WANG

6200

VS I/O STRUCTURE AND THE IOP MOTHERBOARD

Models:

209-7110
209-7810

Customer Engineering
Product Maintenance Manual

741-0824

PREFACE

Every I/O interface on a VS 80, 85, 90 or 100 system consists of an IOP Motherboard (IOPMB) as a motherboard and a Device Adapter (DA) as a daughterboard. This manual covers the way in which the operating system and the peripheral device communicate with each other. Information about specific DA daughterboards is not provided here.

There are several other publications related to this document. The VS Data Processing Field Guide, part number 729-1265-A, is a handbook summarizing the operating system information which is covered here in chapter 2. The VS 80 Reference Guide, part number 741-0716, is a handbook containing IOP part numbers for VS 80 machines. The VS Service Handbook, part number 729-1098-A, is a handbook containing part number information for VS 85, 90 and 100 IOPs.

This manual obsoletes 729-0824 and its update 729-0941.

Proprietary Notice

This document is the property of Wang Laboratories, Inc. Information contained herein is considered company proprietary information and its use is restricted solely to the purpose of assisting you in servicing Wang products. Reproduction of all or any part of this document is prohibited without the prior consent of Wang Laboratories.

REVISION STATUS

DIAGNOSTICS HANDBOOK

VS I/O STRUCTURE AND THE IOP MOTHERBOARD

PAGES	REVISION: DATE
Front Matter	Original Issue: 09/23/84
1-1 to 1-5	Original Issue: 09/23/84
2-1 to 2-17	Original Issue: 09/23/84
3-1 to 3-21	Original Issue: 09/23/84
4-1 to 4-6	Original Issue: 09/23/84
A-1	Original Issue: 09/23/84
B-1 to B-8	Original Issue: 09/23/84

TABLE OF CONTENTS

VS I/O STRUCTURE AND THE IOP MOTHERBOARD

USER COMMENTS AND FEEDBACK

VS I/O STRUCURE AND THE IOP MOTHERBOARD
741-0824

Please help us provide the best publication possible by using this comment sheet when you find things you'd like to see corrected, changed or added to this document. Attach copies of the marked-up pages, if possible, and mail to:

WANG LABORATORIES, INC.
437 South Union Street
Lawrence, Massachusetts 01843
USA

Attention:
Technical Training & Documentation
mail stop 8237

Your Name:____

Phone:__________________________ RDB:____________

Address:____

____

Comments:_______________________________________

____

____

____

____

____

____

____

____

____

CHAPTER 1
OVERVIEW

INTRODUCTION

The operating system of a computer is the entity which manages system resources and assigns those resources to the programs and applications running under it. One of the resources controlled by the operating system is the I/O subsystem. The I/O subsystem consists of all the devices attached to the system, of their interfaces and controllers, and of the manner in which they communicate.

This manual was written to help CE understand the the I/O subsystem in both intermediate and large VS systems. It explains the protocol used in communications between the CP and the I/O devices. It also provides an overview of the hardware involved in this information exchange. By discussing both software and hardware issues, this manual attempts to show a complete picture of I/O operations.

Chapter 1	introduces the manual itself and provides an I/O overview.
Chapter 2	describes how the operating system communicates with the I/O Processors (IOPs).
Chapter 3	provides a block diagram description of the IOP Motherboard (IOPMB).
Chapter 4	describes how the IOPMB communicates via each of its three interfaces.
Appendix A	provides the instruction set used in the PROMs on the IOP.
Appendix B	provides schematics for the two IOPMBs covered by this manual.

ACRONYMS

The following acronyms are used for elements of the I/O subsystem of VS machines.

Acronym	Definition
IOPMB	IOP Motherboard.
CIO	Control I/O instruction (privileged).
CP	Central Processor.
CP3	Central Processor of a VS 80 system.
CP4	Central Processor of a VS 85, 90 or 100 system.
CP5	Central Processor of a VS 15, 25 or 45 system.
DA	Device Adapter.
DLI	Device-Level Interface.
DLP	Device-Level Processor, i.e., the microprocessor which is the intelligence of the device itself (often a Z80), not the IOPMB's Microprocessor logic.
IOCA	I/O Command Area or I/O Command Address.
IOCT	I/O Command Table.
IOCW	I/O Command Word.
IOP	I/O Processor, a two-board assembly consisting of an IOP Motherboard and a Device Adapter.
IOPMB	I/O Processor Motherboard.
IOSW	I/O Status Word.
HIO	Halt I/O instruction (privileged).
PCB	Processor Communications Bus.
SIO	Start I/O instruction (privileged).
XIO	Execute I/O instruction.

IOP FUNCTIONS

Input/Output Processors are used to control input/output requests to both serial and parallel devices in intermediate and large VS computer systems (VS 80/85/90/100). The position occupied by the IOP in the systems is shown in Figures 1-1 and 1-2.

The operating system executing in the central processor (CP) starts an IOP's tunctions, but, once started, the IOP is able to perform the operation independently of the CP. Meanwhile, the CP places the program requiring I/O on hold and executes some other program. In this manner, the IOP and the CP perform concurrent processing.

Every IOP consists of a IOP Motherboard as a motherboard and a Device Adapter as a daughterboard. The IOPMB interfaces to the system buses and translates CP commands into functions for the DA. The DA furnishes the direct link to any attached devices.

There are two types of motherboards used in IOPs. The first model is the 209-7110 board. The second model is the 209-7810 board.

The difference between IOPMB models is in how many addresses each can handle. Both IOPMB boards handle up to 16 devices. On the 7110 IOPMB, these 16 devices may have only 16 different addresses. On the 7810 IOPMB, however, these 16 devices may have up to 32 addresses. This means that the 7810 may handle devices like archiving workstations which have separate addresses for the floppy disk drive and the workstation itself. The 7110 can handle only regular workstations which do not need a second address.

Figure 1-1
Printed Circuit Boards in a VS 80

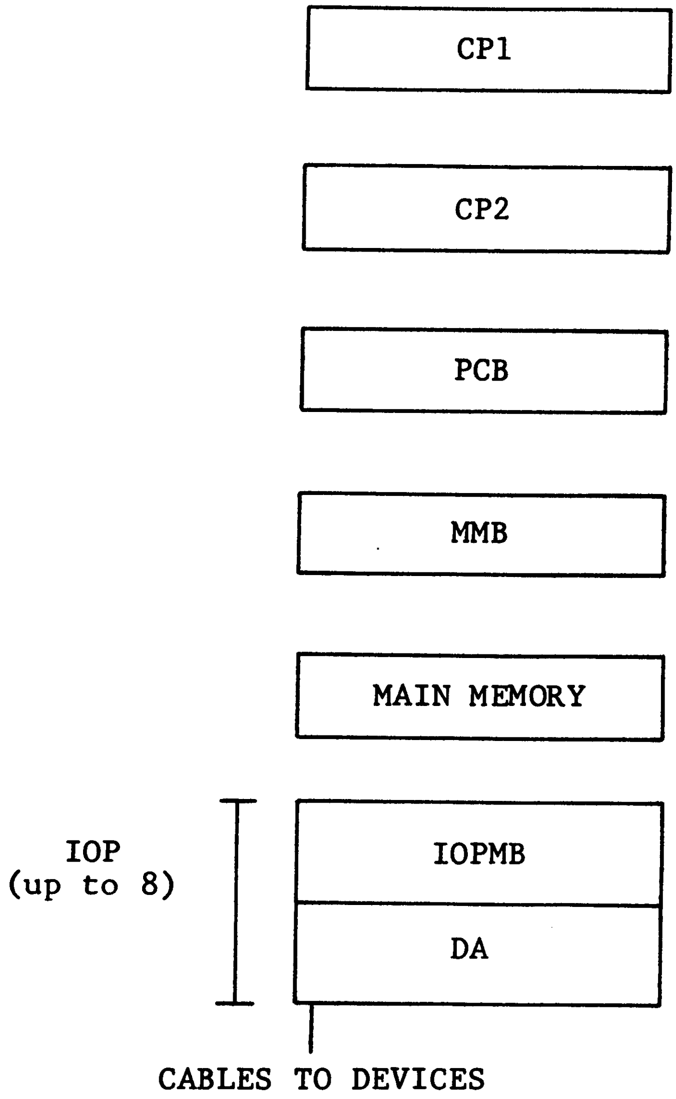
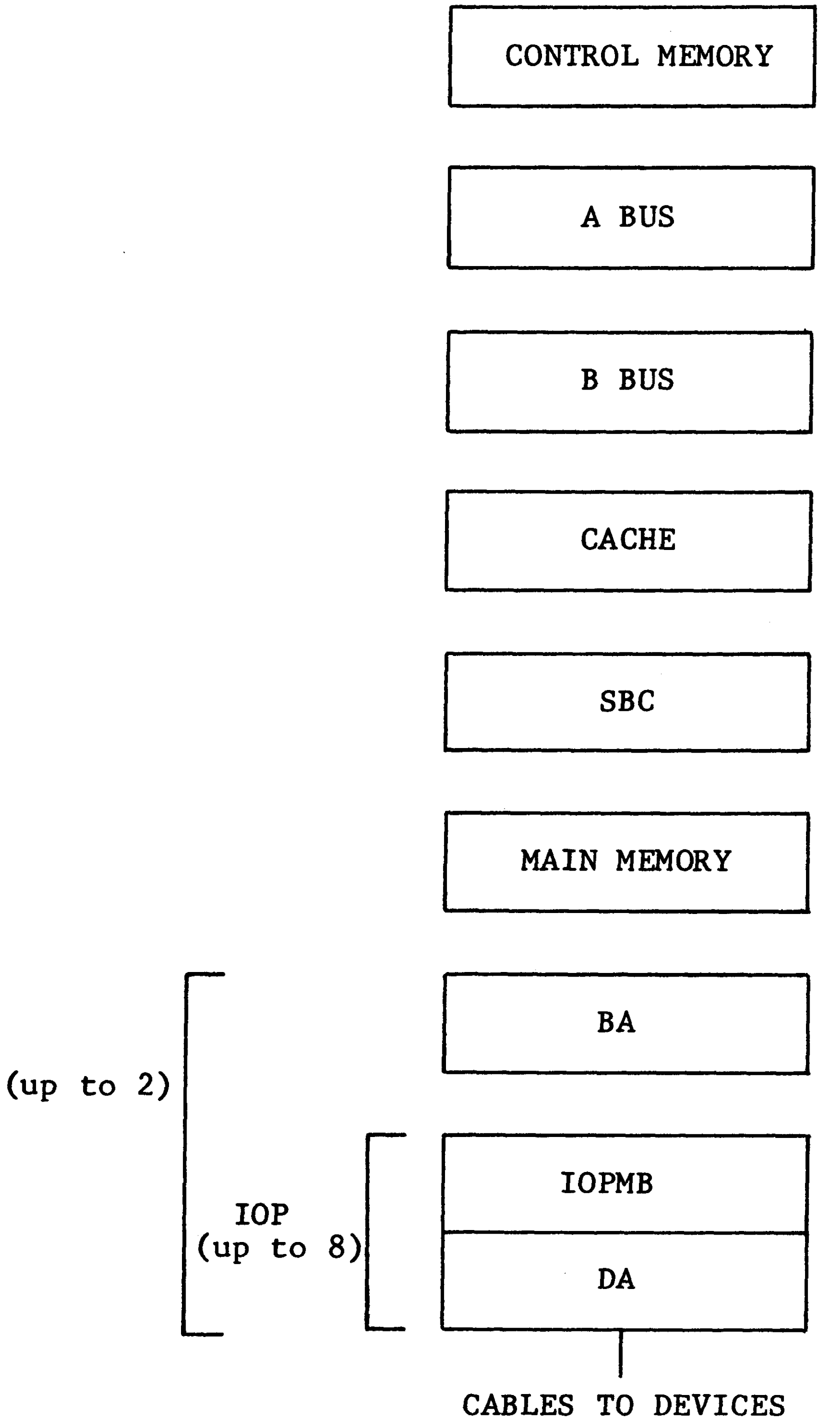


