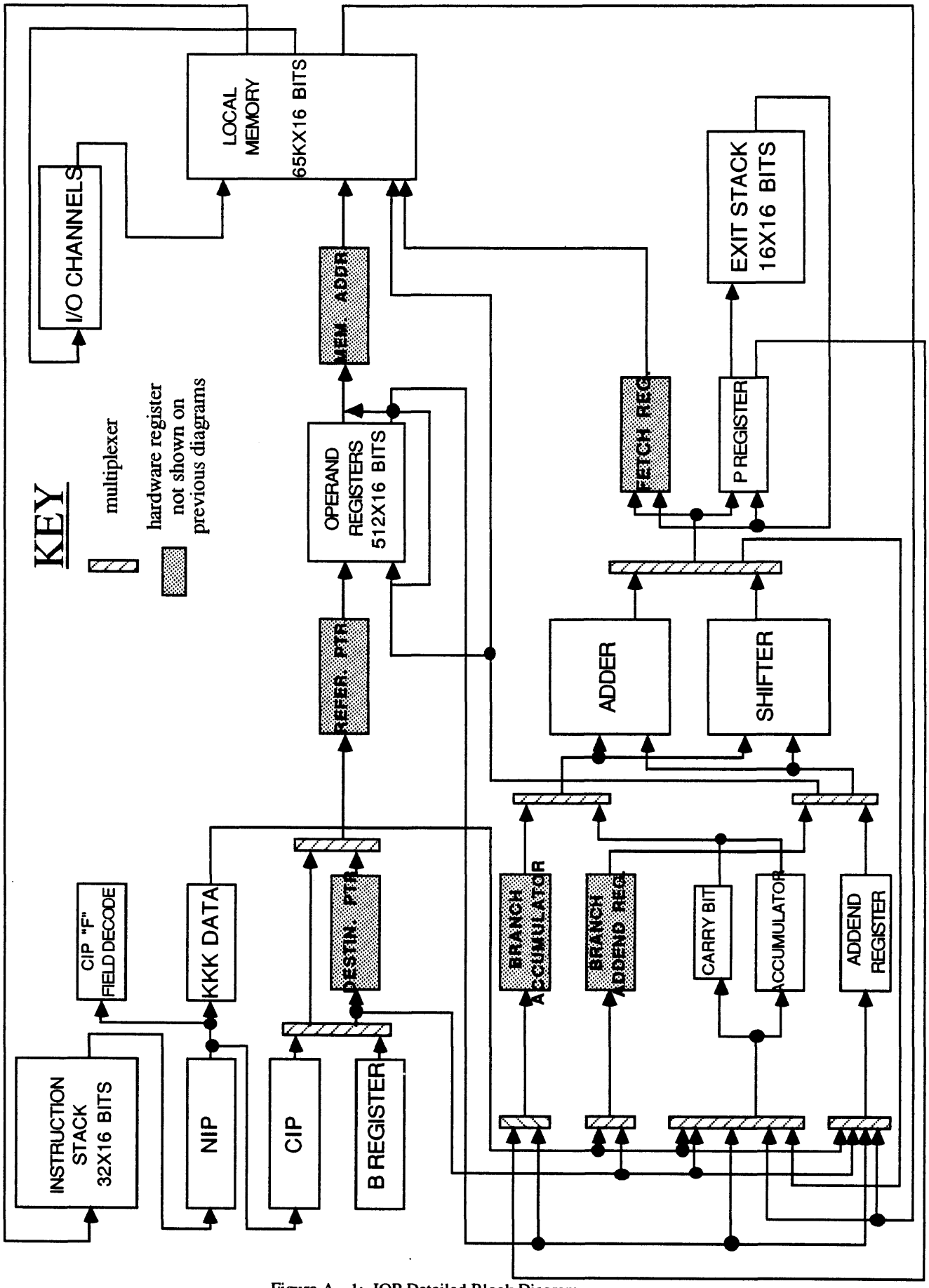


Figure A - 1: IOP Detailed Block Diagram

APPENDIX B
INDIVIDUAL I/O PROCESSOR I/O CHANNELS

I/O PROCESSOR I/O CHANNELS

The next four pages illustrate the conventional channel assignments on each I/O Processor; both the standard, or dedicated, channels as well as the IOP specific channels are shown.

