580

VOL. II

National Brand

College Ruled White Paper

Single Subject

31-987

80 Sheets/11 x 8½

Dennison National Company, Holyoke, MA 01041

Brian R. Page

580 Fundamentals, Vol. II

## I/O Subsystem

Functions of the subsystem:

1. Relieving the CPU on direct involvement in data transfer.
2. Translating requests and response from an internal format to a standard channel protocol.
3. Moving data & control signals.

The subsystem consists of one or two IOP, each capable of using up to 16 interface handlers (IHs).

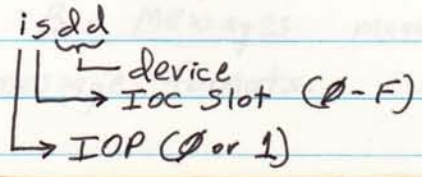
The IOP consists of:

Bus Handler - manages IOP traffic on the A & B Buses, accesses main storage on behalf of the IHs and I/O controller.

I/O Controller - a microprocessor (IOC).

Interface Handlers (IHs) - buffer data and perform tag & bus sequencing. Individual IHs are dedicated to a channel. Different type IHs are required for block mux or byte mux channels. Up to four channels on an IOP may be byte mux. An IH may address up to 256 devices. All IHs are located in the SSF (Ø2).

Channel Configuration - The channels used by the SCP must be mapped to an SCA (System Channel Address), which, in turn, may be mapped to any slot on either IOP. The map is loaded into the MBC and by the time a request reaches the IOP it is:



## I/O Overview

CPU - decodes I/O instruction
- passes instruction to IOP via MBC

MBC - assigns IOP and IH

IOP - buffers data
- passes interrupts to the interrupt router.

IH - device selection & data transfer
- buffers data
- passes interrupts to the IOP.

## IOP Bus Messages

Most I/O bus messages are two quarterlines.
Interrupts are one quarterline.
Data messages are five quarterlines.

IOP Bus messages may be either solicited (data fetch) or unsolicited (start I/O).
Start I/O Bus message:

| | Byte | Meaning |
|---|---|---|
| Header | 0 | SCA |
| | 1 | unused |
| | 2 | DCA |
| | 3 | device |
| | 5 | op code |
| | 6 | msg originator |
| | 7 | slot number + BH RAM line number |
| Data | 0-3 | CAW |
| | 4,5 | Domain id |
| | 6,7 | Domain CPU |

See the Bus Messages manual for descriptions of all bus message formats.

# IOP

One or two IOP MCCs may be installed in the LSI stack.

IOP Ø → slot 5
IOP 1 → slot 4 (optional)

The IOP has two components: IOC & Bus Handler. The IOC is a 16 slot barrel processor which executes IOP microstore and nanostore. Sixteen processes run concurrently on the IOC and are serviced serially. Each process has its own program counter, working storage, flags, stack pointer, & stack areas.

During each IOC cycle, a different process (barrel processor slot) gets one microinstruction started. Fetching & execution of microcode by the IOC is governed by an 8 cycle pipeline. Thus, one instruction is completed every 8 cycles.

Each IOC process (slot) is physically connected to an IH in the SSF. Thus a process is dedicated to an IH, but the SCA is not dedicated to a process.

## IOC Functions:

Decodes the bus message to determine the operation and the main store address to be used.
Causes the IH to perform initial device selection.
Transfers control to the IH for data transfer.
Receive interrupt from IH and causes the bus handler to create a message for the CPU.
Transfers status (CSW) back to the CPU.

Bus Handler - the task of the bus handler is to interface with other components. To do this, it has four interfaces:

B-Bus Interface
A-Bus "
IOC "
IH "

To permit the IOC to use the IHs, the BH maintains:

1.) a Command Status File (CSF) to track message status, and
2.) a Data Status File (DSF) to track data transfer operatrons.

The Bus Handler RAM contains 8 lines (256 bytes) of storage for each of the 16 processes. This is the work area and buffer for each IOC slot.

Status for each device currently being accessed through an IH is maintained in a Subchannel Buffer Storage (SBS) area in main storage.

Interface Handlers

IHs are BLCs in the SSF. Row B2 is dedicated to IOPØ, and row B3 is dedicated to IOP1. If an optional CE frame is installed it will not contain any IHs, only cable sockets.

A maximum of four of the sixteen channels on an IOP may be byte mux channels. They require special IH BLCs and must be installed as such:

| | slot | SSF card |
|---|---|---|
| First | 7 | 24 |
| Second | 8 | 32 |
| Third | 6 | 22 |
| Fourth | 9 | 34 |

Also, a maximum of four channels may be used with a two-byte interface (two bus cables).