Figure 1-2
Printed Circuit Boards in a VS 100



CHAPTER 2
I/O PROTOCOL

OVERVIEW

The protocol of the I/O subsystem consists of two things:

- the format in which commands and responses are communicated and

- the order in which operations are performed.

This chapter concentrates on the format in which commands and responses are communicated. Chapter 4 discusses the order in which I/O operations are performed.

The protocol established between the VS operating system and the IOPs hinges on the I/O Command Word (IOCW) and the I/O Status Word (IOSW). The CP and all IOPs in the system use the same IOCW/IOSW structure when executing the I/O instructions.

This chapter discusses:

- the format of the IOCWs,

- the format of the IOSWs, and

- the privileged I/O instructions.

REFERENCE DOCUMENTATION

- VS Principles of Operation, part number 800-1100PO-04, describes the SIO, CIO and HIO instructions and the algorithms for memory addressing. (RR addressing is used.)

- The VS Data Processing Field Guide, part number 729-1265-A, is a small handbook summarizing the operating system.

IOCW

IOCW Contents: The IOCW is generated by the VS operating system to communicate to the IOP specific information about the requested I/O operation. This information consists of four fields of data. The four data fields are:

- Command Code: the type of operation to be performed.

- Data Address: the starting address for main memory data.

- Data Count Field: the number of bytes of data to be transferred.

- Device-Dependent Section: various types of information, depending on the device.

BYTE	0	1	2	3	4	5	6	7	8
	CC	Data Address			Data Count		Device-Dependent		

IOCW Storage: Once the operating system has generated an IOCW, it places the IOCW in main memory. To aid the IOP in locating the IOCW, the operating system then leaves this storage address at another place in main memory. This place is called the I/O Command Area or the I/O Command Address. Both are abbreviated as the IOCA.

The address of this address storage place is known to the IOP involved: it is a function of the device address. The IOP goes to this preliminary address to find the pointer to the actual I/O Command Word. The particular protocol about where to store the storage address depends on the CP in which the operating system is executing.

DISK IOCW FORMAT

0	1	2	3	4	5	6	7	8
CC	Data Address			Data Count		Sector Address		

Byte	Hex Value	Description
0	Command Code:	
	00	Fixed platter.
	01	Removable platter.
	02	Indirect addressing is used in the Data Address field.
	04	Suppress retry on errors.
	08	Use Diagnostic Mode.
	40	Read.
	80	Write.
	A0	Write/verify.
	C0	Seek.
	E0	Format.
1-3	Data Address:	
	0-512K	Main memory physical address of data area. If indirect addressing is in use, this contains the address of the start of the Indirect Address List.
4-5	Data Count:	
		Number of bytes to be transferred (in hexadecimal), depends on the type of disk:
	100	Ten-Meg drives, floppy drives.
	800	75/288 Meg drives, 30/60/90 Meg drives.
6-8	Device-Dependent: Sector Address	
		Sector at which transfer starts, ranges from 0 to maximum for that particular disk.

TAPE IOCW FORMAT

BYTE	0	1	2	3	4	5
	CC	Data Address			Data Count	

Byte	Hex Value	Description
0	**Command Code:**	
	01	Selects low increment values. Can be added to other values below.
	02	Selects indirect addressing for Data Address Field. Can be added to other values below.
	40	Read data.
	80	Write data.
	C0	Sense.
	D0	Erase.
	D4	Write tape mark.
	D8	Forward space block.
	DC	Forward space file.
	E0	Rewind.
	E4	Rewind and unload.
	E8	Back space block.
	EC	Back space file.
	F0	Kennedy: Set density high. Telex: Set PE mode.
	F4	Kennedy: Set density low. Telex: Set NRZI mode.
	F8	Kennedy: Set parity odd. Telex: Set GCR mode.
	FC	Kennedy: Set parity even. Telex: Set drive-selected mode.
1-3	**Data Address:**	
	0-512K	Main memory physical address of data area. If indirect addressing is in use, this contains the address of the start of the Indirect Address List.
4-5	**Data Count:**	
		Number of bytes to be transferred.
6-8	**Device-Dependent: Not used for tapes.**	

SERIAL IOCW FORMATS

BYTE	0	1	2	3	4	5	6	7	8
	CC	Data Address			Data Count		Device-Dependent		

Byte	Hex Value	Description
0	Command Code: Interpretation of this area by the IOP depends on whether it is passed to the IOP under an SIO or a CIO instruction. The two possibilities are listed separately below.	
0 for SIO:		
	40	Read data from the device.
	44	Read the tab settings.
	48	Read the diagnostic code.
	4C	7810 IOPs: Interrogate.
	50	Read altered.
	80	Write data to the device.
	84	Write the tab settings.
	90	Write selected.
	CO	Control device. When this command is issued, the rest of the IOCW is passed on to the device for use as an immediate operand. Do not use any DMA operations when using the CO option.
0 for CIO:		
	40	Read the Device Routing Table from the Device-Level Processor to main memory. 1024 bytes total.
	60	Read the device's microcode.
	80	Load the Device Configuration Table from main memory into the Device-Level Processor. 64 bytes total.
	90	Load the Device-Level Processor microcode. This microcode resides not on the IOP but in the peripheral itself.
	AO	Load device microcode.
	A8	Read the Device Routing Table from the Device-Level Processor to main memory. 1024 bytes total.

Serial IOCWs, continued

Byte	Hex Value	Description
0	Command Code for CIO, continued:	
	C0	Reinitialize the Device-Level Processor and restart the IOP.
	D0	Restart the Device-Level Processor.
	E0	Restart the Device Processor.
	F0	Control resources.
1-3	Data Address:	
	0-512K	Main memory physical address of data area. If indirect addressing is in use, this contains the address of the start of the Indirect Address List.
4-5	Data Count:	
		Size of the valid data buffer for this I/O request. The device is allowed an unlimited number of accesses to this buffer. The only requirement is that it stay within the boundaries defined by this field and the Data Address field.
6-8	Device-Dependent:	
	For SIO commands: passed unaltered to device. Can be used as anything desired.	
	For CIO commands: Device RAM Address.	

IOSW

IOSW Contents: The IOP generates an IOSW to communicate to the CP the outcome of the I/O operation. Each IOSW can have up to 8 bytes of information. This information consists of five fields of data as follows:

- General Status Byte: indicates whether the operation was successfully completed or if a hard or soft error occured.
- Error Status Byte: indicates the type of error which occured.
- Device-Dependent Status Bytes: vary according to device type.
- Residual Byte Count: tells the operating system how much of the job remained to be done at the time the IOSW was logged.
- Device-Dependent Extension: varies both in length and in meaning by device type.

BYTE	0	1	2	3	4	5	6	7
	GS	ES	DDS		Byte Count		Dev-Dep. Ext.	

IOSW Storage: Once the operation is complete, the IOP stores the IOSW in main memory. The address at which it is stored depends on the type of CP in which the operating system is executing.

For a CP3-based system, the IOSW is written into main memory beginning at location 0000 0000.

For a CP4-based system, the IOSW is written into an operating system table. Then, as soon as the IOP notifies the operating system that the operation is completed, the operating system re-writes this IOSW into location 0000 0000.

DISK IOSW FORMAT

BYTE	0	1	2	3	4	5	6	7
	GS	ES	DDS		Byte Count		Retry Info.	

Byte	Hex Value	Description
0	General Status:	
	80	IRQ: Intervention required.
	40	NC: Normal completion.
	20	EC: Error Completion. When both NC and EC are set, a soft error occured. When only EC is set, the error was a hard error.
	10	U: Unsolicited interrupt.
	08	PC: IOP/DLP now ready.
	04, 02, 01	Reserved for future use by the operating system.
1	Error Status:	
	80	IC: Invalid command.
	40	MPE: Memory parity error.
	20	MAE: Memory address error.
	10	DM: Device malfunction.
	08	DAM: Device memory error, or device damaged.
	04, 02, 01	Reserved for future use by the operating system.
2-3	Device-Dependent Status Information:	
2	80	HDF: Header reformatted.
	40	HDS: Header skipped.
	20	DIA: Diagnostic mode.
	10	OPT: Optimization was used.
	08	IDA: Invalid disk address.
	04	IDC: Invalid data count.
	02	SO: Sector overrun.
	01	SI: Seek incomplete.