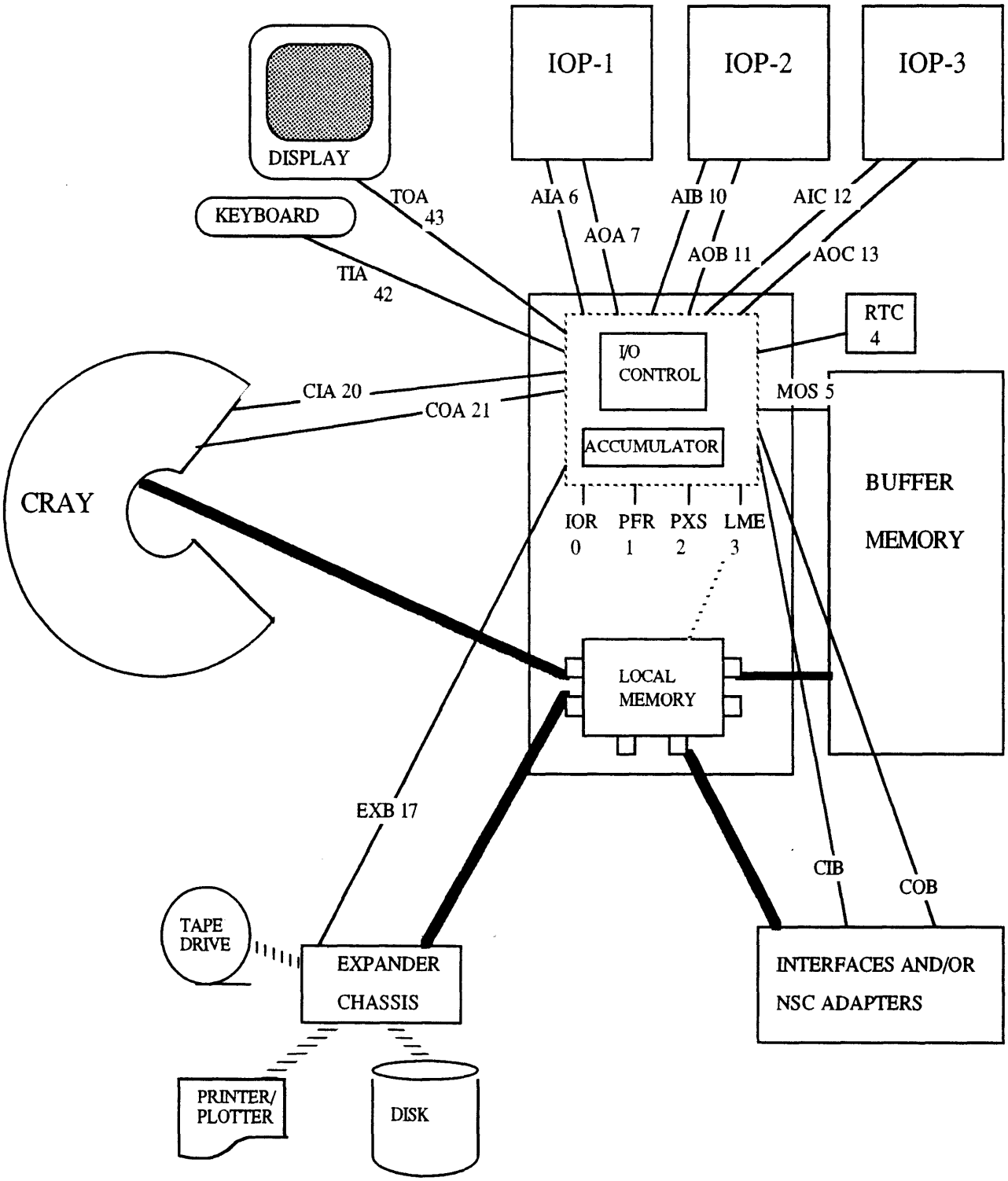


Figure I.B-1: MIOP I/O Scheme

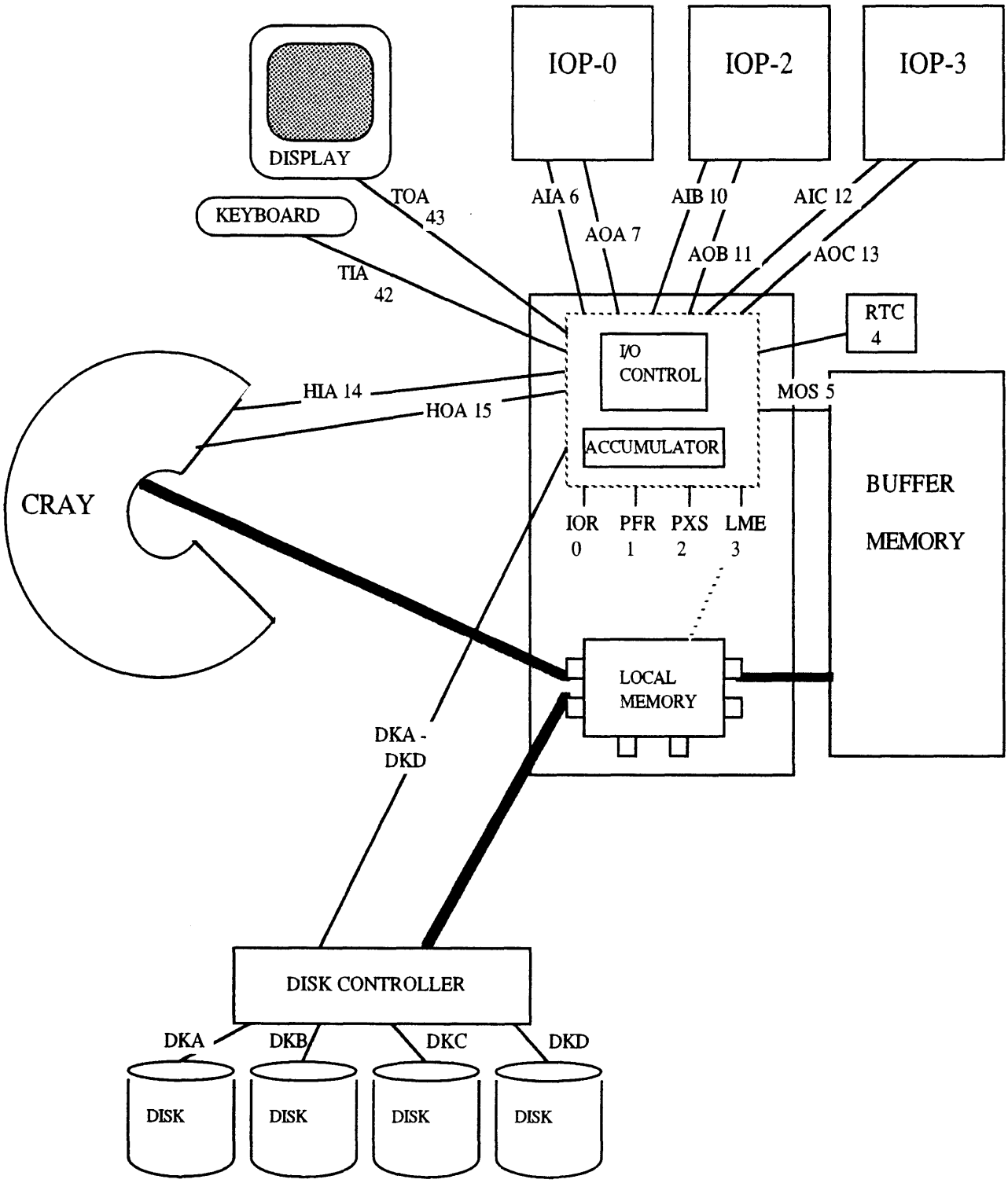


Figure I.B-2: BIOP I/O Scheme

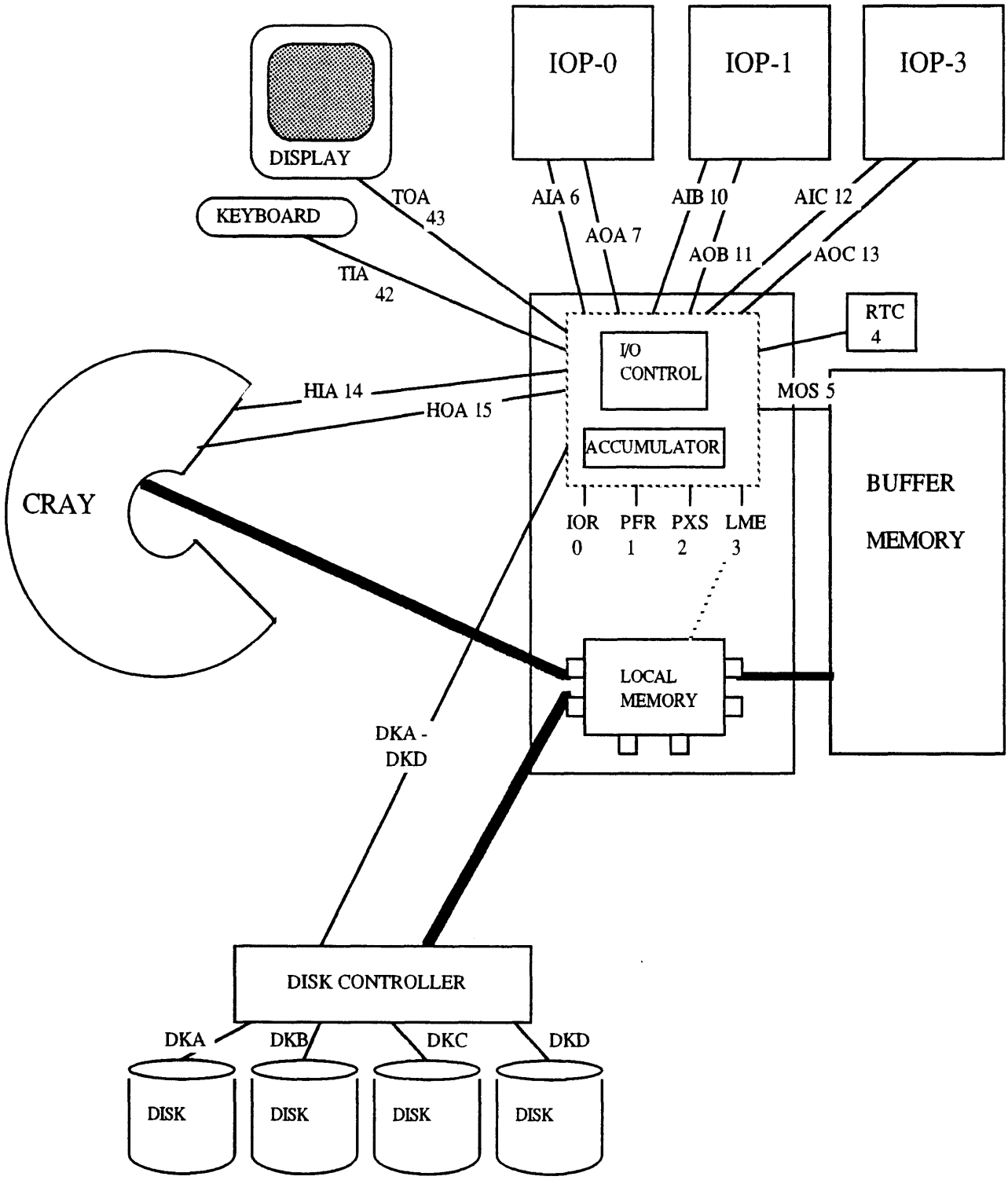


Figure I.B-3: DIOP I/O Scheme