The byte mux IH has 8k additional RAM over that on a block mux IH. The additional storage is used to hold one SBS device.

During a read, data moves from the device to the IH, to the Bus Handler. On a write, data comes from the Bus Handler through the IH to the device. The <u>In pointer</u> of the IH contains the location where to place the next byte in the buffer. The <u>Out pointer</u> informs the Bus Handler where to take the next byte. A <u>Stop pointer</u> determines the quantity of data to move in or take out.

The IH consists of:

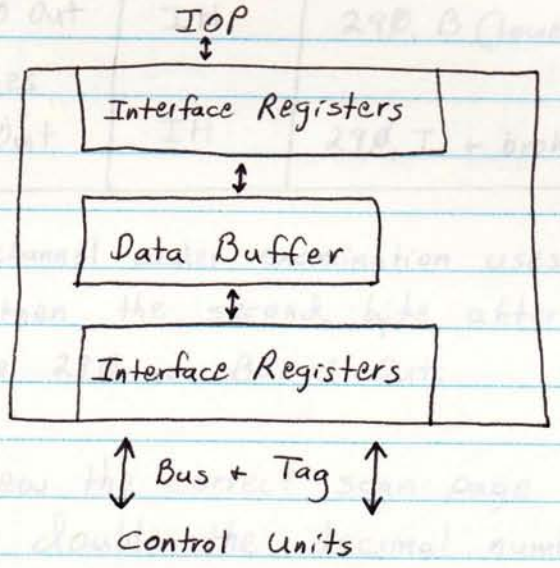
1. Data Buffer
2. Interface Registers (2 sets: IOP/IH and IH/Device)
3. Three State Machines

Bus Handler Interface State Machine (BHIF) - transfers data between the BH and the IH.

Data Sequencer State Machine (DTSQ) - moves data between the IH and device.

Tag Sequencer State Machine (TGSQ) - controls the channel tags.

These state machines are hard-wired and relieve the IOC of direct involvement in device control. The BHIF and DTSQ work together to step data through the interface registers and data buffer.



## Bus Handler - IH Data Flow

### Write Operation

| | Register | Location | Scan Page* Label |
|---|---|---|---|
| 1. | BBR | IOP | 270, BBR (8 bytes) |
| 2. | BH RAM | IOP | ACS: D IOPn slot= |
| | The IH Pointer in the BH DSF tracks data movement. The BH moves 2 bytes at a time out to the IH, incrementing the IH pointer from Ø to 31. | | |
| 3. | IHR | IOP | 270, IHOUT (2 bytes, parity checked) |
| 4. | BH Data Out | IH | 290, B (upper right), 2 bytes |
| 5. | IH RAM | IH | Data are parity checked before storing |
| | Data are moved in and out of IH RAM one byte at a time under the control of In, Out, + Stop pointers. | | |
| 6. | Bus Ø Out | IH | 29Ø, B (lower left) 1 byte, Parity chk occur on exi |
| 7. | Cables Tag Out | IH | 29Ø, T + broken out 291, TAGO |

If the channel under examination uses a two-byte interface, then the second byte after Bus Ø Out on scan page 29Ø is Bus 1 Out.

* To view the correct scan page for any IH on either IOP double the decimal number of the IH and add this to the pages listed below.

| | IOPØ | IOP1 |
|---|---|---|
| Block Mux | 29Ø, 291 | 59Ø, 591 |
| Byte Mux | 34Ø, 341 | 64Ø, 641 |
| IOP Latches | 27Ø | 57Ø |

## Read Operation

| | Register | Location | Scan Page, Label |
|---|---|---|---|
| 1. | Cables | | |
| | Tag In | IH | 29Ø, T (upper - on cable / 2nd - in Reg), Broken out on 291, TAGI |
| 2. | Bus Ø In | IH | 29Ø, B (upper - on cable / 2nd - in reg), 1 Byte, Parity chk. |
| 3. | IH RAM | IH | Parity chk occurs prior to store. |
| 4. | BH Data In | IH | 29Ø, B (lower right). 2 bytes |
| 5. | IHD | IOP | 27Ø, IHIN (2 bytes, parity checked) |
| 6. | BH RAM | IOP | ACS: D IOPn slot# |
| | ABR | IOP | 27Ø, ABR (A-bus msg header) |
| 7. | A-BUS | | |