Disk IOSW Formats continued

Byte	Hex Value	Description
2-3	Device-Dependent Status Information continued	
3	80	WP: Write-protected.
	40	NRO: Disk not ready.
	20	ST: Sector time-out.
	10	DCE: Data compare error.
	08	HDE: Header check.
	04	CRC: Cyclical redundancy check.
	02	O: Memory overrun.
	01	ISP: Short sector.
4-5	Residual Byte Count:	
	When there is an error completion, these two bytes contain the number of bytes of data that remain to be transferred.	
6	Retry Information:	
	First digit describes the retry set-up and second digit (n) shows the error count.	
	0n	Normal.
	1n	Data strobe early.
	2n	Data strobe late.
	3n	Offset minus.
	4n	Offset plus.
	5n	Data strobe early and offset plus.
	6n	Data strobe late and offset plus.
	7n	Data strobe early and offset minus.
	8n	Data strobe late and offset minus.
	9n	ECC used.

TAPE IOSW FORMAT

BYTE	0	1	2	3	4	5	6	7
	GS	ES	DDS		Byte Count		Device-Depndnt	

Byte	Hex Value	Description
0	General Status:	
	80	IRQ: Intervention required.
	40	NC: Normal completion.
	20	EC: Error Completion. When both NC and EC are set, a soft error occured. When only EC is set, the error was a hard error.
	10	U: Unsolicited interrupt.
	08	PC: IOP/DLP now ready.
	04, 02, 01	Reserved for future use by the operating system.
1	Error Status:	
	80	IC: Invalid command.
	40	MPE: Memory parity error.
	20	MAE: Memory address error.
	10	DM: Device malfunction.
	08	DAM: Device memory error, or device damaged.
	04	IL: Incorrect length.
	02, 01	Reserved for future use by the operating system.
2-3	Device-Dependent Status Information:	
2	80	VRC: Vertical redundancy check.
	40	LRC: Longitudinal redundancy check.
	20	CRC: Cyclical redundancy check.
	10	On write only: SC: Skew check. On read only: NB: Noise block.
	08	IMODE: Incorrect mode.
	04	PHASE: PE ID burst indicator detected.

Tape IOSW formats, continued

Byte	Hex Value	Description
2-3	Device-Dependent Status Information continued	
2	02	Kennedy: DS1: Drive type (0 is for 9-track drives and 1 is for 7-track drives). Telex: 0.
	01	Kennedy 7-Track only: DS2: Even Parity. All others: 0.
1	80	FP: File is write-protected.
	40*	Kennedy: DS3: Recording mode (0 is for PE and 1 is for NRZI). Telex: see below.
	20*	Kennedy: DS4: Density (0 is for high and 1 is for low). Telex: see below.
	10	LP: Load point.
	08	TM: Tape mark detected.
	04	EOT: End of tape detected.
	02	O: FIFO overflow or underflow.
	01	OFF: Offline.
	*Telex: bits	40 and 20
		00 PE.
		01 NRZI.
		10 GCR.
		11 Drive selected mode.
4-5	Residual Byte Count: When there is an error completion, these two bytes contain the number of bytes of data that remain to be transferred.	
6	Error Retry Count: Number of times IOP attempted unsuccessfully to complete the I/O operation normally.	
7	Reserved for future use of the operating system.	

SERIAL IOSW FORMAT

BYTE	0	1	2	3	4	5	6	7
	GS	ES	DDS		Byte Count		Device-Depndnt	

Byte	Hex Value	Description
0	General Status:	
	80	IRQ: Intervention required.
	40	NC: Normal completion.
	20	EC: Error Completion.
	10	U: Unsolicited interrupt.
	08	PC: IOP/DLP now ready.
	04	DAR: Data area early released. Means that the page in main memory is now released for use by other programs.
	02, 01	Reserved for later use by the operating system.
1	Error Status:	
	80	IC: Invalid command.
	40	MPE: Memory parity error.
	20	MAE: Memory address error.
	10	DM: Device malfunction.
	08	DAM: Device memory error, or device damaged.
	04	IL: Incorrect length.
	02	PP: Peripheral processor (Device-Level Processor) needs microcode.
	01	DP: Device Processor needs microcode.
2-3	Device-Dependent Status Information:	
2	All values	Used by telecommunications controllers as the AID character.

Serial IOSW Formats, continued

Byte	Hex Value	Description
3	<u>Device Information</u>:	
	x0	Value of first digit represents device type.
	Note that the second digit is zero for telecommunications controllers.	
	0x	For non-TC controllers, the following apply:
	01	DNR: Device Processor not running or halted. IPL requested.
	02	DCE: Device cable error.
	04	DPE: Device parity error.
	08	DPO: Device is powered-on.
4-5	<u>Residual Byte Count:</u>	
	When there is an error completion, these two bytes contain the number of bytes of data that remain to be transferred.	
	In 22V27 SIOPs operating in VS 100 machines, if byte 4 is 00, byte 5 can contain any of the following error codes:	
	00	Invalid BA command.
	01	BCC error.
	02	Memory address error.
	03	IOP/BA parity error.
6-7	<u>Device-Dependent: Extended Error Codes</u>	
	Used with telecommunications controllers. For serial controllers, these two bytes pass unaltered between the IOP and main memory.	

I/O INSTRUCTIONS

Whenever a program requires I/O, it calls on the operating system using the XIO instruction. It is the operating system which then actually orders the IOP to perform the I/O. The three instructions the operating system uses are privileged, i.e., only the operating system may use these three instructions. This means that the CP must be in supervisor state.

- o Start I/O (SIO) is the privileged instruction which institutes I/O operation.
- o Control I/O (CIO) is the privileged instruction which performs different types of operations with the device's intelligence.
- o Halt I/O (HIO) is the privileged instruction which immediately terminates an I/O operation.

All instructions use the RR addressing capabilities.

START I/O

SIO R1

02	R1	... ...	
0	8	12 15	BIT

The operating system uses the SIO command to initiate an I/O operation for the I/O device whose address is in R1. The instruction START I/O is executed only when the system is in the supervisor state.

The register specified by R1 contains the I/O device address in the following format.

....	DEVICE ADDRESS	
0 23	24 31	BIT

Before the SIO instruction is issued, the operating system stores the IOCW in memory then loads the pointer to the IOCW at the correct main memory address for the device in question.

After the SIO instruction is issued, the I/O operation will take place provided the addressed device and the IOP are available. Under any of the following conditions, the I/O operation will not take place:

o The IOP is not connected or operational.

o The IOP is busy.

o The device is busy with a previous IOCW.

After the SIO instruction is executed, the operating system cannot change the IOCW until the completion interrupt is received from the IOP.

When the IOP receives the IOCW, it returns a condition code. The various condition codes which can result from this operation are:

0 SIO instruction is accepted by IOP and operation is proceeding.
1 The device is busy with a previous operation.
2 IOP BUSY is true.
3 The IOP is not operational.

CONTROL I/O

CIO R1

OC	R1	...	
0 7	8	12 15	BIT

CIO commands the IOP to perform its control function. In most cases, this means that the IOP loads microcode for the Device-Level Processor.

The register specified by R1 contains the I/O device address in the following format.

...	DEVICE ADDRESS	
0 23	24 31	BIT

Before the CIO instruction is issued, the operating system stores the IOCW in memory then loads the pointer to the IOCW at the correct main memory address for the device in question. After the CIO instruction is issued, the I/O operation will take place provided the addressed device and the IOP are available. Under any of the following conditions, the I/O operation will not take place:

- The IOP is not connected or operational.
- The IOP is busy.
- The device is busy with a previous IOCW.

Not all devices support CIO. When issued for a device for which control is not accepted, the 0 condition code is returned and program execution continues.

The various condition codes which can result from this operation are:

0 Means one of two things:
   - CIO instruction is accepted by IOP and operation is proceeding.
   - This device cannot accept a CIO instruction.

1 The device is busy with a previous operation.
2 IOP BUSY is true.
3 The IOP is not operational.

HALT I/O

HIO R1

03	R1	...	
0 7	8	12 15	BIT

HIO commands the IOP to immediately terminate activities in the addressed device.

The register specified by R1 contains the I/O device address in the following format.

....	DEVICE ADDRESS	
0 23	24 31	BIT

Not all devices support HIO. When issued for a device for which control is not accepted, the 0 condition code is returned and program execution continues.

HIO clears all interrupts that may be pending for the device. In addition, if any I/O operations are currently going on, HIO terminates them as well. If a current operation is terminated, a completion interrupt will be generated. The IOSW that the IOP places in main memory may or may not have the Error Completion and Incorrect Length bits set to 1.

The various condition codes which can result from this operation are:

0 Means one of two things:

- HIO instruction is accepted by IOP and operation is proceeding.
- This device cannot support the HIO instruction.