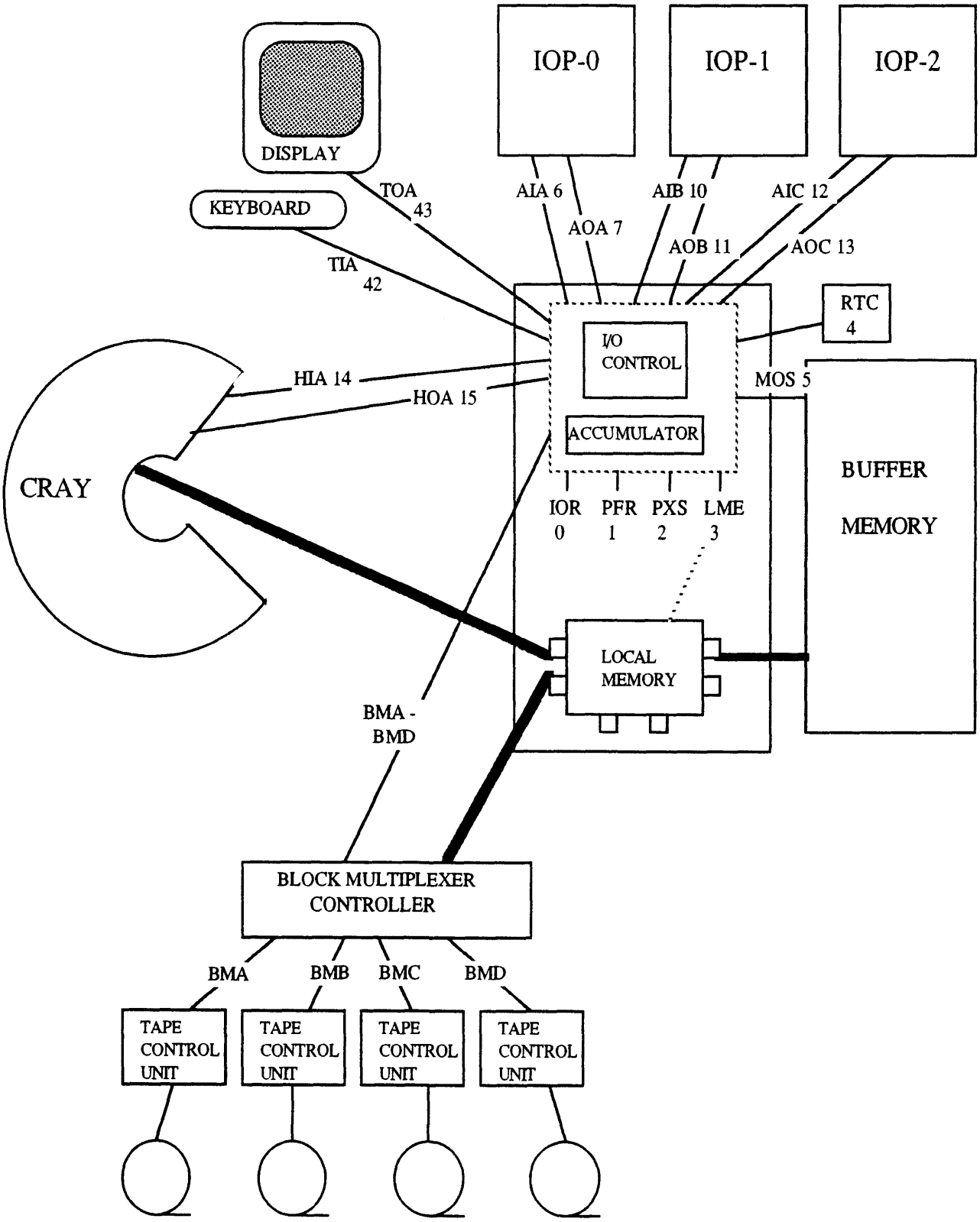


Figure I.B-4: XIOP I/O Scheme

READERS COMMENT FORM

I/O SUBSYSTEM ARCHITECTURE WORKBOOK TR-IA

Your comments heop us to improve the quality and usefulness of our materials. Please use the space provided below to share with us your comments. When possible, please give specific page and paragraph references.

NAME _____
JOB TITLE _____
FIRM _____
ADDRESS _____
CITY _____STATE _____ZIP _____

CRAY RESEARCH, INC.

TECHNICAL TRAINING & DEVELOPMENT

Cray Research, Inc.
Software Training
2520 Pilot Knob Road, Suite 300
Mendota Heights, MN 55120

Cray Research, Inc.
Hardware Training
890 Industrial Blvd.
Chippewa Falls, WI 54729