## Scan Page 270

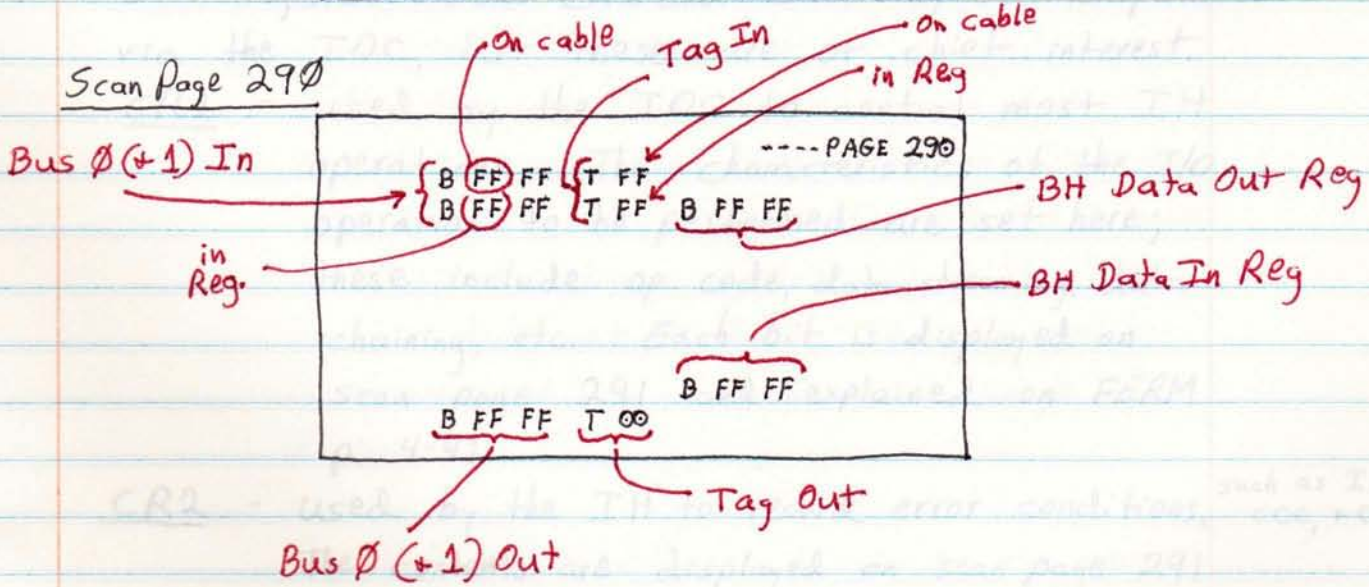
```
--- BH DATA PATH ------ IOP0 --PAGE 270
        BBR 00 00  00 90  18 18  04 70
       DBBR FF FF  FF FF  FF FF  FF FF
       IHIN 00 00  IOCIN 43 F0

      IHOUT 00 00  IOCOUT 00 00
        ABR 00 00  00 00  02 02  43 F0
```

Delayed BBR → allows holding BBR for one cycle before storing into BH RAM on a write operation.

Used for IOC to IH communications.

## Scan Page 29Ø



IOC to IH Interface:

The IOC controls operation of each IH by altering and reading registers on the IH. Although the IOC uses several registers on the IH, the data path has a common element:

IH to IOC:

| | Register | Location | Scan Page, Label |
|---|---|---|---|
| 1. | IOC IN | IH | 27Ø, IOCIN |
| 2. | B Operand | IOP | 28Ø, B (lower right) |
| 3. | RR | IOP | 276, RR |

IOC to IH:

| | Register | Location | Scan Page, Label |
|---|---|---|---|
| 1. | RR | IOP | 276, RR |
| 2. | IOC Out | IH | 27Ø, IOCOUT |
| 3. | CR1 | IH | 291, CR1 (Broken out by bit) |
| | CR2 | IH | 291, CR2 (Broken out by bit) |

Other registers besides CR1 and CR2 may be manipulated via the IOC, but these are of chief interest.

CR1 - used by the IOC to control most IH operations. The characteristics of the I/O operation to be performed are set here; these include op code, data streaming, data chaining, etc. Each bit is displayed on scan page 291 and explained on FERM p. 4-41.

CR2 - used by the IH to record error conditions, such as IFCC, CCC, & CDC. The contents are displayed on scan page 291 and explained on FERM p. 4-4Ø.