1 The device is busy with a previous operation.

2 IOP BUSY is true.

3 The IOP is not operational.

CHAPTER 3
IOPMB BLOCK DIAGRAM

INTRODUCTION

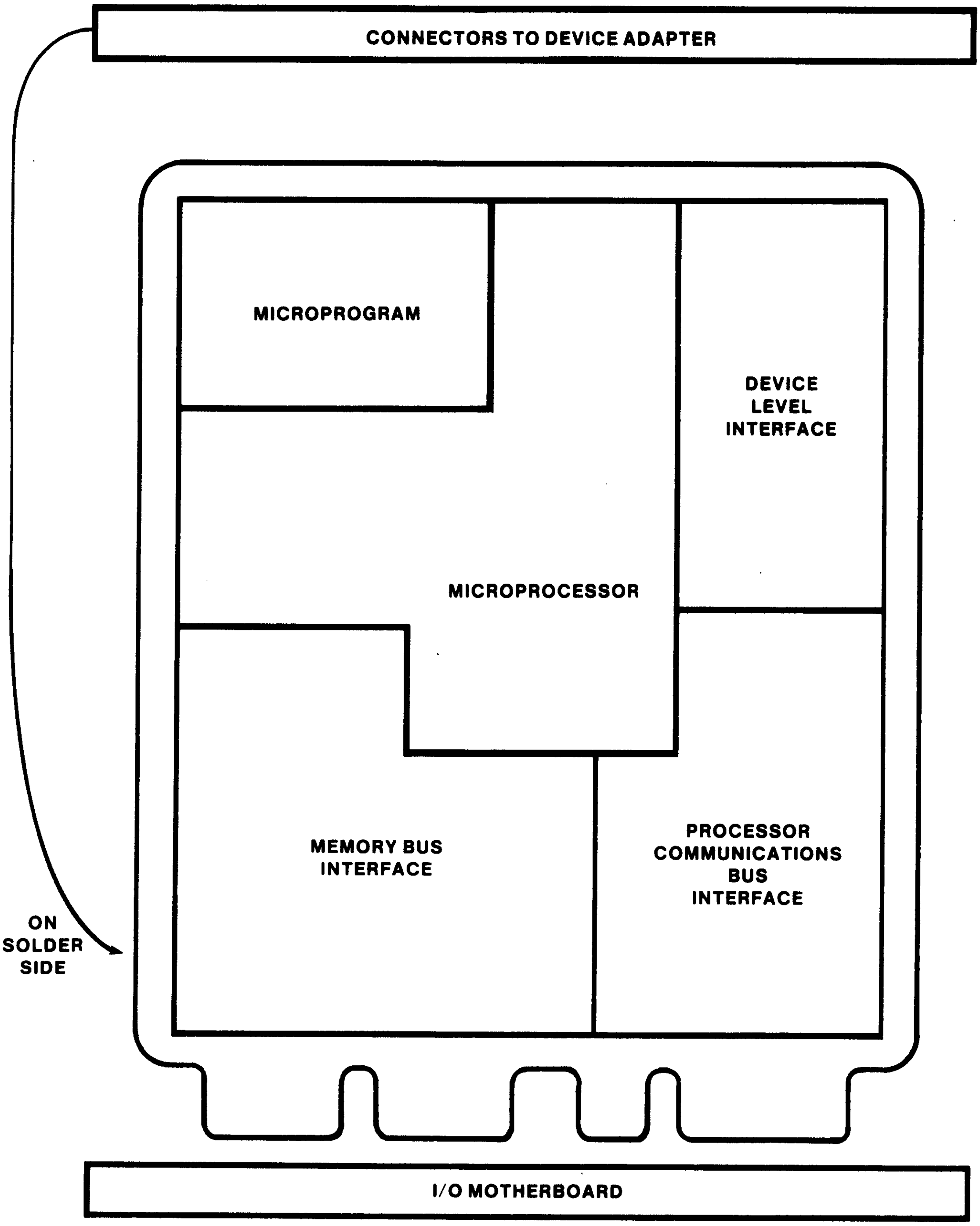
This chapter provides a block diagram of the IOPMB portion of an IOP assembly. The block diagram presents the logic at the functional level. Block diagrams for the DA portion are covered in the pertinent Product Maintenance Manuals.

Overall, there are five major functional areas on the IOP Motherboard.

- o Intelligence:
  - Microprogram
  - Microprocessor
- o Interfaces:
  - Memory Bus Interface
  - Processor Communications Bus Interface
  - Device Level Interface

Figure 3-1 presents a simple view of a IOP Motherboard. Each of these five functional blocks are discussed separately below.

Figure 3-1
Simple Block Diagram of IOPMB



B-01813-FY85-1

MICROPROGRAM

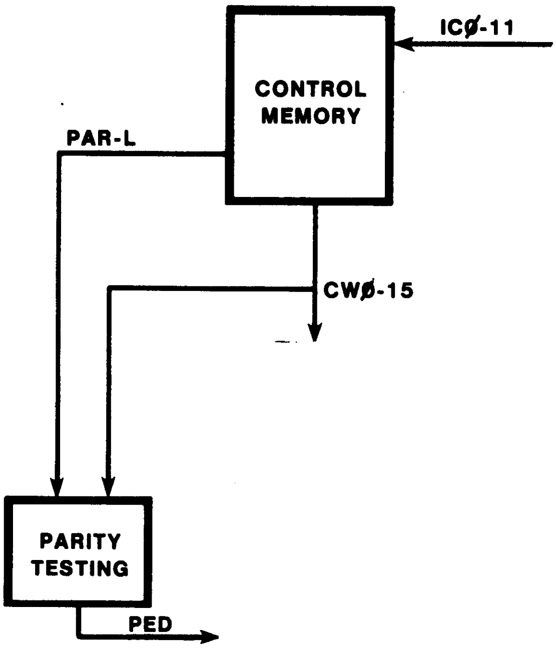
The Microprogram section of the IOPMB is diagrammed in detail in Figure 3-2. This function of the IOPMB consists of the Control Memory which stores the IOPMB microcode and of the Parity Testing Circuit which checks this microcode for errors.

The Control Memory consists of PROMs. The PROMs vary from one IOP to the next. Each set of PROMs contains the particular microcode program which drives the IOPMB for use in a specific environment. One set of PROMs makes the IOPMB run large disks; another set makes the IOPMB run workstations and printers.

Control Memory has 4 kilowords of 16 bits. It is the Microprocessor section which addresses the PROMs and latches the control word output.

The Parity Testing Circuit includes both a parity storage PROM and a parity generation chip. As each control word is read into the Microprocessor section, the parity generator creates the correct parity bit for the word. This bit is then checked against the stored parity bit from the PROM. If the two bits do not match, the Parity Testing Circuit generates an error signal called PED for Parity Error Detect.

Figure 3-2
Microprogram Detailed Block Diagram



B-01813-FY85-7

MICROPROCESSOR

The Microprocessor function of the IOPMB is shown in the detailed block diagram of Figure 3-3. This section consists of the ALU, its input multiplexor and registers, the Control Word Latch, the Instruction Counter, a Subroutine Address Stack and Operation Decode circuitry.

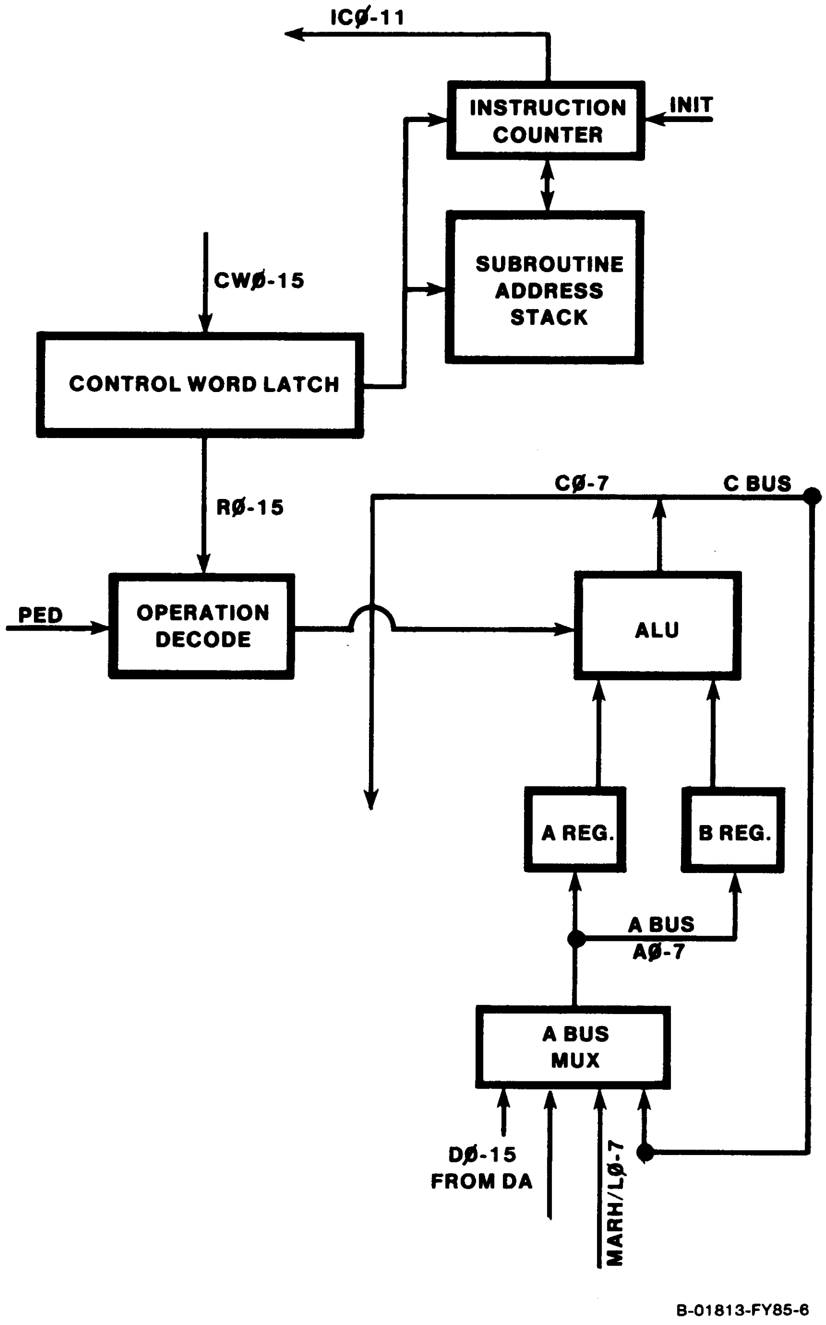
The ALU is 8 bits wide and its multiplexor and registers are also 8 bits wide. The input bus to the ALU is the A bus. The output bus from the ALU is the C bus. Both eight-bit buses are used as board-wide communication paths. Most on-board communication happens via the ALU. Data from one storage area is gated through the ALU to the C bus and on to its destination buffer.

The ALU input multiplexor is called the A Bus Mux. One of its major functions is to select which parts of the wider buses will appear on the eight-bit A bus. Its inputs come separately from both the high and low sides of the latches in the Memory Bus Interface and the Device Level Interface.

The Microprocessor stores the current control word instruction in the Control Word Latch. The various bits of this instruction are decoded by the operation decode logic. Outputs of the Operation Decode logic select the ALU's functions and gate with many other signals on the IOPMB board.

To address the instructions in Control Memory, the Microprocessor includes an Instruction Counter. The Instruction Counter can be loaded from the previous control word or from a Subroutine Address Stack. The Subroutine Address Stack stores up to sixteen 12-bit Control Memory addresses. When necessary, the Microprocessor can force a low address into the Instruction Counter so that it can handle an emergency interrupt.

Figure 3-3
Microprocessor Detailed Block Diagram



MEMORY BUS INTERFACE

Figure 3-4 presents a detailed block diagram of the Memory Bus Interface of the IOPMB. This interface consists of the buffers for the memory data and address buses as well as memory address generation and memory control signals.