BH Registers and Files:

B-Bus Register (BBR) - holds incoming B-Bus messages before they are placed into BH RAM. See Scan Page 270, BBR.

Delayed BBR (DBBR) - a staging register for BBR data that holds data for one cycle before storing into the BH RAM. This register is used when the BH is busy. See Scan Page 27Ø, DBBR.

A-Bus Register (ABR) - holds data from the BH RAM that is ready to be transmitted on the A-Bus. See Scan Page 27Ø, ABR.

Command Status File (CSF) - holds the status of messages sent or received by the IOC. The IOC interrogates the CSF to find if unsolicited messages have arrived. The CSF, in a sense, drives the BH because requests for services originate outside the BH. See Scan Page 273.

Data Status File (DSF) - controls data transfer operations into and out of BH RAM. Holds the op code indicating the type of operation and the IH pointer specifying the next byte to use. The BH RAM holds a line of storage (64 bytes) and moves it two bytes at a time. The IH pointer is incremented by 1 through a range of Ø to 31.

The DSF and CSF are displayed on scan page 273 and the contents are explained on FERM p. 4-43.

## Cables

**Red Ribbon Cables** - connect the IOPs to the IHs in the SSF. These cables may not be swapped.

**White Ribbon Cables** - connect the IH to the tailgate. Each IH has three jacks: J1, Tag; J3 Bus Ø; and J2, Bus 1 (optional). These cables may be swapped for diagnostic purposes, although this is difficult.

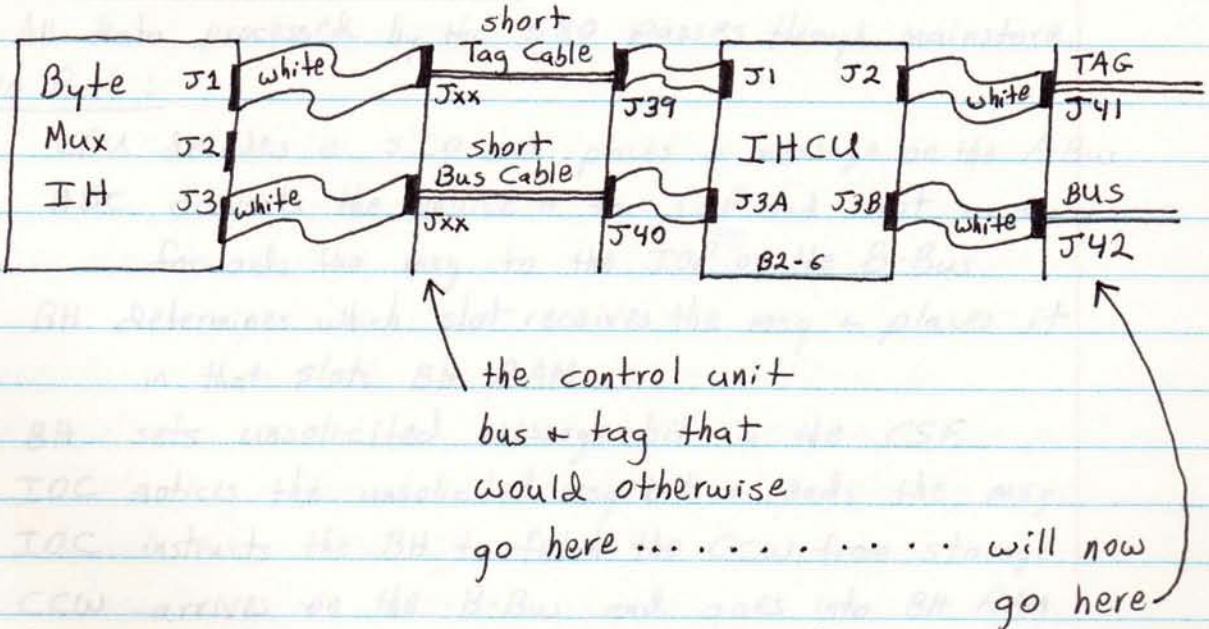
**Bus + Tag** - connect the tailgate to control units. The SSF contains slots for 16 channel sets. Four of these may be used for either one or two byte interfaces. Additional channels may be plugged into the CE frame (Ø3). Panel A1 holds channels 1Ø - 17 and panel A2 holds 18 - 1F. Each panel has two slots which accomodate a two byte interface.

Operator Service Interface (OSI) - consists of:

1. Byte Mux IH (in slots 7, 8, 6, 9)
2. IHCU (SSF, B2-6)
3. Driver C

In OPER mode, the console emulates a 3277. The IHCU functions as the control unit for the console 3277 and must be placed in the daisy chain of any other control units sharing the same channel.