The Memory Data Buffer consists of tristate drivers which can send out data, strobe data in or stay at a high impedance so that other IOPs can drive signals on the bus unhindered. There are two parts to the Memory Data Buffer: one-half stores the low-order byte of data and one-half stores the high-order byte of data. From this buffer, data is bused to all parts of the IOPMB.

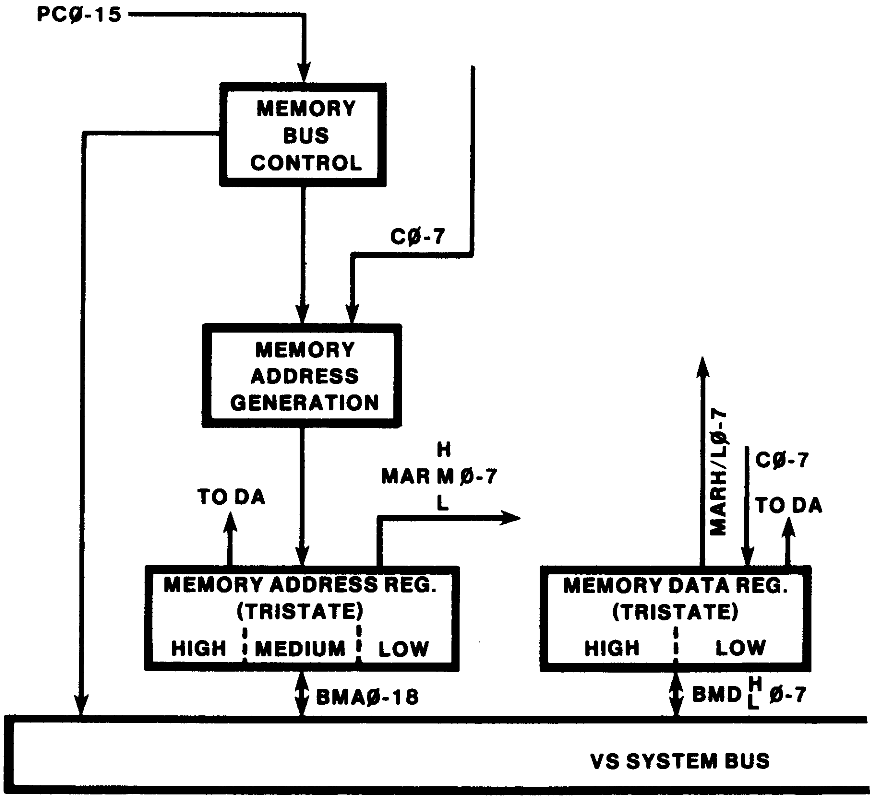
One important thing to note about the IOPMB's Memory Bus Interface is that it does not include any large-scale buffering for data en route from the device to main memory. As soon as the IOPMB receives a word of data from the device, it must get access to the memory bus in order to store that data in main memory. The amount of time which the IOPMB can take to clear the Memory Data Buffer before the next word comes from the Device Adapter logic depends on the type of device and on any buffering which may be present in the DA itself.

The Memory Address Buffer consists of tristate drivers which can send addresses out to the bus, strobe addresses in from the bus or stay at a high impedance so that other IOPs can drive signals on the bus unhindered. There are three parts to the Memory Address Buffer: one stores the low-order byte of the address, one stores the middle byte of addresses and the last stores the highest-order three bits.

In all, the IOP handles 19-bit addresses for a 512-kilobyte address space. This is okay for VS 80 machines. In VS 85, 90 or 100 machines, however, there is a maximum of 16 megabytes of memory, all of which is handled by a 24-bit address. In these machines, the BA handles the change from 19-bit to 24-bit addresses.

The Memory Bus Interface includes Memory Address Generation logic. This logic permits the IOP to sequence through all the memory addresses involved in block transfers between main memory and the devices it services. The Memory Address Generation logic can operate incrementally or it can be loaded directly from the ALU.

Figure 3-4
Memory Bus Interface Detailed Block Diagram



B-01813-FY85-5

Table 3-1
Main Memory Bus Interface Signals

Signal Name	From	To	Function
BMA0-18	IOPMB	Memory	Carry 19-bit main memory physical address.
MM0-15	Memory	IOP	Carry data that has just been read from memory.
BMDH0-7	IOP	Memory	Carry the high-order byte of data.
BMDL0-7	IOP	Memory	Carry the low-order byte of data.
MRI	IOP	Memory	Requests use of the memory bus.
MGS	Memory	IOP	Grants use of the memory bus.
MMCB1-2	IOP	Memory	Indicates type of transfer: 00 No transfer. 01 Read 16 bits. 10 Write 16 bits. 11 Write 8 bits.
MCOS	Memory	IOP	Indicates when valid data is on the bus so that the IOP can strobe it into the memory data buffers.
MMP	Memory	IOP	Indicates that a parity error occured when main memory was reading the data.
IMA	Memory	IOP	Indicates that the address supplied to memory by the IOP was illegal.

PROCESSOR COMMUNICATIONS BUS INTERFACE

Figure 3-5 shows the Processor Communications Bus Interface. This interface consists of Device Selection Logic and PCB decode & control logic.

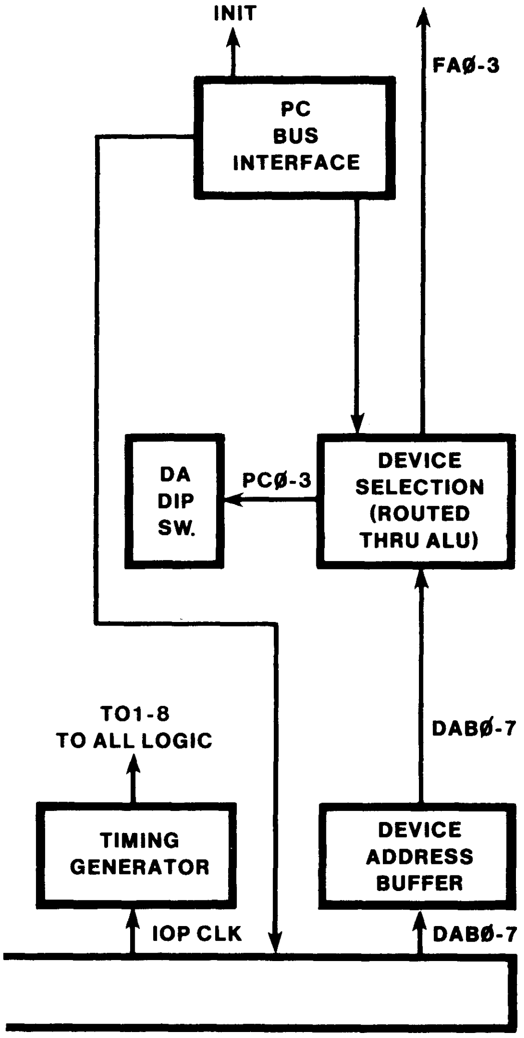
The Device Selection Logic determines if the device address on the system bus is the same as the device address range set into the dip switch. It consists of a Device Address Buffer, a dip switch and some routing logic. The system bus uses lines DAB0-7 to send the device address out to the IOPs. The IOPMB stores this address in the Device Address Buffer. When the CP sends out the device address, the IOPMB Microprogram responds by comparing the Device Address Buffer to the device address dip switch settings. When the two match, the SEL SIXTN signal is generated.

The PCB decode & control logic looks at the discrete signals on the Processor Communications Bus. It decodes incoming signals to determine if any I/O functions are occuring. It also provides outgoing signal responses when required. It acts in conjunction with the Microprogram which determines the operating status of the IOPMB. Table 3-2 shows the various discrete PCB control signals and their functions.

TIMING GENERATOR

The IOP uses a crystal oscillator to produce timing cycles. A single bit is cascaded through the timing generator. This bit becomes, in turn, T01, T02, ..., T08. Under two types of conditions this timing sequence is extended. To access on-board Control Memory, the IOP adds T09 and a T010. To access Main Memory, the IOP extends the duration of T02.

Figure 3-5
Processor Communications Bus Detailed Block Diagram



B-01813-FY85-4

Table 3-2
PCB Control Signals and Functions

Signal Name	From	To	Function
CCB1-2			Control Command Bits are used at two times.
	CP	IOP	o The CP places the command type on these lines: 0 Control Mode Alert 1 Start I/O 2 Control I/O 3 Halt I/O
	IOP	CP	o The IOP places its immediate response code on these lines: 0 Available 1 Device Busy 2 IOP Busy 3 Device Not Operable
DAB0-7			Device Address Buffered is used at two times.
	CP	IOP	o When the CP is sending a command to the IOPs, it places the address of the device under question on these lines.
	IOP	CP	o After the CP grants the Processor Communications Bus to an IOP, the IOP indicates the responding device by placing the device address on these lines.

Table 3-2 (continued) PCBI Signals

Signal Name	From	To	Function
PCBCS	CP	IOPMB	PCB Control Strobe In is used by the CP to indicate when PCB bus data is valid. It is used as a strobe signal on the Device Address Buffer.
PCBSI	IOPMB	CP	PCB Strobe In is used by the IOP to indicate to the CP when PCB bus data is valid. It is used as a strobe signal in the CP.
PCBGS	CP	IOPMB	PCB Grant Strobe is used by the CP as a signal to the IOPMB that it can use the Processor Communications Bus at this time. It is issued in response to a PCBRI.
PCBRI	IOPMB	CP	PCB Request In is activated by the IOPMB whenever it wants to use the Processor Communications Bus.

DEVICE LEVEL INTERFACE

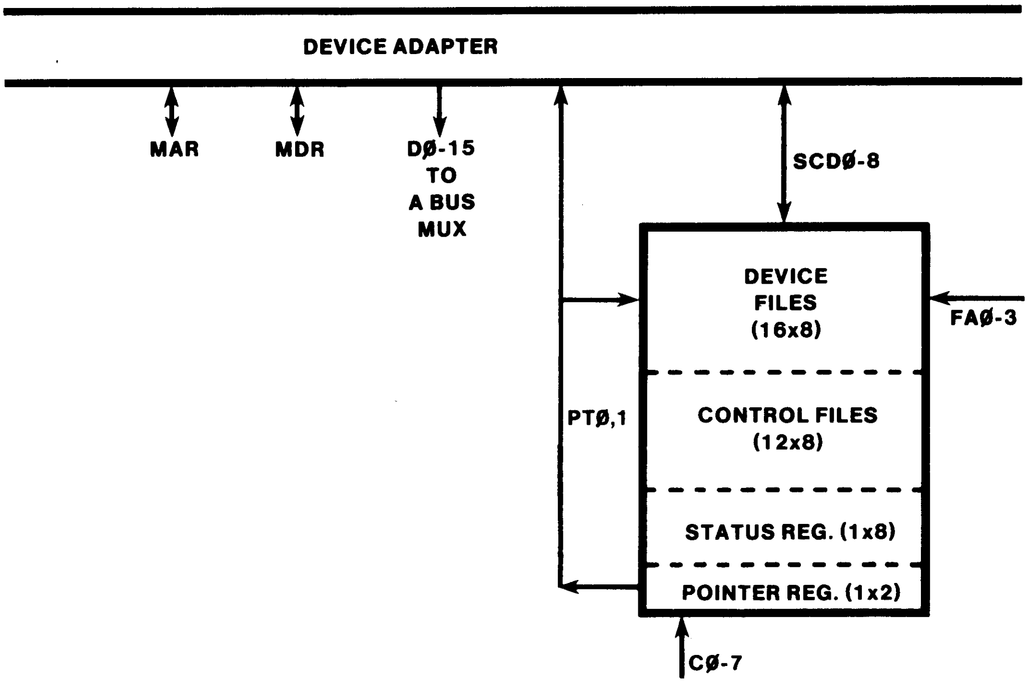
It is the Device Level Interface which speaks directly to the Device Adapter. This communication takes place over four connectors mounted on the solder side of the IOPMB board. The interface consists of RAM storage for Device Files, Control Files, a Status Register and a Pointer Register. Figure 3-6 is a detailed block diagram of the Device Level Interface.

The RAM storage consists of 32 eight-bit words. The first sixteen words are devoted to the Device Files. The Microprogram arranges these into four stacks. Each stack has four files for its DA port. The next twelve words are used for the Control Files. The Microprogram uses the Control Files for the register-to-register activities of the ALU. The last four words are special-purpose. One word is used as a Status Register. Two bits of another word are used as the Pointer Register.

The Pointer Register is two bits long. It addresses the four ports of the Device Adapter or the four 4-byte stacks of the Device Files.

Many of the signals to the Device Adapter are sourced by the various buffers of the Memory Bus Interface and the Processor Communications Bus Interface. They are indicated on the block diagram. The IOPMB provides enough signals to the DA so that it can, if it wishes, control the Main Memory Bus Interface. This allows large disk adapters to access the DMA facility directly.

Figure 3-6
Device Level Block Diagram



B-01813-FY85-3

Table 3-3
Device Level Interface Signals

Signal Name	From	To	Function
RO-15	IOPMB	DLI	Contains the value of the Control Word Latch. Allows the DA to monitor the microinstructions.
CDLI	IOPMB	DLI	When the instruction in the Control Word Latch is one to control the DLI, this line is asserted.
CM DISABLE	DLI	IOPMB	Used by large disk controllers to turn off the IOP's intelligence so that the DA can control the MMBI.
PARH,PARL	DLI	IOPMB	Allows memory on the DA to pull the correct lines on the IOPMB parity circuitry whenever a parity error occurs.
INIT2	IOPMB	DA	When a parity error occurs in the IOP's Control Memory, it uses this signal to initialize the DA. The parity error represents an unreliable IOP and this signal is used to halt all devices (without zeroing any registers) while the IOP waits for the CP to respond and read those registers.
ICO-11	IOPMB	DA	Contains the value in the Instruction Counter. Allows access to external alternate memory on the DA.
BXTAL	IOPMB	DA	150 nanosecond clock supplied by IOP.
IOP CLK	DA	IOPMB	Used by DAs which want to supply their own clock.
TO1,2,4,6,7	IOPMB	DA	Supplied by IOPMB so that DA can synchronize to certain events of the IOP instruction cycle.

Table 3-3 continued: DLI Signals

Signal Name	From	To	Function
STOP,START	DA	IOPMB	Available for test equipment in the Repair Centers.
RIPPLE CLK	DA	IOPMB	Allows the DA to cause the memory address generator to increment during direct memory access transfers.
SINPT	DA	IOPMB	Used by DA to set to 1 status register bit 1. Indicates that the DA has completed the operation requested by the previous CDLI instruction.
STO-PS	DA	IOPMB	Used by DA to set status register bit 0.
STO-Q	IOPMB	DA	Shows the DA the current value of status register bit 0.
ST4-PS	DA	IOPMB	Used by DA to set status register bit 4.
ST4-Q	IOPMB	DA	Shows the DA the current value of status register bit 4.
DO-15	DA	IOPMB	Carries the 8-bit or 16-bit data value to be strobed into the IOP's memory data buffer.
DSMH	DA	IOPMB	Used by DA to strobe the high-order eight bits of DO-15 into the memory data buffer.
DSML	DA	IOPMB	Used by DA to strobe the low-order eight bits of DO-15 into the memory data buffer.
MDHO-7	IOPMB	DA	Carries the current value of the MDRH register.
MDLO-7	IOPMB	DA	Carries the current value of the MDRL register.

Table 3-3 continued: DLI Signals

Signal Name	From	To	Function
MA0-18	DA	IOPMB	Used by the DA to supply a nineteen-bit address for main memory.
RIPPLE MAR	DA	IOPMB	Allows the DA to cause the memory address generator to increment during direct memory access transfers.
INC2	DA	IOPMB	Allows the DA to increment the memory address generator two steps.
TP4-3	DA	IOPMB	Used by test equipment in the Repair Center for control of the memory address generator.
HW16	DA	IOPMB	Used by DA to specify to MMBI that DMA transfer is a 16-bit write.
HR16	DA	IOPMB	Used by DA to specify to MMBI that DMA transferis a 16-bit read.
HTS	DA	IOPMN	Used by DA to halt a DMA transfer.
HSB1	IOPMB	DA	Used by IOPMB to tell DMA device that a Main Memory parity error occured.
HSB2	IOPMB	DA	Used by IOPMB to tell DMA device that it tried to access an illegal Main Memory address.
BMOF	IOPMB	DA	Informs DA that the last single cycle transfer finished.
SEL SIXTN	DA	IOPMB	Used in VS 80 systems to tell IOPMB that it should be checking only the highest-order four bits of the device address buffer.
PT0-1	IOPMB	DA	The value of the Pointer Register. Used when the DA has Device File registers in addition to the sixteen on the IOPMB. Points to one of a stack of four areas.

Table 3-3 continued: DLI Signals

Signal Name	From	To	Function
FA0-3	IOPMB	DA	The value of the file address within the Device File stack. Extends the addressing of PT0-1.
TP39	IOPMB	DA	Provides write strobe for extended Device File stacks.
TP40	IOPMB	DA	Monitors the write enable strobe for the Control File registers on the IOPMB.
SW PANEL INIT	IOPMB	DA	Used for test equipment in Repair Centers.
SYSTEM INIT	IOPMB	DA	Provides initialization for DA during a system initialization or an IOP Control Memory parity error.

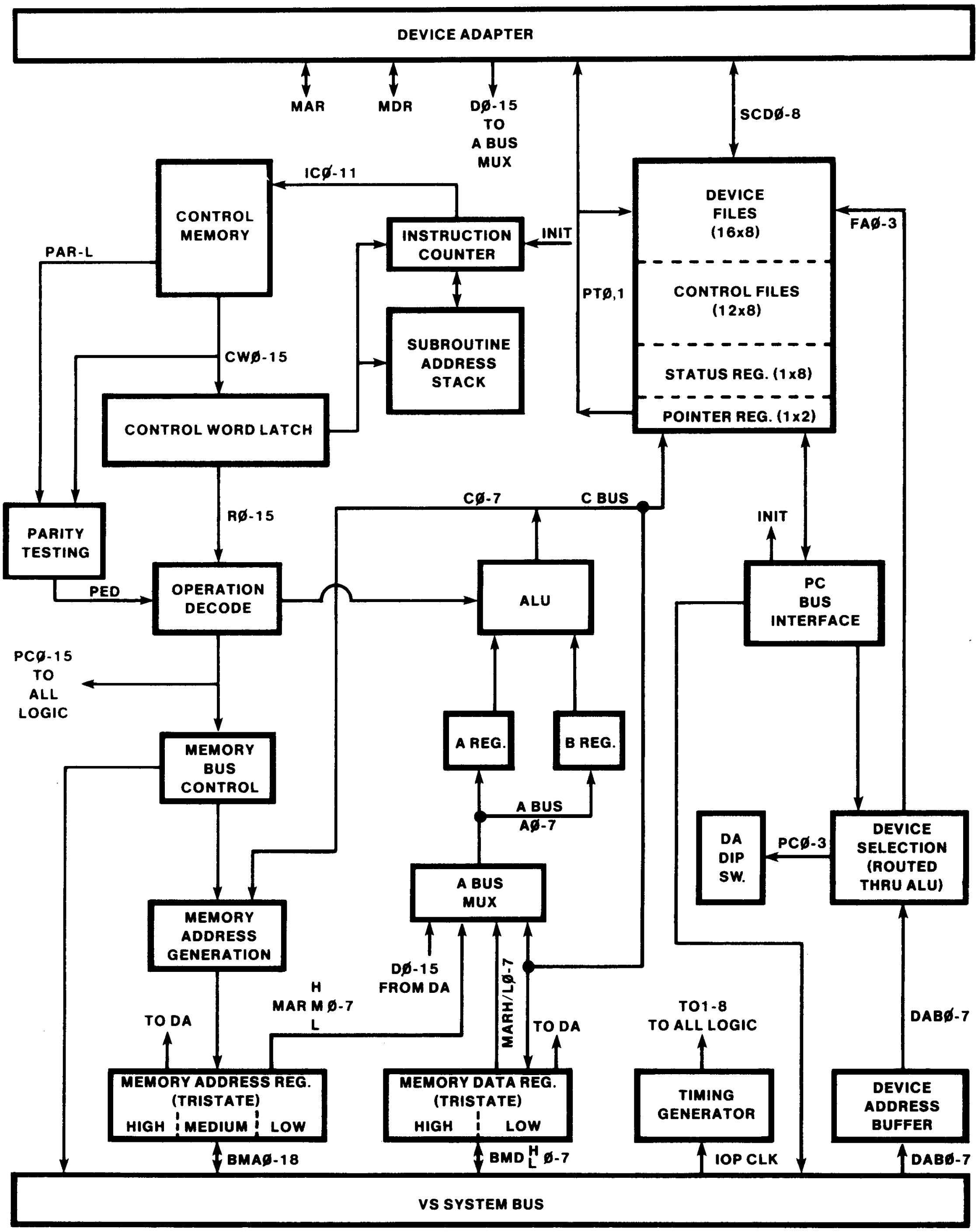
SUMMARY

This chapter has developed piece-by-piece a detailed block diagram of the IOP Motherboard. The logic which has been discussed is general-purpose circuitry whose specific functionality is determined by the Microprogram PROM set.