A short pair of Bus and Tag cables connect the IHCU with the Bus & Tag connectors for the byte channel. Then, the other control unit Bus & Tag connectors go to the IHCU jacks.




The result is that the IHCU has been inserted into the daisy chain.

OSI Data Path:

CPU - SIO decoded
Bus Message sent to IOP (via MBC)
IOP - Bus Handler receives data from mainstore
IH - Byte Mux IH pushes data out bus cable
IHCU - intercepts data
Driver C - passes data to console MCC
Console MCC - slot 5, CSA
slot 3, console channel
(Both under control of slot Ø, console CPU)
to display station
CRT - data displayed

I/O from Device to MSU

All data processed by the 580 passes through mainstore.

I/O Path:

CPU decodes a SIO and places a message on the A Bus.
MBC converts the device # to IOP and slot and forwards the msg to the IOP on the B-Bus.
BH determines which slot receives the msg & places it in that slots BH RAM.
BH sets unsolicited message bit in the CSF.
IOC notices the unsolicited msg bit & reads the msg.
IOC instructs the BH to fetch the CCW from storage.
CCW arrives on the B-Bus and goes into BH RAM.
IOC instructs IH to perform selection sequence.
IH juggles tag lines.
IOC instructs IH to perform data transfer.

The sequence splits here depending on whether the operation is a read or a write.

Read:

Device transmits one byte at a time over bus cable.
IH stores in its data buffer.
BH RAM accumulates data until it has a full line (32 bytes)
BH builds bus msg. to send data to main store. (A-Bus)

Write:

BH fetches a line of data from mainstore.
MSU places data on B-Bus.
IH gets data from BH RAM + stores in data buffer.
IH places data on bus cable one byte at a time

Ending Sequence - common for reads and writes:

Device signals end to IH
IH sets status bits to be read by IOC.
IOC instructs BH to forward interrupt to IR.
IR passes interrupt to CPU.

## I/O Troubleshooting

CCCs, CDCs, and IFCCs usually do not cause the entire 580 to clock off. When a channel occurs, the path is reset. If the reset is not successful, the path may be taken offline by the SCP or macrocode. The SCP continues to run.

Debugging information, therefore, is usually restricted to channel logs (CLOGs) preserved on the hard disk. 100 clogs may be stored. When the maximum is reached, the most recent is over-written.

CLOG Format:

Top line - type of channel (byte/blk), mode (370/XA), and CPUID.

Registers on the IH:

Tags IN/OUT } shows the bits set on all
Bus0/1 IN/OUT } bus + tag connectors.

FERM page 4-39 explains the contents of each of these registers. For example, Tag in may indicate address in and the bus in may not indicate the same address issued through bus out. This would most likely indicate a cu or cable problem. Bus registers are as useful as the tag registers.

CR - this field contains both CR1 and CR2. CR2 is the error register. Its contents are explained on FERM page 4-40.

ERROR PC - program counter for the IOC. This lists the micro instruction being processed when the error was detected. A unique EPC Dictionary is provided for each IOP microcode level (in microfiche). Checking the EPC may reveal detailed information on the error.

Error History Registers - the error registers for each IOP (4 for each) are described on scan page 94. See the Scan Page Reference Manual.

Working Storage - the lower portion of the clog contains the IOP working storage. The content varies with the type of chamel. See the FERM Channel section table of contents for a map of each type. Working storage holds the CSW, SCA, and device number plus lots more.

Commands:

Q CLOG ALL - one line description of each log in chronologic order.

D CLOGn - displays selected log.

PROT CLOGn - protects designated CLOG from being over-written.

UNPROT CLOGn -

COM CLOGn string - adds a comment to the designated CLOG.

P CLOG ALL - purges all unprotected CLOGs.

CLOGs will also be printed with Amdahl EREP.

Scan Pages - for I/O errors which occur while in check stop, certain scan pages are very useful.

Page 269 - provides a high-level over-view. The left 3/4 of the page list two bits for each IH. The right portion signals an IOP error.

Page 291 (or appropriate page for suspect IH) - shows CR1 and CR2 as well as all tag in and tag out bits.

IOP errors are indicated on:

IOPØ → 283 and 275

IOP1 → 586 and 575

Page 283 - all field implicate the IOP with the exception of BOP in the lower right. This may indicate a bad IH.