Figure 3-7 is a complete detailed block diagram for the IOPMB.

The only real difference between a 209-7810 and a 209-7110 IOP Motherboard is in how the microcode handles the device address. On one board, an additional bit is used to extend the address space from 16 to 32. The number of Device Files, however, remains the same.

Figure 3-7
IOPMB Detailed Block Diagram



B-01813-FY85-2

CHAPTER 4
INTERFACE CONTROL

INTRODUCTION

This chapter discusses two things: the sequence in which I/O steps occur and the particular ways in which the PCBI, the MMBI and the DLI perform these steps. The command structure was explained in chapter 2 and the capabilities of the IOPMB hardware were explained in chapter 3. This chapter will explain how the IOPMB hardware interacts with the operating system software.

Remember that the IOP in a VS 80 is placed directly on the system bus Motherboard and speaks directly to the CP. In a VS 85, 90 or 100, the same IOP is placed not on the system bus Motherboard but rather on an I/O Motherboard governed by a Bus Adapter (BA). The Product Maintenance Manuals on the advanced VS systems explain how the BA buffers IOP communication with the CP. This difference will not be discussed in this manual.

SEQUENCE OF I/O OPERATION

In general, three things happen during the execution of an I/O instruction: the CP passes the command to the IOP, the IOP performs the operation, and the IOP passes the outcome status back to the CP. In practice, this amounts to several stages of handshaking and information-passing. Figure 4-1 is a flowchart outlining these steps.

CP Places IOCW in Main Memory: The operating system creates the command word and places this word in main memory. This step involves the CP gaining access to the main memory bus. The hardware involved in the main memory write is described in the appropriate Product Maintenance Manual.

CP Issues SIO Command to IOP: The second step is to let the IOP know that a command is waiting for a particular device. The CP issues this command on the system bus and the IOP issues an immediate response on the same system bus lines. These activities take place over the PCBI.

IOP Reads IOCW from Main Memory: The IOP must then read the command from main memory. It must know exactly where to find the command in memory and it must contend for use of the main memory bus. Once it has been granted the bus, it must adjust its timing to that of main memory. These activities take place over the MMBI.

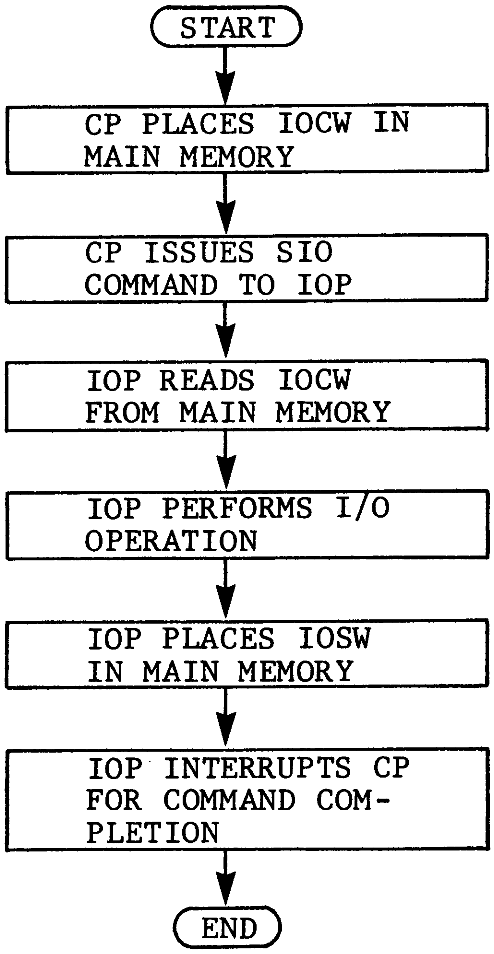
IOP Performs I/O Operation: The IOP must then use the device-level interface to control the device and to perform the operation requested. The Microprogram uses the lines to the DA to exert device control. These hardware involved in this transaction are the DLI and the DA.

IOP Places IOSW in Main Memory: After the operation is complete, the IOP creates an outcome status word in the prescribed format and places that word in main memory. Once again the IOP must contend for the main memory bus. These activities take place through the MMBI.

IOP Interrupts CP for Command Completion: The IOP lets the CP know that an I/O operation is complete and that the outcome status word is waiting in main memory. It does this by issuing an I/O interrupt on the system bus. The CP responds to the interrupt by checking to ensure that the operation was error-free and by taking the program off hold status. These activities take place through the PCBI.

Figure 4-1 is a flowchart of the major steps involved in the entire performance of an I/O operation.

Figure 4-1
I/O Operation


START
CP PLACES IOCW IN MAIN MEMORY
CP ISSUES SIO COMMAND TO IOP
IOP READS IOCW FROM MAIN MEMORY
IOP PERFORMS I/O OPERATION
IOP PLACES IOSW IN MAIN MEMORY
IOP INTERRUPTS CP FOR COMMAND COMPLETION
END

HOW THE PCBI RESPONDS TO THE SIO

When an SIO command is received by an IOP, the IOP response facility comes into operation. This facility must determine the following conditions:

- o Is this command for a device serviced by this IOP?
- o Is the IOP itself free to respond to the command?
- o If the IOP is free, is the device also free to respond to the command?

The conditions are determined through the ALU, the Control Files and the Status Register.

While the SIO command is fed into the IOP on the CCB1-2, the device address is fed into the IOP on DAB0-7. A test for matching device addresses is performed by the IOP's Microprogram and Microprocessor facilities. If a match occurs, the IOP knows that it must respond to the CCB lines and take on the I/O operation.

The result of this determination of operating condition and device address is a two-signal response to the CP. This two-signal response indicates whether the IOP is able to take on the I/O operation at this time.

To summarize, the lines involved in responding to the SIO are:

CCB1-2	The control lines from the CP and, at the same time, the response lines from the IOP.
DAB0-7	The device address.

HOW THE MMBI CONTENDS FOR BUS ACCESS

The IOP contends for the main memory bus with other IOPs and with the CP. To indicate to the memory that it wishes bus access, the IOP raises the signal MRI. When the IOP's turn to access the bus comes up, memory lets it know by asserting the signal MGS.

HOW THE MMBI CHANGES ITS TIMING TO MEET BUS SPECS

Once the IOP has access to the memory bus, it must control that bus. It must place the appropriate memory command on the memory bus; it must sync into the memory bus timing, and it must drive and strobe the address and data buses.

To indicate to the memory which function is being performed, the IOP asserts MMCB1-2. These lines indicate whether the function is a read or a write; they can also indicate whether 1 byte or 1 word are being transferred.

To change its timing to agree with that of the main memory bus, the IOP holds its T07 timing state for an extra 750 nanoseconds.

The IOP places write data going to main memory on BMDH0-7 and BMDL0-7. It indicates when this data is valid by asserting MCOS. Read data coming from main memory to the IOP is passed on lines MM0-15.

HOW THE DLI TALKS TO THE DEVICE

The IOP controls the device through the Device Level Interface. This interface consists of sixteen registers to keep track of device data and 120 signals. The 120 signals are to give the Device Adapter access to the resources of the IOPMB. This access is general-purpose and is used by each Device Adapter designer as required for the particular devices. See the pertinent Product Maintenance Manuals for descriptions of particular DAs.

The IOPMB does have one instruction which gives the Microprogram a twelve-bit command field for direct communication with the DA. When this command is executed, the control signal CDLI is asserted and the 16 bits of the Control Word are passed on to the DA. Twelve of these 16 bits can be used to implement a command structure within the DA.

CONTROL DEVICE LEVEL INTERFACE

CDLI

BIT 0 15

0 1 0 1	0 C C C	C C C C	C C C C

where the C field is the control operand. The actions taken by the Device Adapter on receipt of this command depend on the DA under consideration.

HOW THE PCBI INTERRUPTS THE CP

The IOP signals the CP that an IOSW is waiting by issuing an I/O interrupt. I/O interrupts are at the lowest interrupt priority. Interrupts which have a higher priority are machine check interrupts, program check interrupts and clock interrupts.

The CP services interrupts only between instructions. Two CP instructions (CLCL and MVCL) are exceptions to this rule. Both of these "Long" instructions can be interrupted.

The CP services I/O interrupts only after interrupts of higher priority have been serviced. For example, if a disk IOP and the clock both request interrupts at the same time, the clock will be serviced first and the disk IOP will be serviced second. The pertinent Product Maintenance Manuals for the different VS machines explain which priorities they implement.

Machine checks can interrupt anything. Machine checks occur if there is a parity error in main memory or if an IOP either times out or violates protocol. Both of these types of errors indicate a system of questionable reliability and thus take the highest priority.

The operating system has the ability to mask interrupts. This means that, if the I/O interrupt bit is masked, any interrupts occuring will be ignored by the operating system.

APPENDIX A
IOP INSTRUCTION SET

INTRODUCTION

This is the instruction set in which the Microprogram is written. This instruction set operates the 74LS181 ALU chips used in the Microprocessor. The Microprogram formed of these instructions is stored in 2708 PROMs on the IOPMB.

Instructions are broken down into

- o logical and arithmetic instructions,
- o conditional and unconditional branch operations, and
- o interface control operations.

In addition to the instruction formats, this appendix describes the various allowed fields for each instruction.

STATUS REGISTER

BIT	0	1	2	3	4	5	6	7
	U1	IN	SV	RB	U2	C2	C1	CA

Bit	Mnemonic	Meaning
0	U1	Undefined.
1	IN	Input: Indicates when a CDLI instruction is waiting to be serviced by the DLI. Issuing a CDLI instruction sets this bit to 1. When the DLI acts on the instruction, it clears the bit to 0.
2	SV	Service:
3	RB	Not Ready/Busy:
4	U2	Undefined.
5	C2	Compare Bit 1: Set to 1 by C instruction when A is less than B. Cleared to 0 when A is greater than or equal to B.
6	C1	Compare Bit 2: Set to 1 by C instruction when operands are not equal. Cleared to 0 when A equals B.
7	CA	Carry Bit: Set to 1 by the AC instruction when there is a carry out of the addition.

SPECIFYING REGISTERS IN INSTRUCTIONS

The following values are allowed for A and B registers in the instruction set.

Field Value	A	B	Data Routed to Register
00000 to 01011	X	X	The Control Files CF0 to CFB.
01100		X	The Pointer Register (2 bits right-justified). Do not use this file in the A field.
01101	X	X	The Status Register.
01110	X	X	MDRH (the high-order eight bits of the Memory Data Register).
01111	X	X	MDRL (the low-order eight bits of the Memory Data Register).
10000 to 11111	X	X	The Device Files DF0 to DFF.

TRAP HANDLING

Trap Conditions:

Interface	Condition Setting Trap
a	MMBI Main Memory Parity Error.
b	" Invalid Main Memory Address.
c	PCBI Initialize.
d	" The request line and the grant line are active.
e	" CP has issued and SIO, CIO or HIO and the RB bit of the Status Register is clear, indicating that the IOP is ready.

Trap Addresses:

Address	Condition	
	0	Initialize, a above
	1	Parity Error, b above
	2	Addressing Error, c above
	4	d and e above

LOGICAL AND ARITHMETIC INSTRUCTIONS

LOGICAL AND

AND A, B

BIT 0 15

0 0 1 1	1 0 A A	A A A B	B B B B

The data specified by the A field is gated to the A register and the data specified by the B field is gated to the B register. The ALU then performs a logical AND, bit-by-bit, on the two values and stores the result in the B register.

LOGICAL OR

OR A, B

BIT 0 15

0 0 0 0	1 0 A A	A A A B	B B B B

The data specified by the A field is gated to the A register and the data specified by the B field is gated to the B register. The ALU performs a logical OR on the values, bit-by-bit, and stores the result in the B register.

LOGICAL XOR

XOR A, B

BIT 0 15

0 0 0 1	0 0 A A	A A A B	B B B B

The data specified by the A field is gated to the A register and the data specified by the B field is gated to the B register. The ALU then performs a logical XOR on the values, bit-by-bit, and stores the result in the B register.

ARITHMETIC ADD

A A, B

BIT 0 15

0 0 0 1	1 0 A A	A A A B	B B B B

The data specified by the A field is gated to the A register and the data specified by the B field is gated to the B register. The ALU then adds the two values together and stores the result in the B register.

ARITHMETIC ADD WITH CARRY

AC A, B

BIT 0 15

0 0 0 1	1 1 A A	A A A B	B B B B

The data specified by the A field is gated to the A register and the data specified by the B field is gated to the B register. The ALU adds the two values together and stores the result in the B register. The value of the carry out to the left is stored in the Status Register as CA, in bit 7.

MOVE

MV A, B

BIT 0 15

0 0 1 0	1 0 A A	A A A B	B B B B

The data specified by the A field is gated to the A register and the data specified by the B field is gated to the B register. The A register value is then copied over to the B register and the A register remains unchanged.

COMPARE

C A, B

BIT 0 15

0 0 1 1	0 1 A A	A A A B	B B B B

The data specified by the A field is gated to the A register and the data specified by the B field is gated to the B register. The ALU then compares the magnitudes of the two registers. The results are reflected in C1 (bit 6) and C2 (bit 5) of the Status Register, as follows:

C1 = 1 if the two are equal.
   = 0 if the two are not equal.
C2 = 1 if A is less than B.
   = 0 if A is greater than or equal to B.

LOAD MEMORY ADDRESS REGISTER

LM A, S

BIT 0 15

0 0 1 0	0 0 A A	A A A SL	SM SH - -

where SL refers to MARL, SM refers to MARM and SH refers to MARH.

This instruction loads the specified Memory Address Register(s) with the contents of A. If more than one bit in the S field is set to 1, the contents of A are loaded into each indicated register. When loading MARH, the instruction takes the two low-order bits of the A register only.

MOVE IMMEDIATE

MVI 'I', B

BIT 0 15

1 0 0 I	I I I I	I I I B	B B B B

The I value in the instruction is loaded into the register specified by the B field.

OR IMMEDIATE

ORI 'I', B

BIT 0 15

1 0 1 I	I I I I	I I I B	B B B B

The I value in the instruction is ORed to the contents of the B register and the results are stored in the B register.

BRANCH INSTRUCTIONS

SKIP IF TRUE ON MASK

ST 'M', B

BIT 0 15

1 1 1 M	M M M M	M M M B	B B B B

The B field contains the number of the register to be tested. The M field contains the mask. The ALU tests for a 1 value each bit in the B register which corresponds to a 1 bit in the mask.

Condition of B	Instruction Counter
All bits = 1	Increment by 2
Any bits = 0	Increment by 1

SKIP IF FALSE ON MASK

SF 'M', B

BIT 0 15

1 1 0 M	M M M M	M M M B	B B B B

The B field contains the number of the register to be tested. The M field contains the mask. The ALU tests for a 0 value each bit in the B register which corresponds to a 1 bit in the mask.

Condition of B	Instruction Counter
All bits = 0	Increment by 2
Any bits = 1	Increment by 1

BRANCH UNCONDITIONAL

B IC

BIT 0 15

0 1 1 0	IC IC IC IC	IC IC IC IC	IC IC IC IC

Bits 4 to 16 of the instruction are placed in the Instruction Counter and the Microprogram continues from the new address.

SUBROUTINE BRANCH

SB IC

BIT 0 15

0 1 1 1	IC IC IC IC	IC IC IC IC	IC IC IC IC

The current value of the Instruction Counter is incremented by 1 and pushed onto the stack. The contents of the IC field are placed into the Instruction Counter and the Microprogram continues from the new address.

SUBROUTINE RETURN

SR

BIT 0 15

0 1 0 1	1 - - -	- - - -	- - - -

The stored value of the Instruction Counter is popped from the stack and placed into the Instruction Counter. The Microprogram continues from this new value.

CONTROL MEMORY BUS INTERFACE

CMBI

BIT 0 15

0 1 0 0	0 - - -	- - - D	M M R R

The MM field indicates the type of memory transfer to be performed. The transfer will take place with the memory address taken from the current contents of the Memory Address Register.

MM	Type of Transfer
00	No operation.
01	Read 16 bits from memory into the Memory Data Register.
10	Write 16 bits from the Memory Data Register into memory.
11	Write 8 bits from the Memory Data Register into memory.

Before issuing this instruction, disable PCB interrupts by setting Status Register bit 3 to 1. This indicates that the IOP is busy. Also, do not use this instruction if there is an outstanding interrupt request to the CP.

The D and RR fields indicate what value to leave in the Memory Address Register after performing the transfer.

D	RR	What Value to Leave in the MAR
x	00	Do not change the MAR.
0	01	Take the current MAR contents, increment by 1 and load the new value in the MAR.
1	01	Take the current MAR contents, decrement by 1 and load the new value in the MAR.
x	10	Invalid.
0	11	Take the current MAR contents, increment by 2 and load the new value in the MAR.
1	11	Take the current MAR contents, decrement by 2 and load the new value in the MAR.

CONTROL PROCESSOR COMMUNICATION BUS INTERFACE

CPBI

BIT 0 15

0 1 0 0	1 - - -	- - - C	C C C C

where the C field contains the control operand.

This instruction performs the control activity specified by the bit set in the C field.

C Field	Operation
00001	Give Buffer, see below for format
00010	Give IOP Address
00100	Send Response and Address, see below for format
01000	Activate Request Line
10000	Deactivate Request Line

The information sent back to the CP goes over the DAB0-7 lines, the MDRL0-7 lines and the CCB0-1 lines. The buffer of the Give Buffer command is sent over the MDRL lines. The IOP address of the Give IOP address command is sent over the DAB lines and the response code of the Send Response and Address command is sent over the CCB lines.

Give Buffer:

- - - G	C C A A

where G is the state of the PCBGS line, CC is the state of the CCB0-1 lines, and AA are the two lowest-order bits of the received address.

Send IOP Address:

D D D D	D D A A

where DDDDDD are the six high-order bits from the device address switch, and AA are the two lowest-order bits from the last SIO command.

CPBI: continued
Send Response and Address:

A A A -	R R A A

where AAA AA are the address bits and RR are the value to be sent back out on the CCBO-1 lines.

CONTROL DEVICE LEVEL INTERFACE

CDLI

BIT 0 15

0 1 0 1	0 C C C	C C C C	C C C C

where the C field is the control operand. The actions taken by the Device Adapter on receipt of this command depend on the DA under consideration.

HOW THE PCBI INTERRUPTS THE CP

The IOP signals the CP that an IOSW is waiting by issuing an I/O interrupt. I/O interrupts are at the lowest interrupt priority. Interrupts which have a higher priority are machine check interrupts, program check interrupts and clock interrupts.

The CP services interrupts only between instructions. Two CP instructions (CLCL and MVCL) are exceptions to this rule. Both of these "Long" instructions can be interrupted.

The CP services I/O interrupts only after interrupts of higher priority have been serviced. For example, if a disk IOP and the clock both request interrupts at the same time, the clock will be serviced first and the disk IOP will be serviced second. The pertinent Product Maintenance Manuals for the different VS machines explain which priorities they implement.

Machine checks can interrupt anything. Machine checks occur if there is a parity error in main memory or if an IOP either times out or violates protocol. Both of these types of errors indicate a system of questionable reliability and thus take the highest priority.

The operating system has the ability to mask interrupts. This means that, if the I/O interrupt bit is masked, any interrupts occuring will be ignored by the operating system.