Page 275 - most fields implicate the IOP with these exceptions:

BBus CNTL } possibly bad MBC
B BUS }

IH IN } possibly bad IH.
IH OUT }

Scan pages may be defined in advance to display all installed channels. A maximum of 64 defined screens are available.

D 54Ø/D PA 29Ø 291 292 293
SAVE SSA
D 54Ø

To examine screen assignments:
Q SSA or (1-32)
Q SSA 33 (33-64)

SMAPs - troubleshooting flowcharts are provided with SMAPs in the Maintenance Manual.

BUGPROCs may also be used for errors caught in check stop.

Part swaps - terminators, white ribbon cables, bus, and tag cables may be swapped without powering down.
When swapping IHs, be sure not to mix byte and block BLCs. If they must be swapped, update the configuration (CONF SCAx) and the machine-level (ML) file. Byte and Block IHs are reset differently.

## SIO Processing

1. I-Unit decodes and requests the CAW.
2. E-Unit builds A-Bus msg based on CAW.
3. MBC substitutes IOP # and slot and places message on B-Bus for the IOP.
4. BH receives messages.
5. IOC detects message and requests CCW from the BH.
6. IOC manipulates tag lines, receives status, and releases the interlocked I-Unit (after some 3800 machine cycles).
7. IOC orders data transfer.
8. IH performs data transfer.
9. IOC posts interrupt via the IR.

## SIO Handloop (370-mode)

A M 1ØØ 9CØØØØ9Ø 82000108 020A0000 00FEFEFE
(9CØØØØ9Ø: device (the console address))

A P Ø 1ØØ ← alter PSW for location 1ØØ.

A M 48 ØØØ24848 ← set up CAW

A M 24848 Ø1Ø6DADA ← set up CCW

A M 6DADA F1F2F3F4F5F6F7F8 ← data

A M 78 ØØØ2ØØØØ ØØØ78Ø78 ← I/O new PSW

Press Start (OP 5).

# Macrocode

The macrocode software resides on the hard disk in the /sys file. During IML, it is loaded into the lower portion of real storage.

Logical processors are numbered Ø and 1.

The default configuration maps all channels to the first activated domain.

Main storage is allocated in 64K increments. The quantity of memory for a domain is specified in the configuration. If macrocode assigns the exact location it is allocated from high to low.

Two IOCDS files may be stored on the XADATA ACS minidisk. An IOCDS may be created with the IOCPGEN domain (370 mode). Remember to DDR backup the /xadata minidisk whenever the customer creates a new IOCDS.

The ACS command

CONF IOCDS #

may be used to initially select an IOCDS file. An IOCDS must be present for the 580 to reset in XA mode. After the SWAPIOCDS non-frame command is issued, an IML C must be performed. The ACS command:

Q CF IO

will list the IOCDS in use along with a date and time stamp for its creation plus any comments.

Pressing the interrupt button on a master console will cause the master console function to move to the next available Amdahl console.

Macrocode will display the MI frame and subsequently execute AUTOACT whenever a macrocode cold start has been performed.

The Console Log contains the last 25 lines of console input and synchronous messages. The system log holds 50 lines of asynchronous messages. The macrocode Event log may be viewed with the ACS command:

D_MACLOG

The underscore is literal. The Event log is contained in the file |SYS|EVENT.LOG.

Macrocode files are usually called MACROA and MACROB and reside in the |sys minidisk directory. Also, a load map (for patching & debugging) is supplied with each macrocode level.

Macrocode is generally installed with the ACS command: INSTALL LIST ALL.

Patch files also reside on |sys. They are designated:

|h#XXXXYY

where: XXXX = HWS Number
YY = version

The patch file contains the name of the macrocode module plus address offsets and old and new data.

Patches may either by issued as Limited Availability Patches (LAP) or General Availability (GAP).

The PATCH command has five operands:

H - produces a patch history in a file called PATCH_AUDIT.

R - rererses a patch

S - produces a summary of patches in the file PATCH_AUDIT.

V - verifies, but does not apply the patch.

C - display the checksum for the patch.

To activate the patched macrocode, issue:

IMC C or

RS SYS C/IMC or

RS PO

To perform a macrocode warm start (thus preserving macrocode control blocks and domain storage) issue the command:

IMC

Troubleshooting

Macrocode storage may be dumped to disk or tape with the MACDUMP facility of ACS. This is documented in the 580 Macrocode Reference Manual.

The trace feature tracks events within macrocode and may be controlled through the TR frame (which requires the FE key on). The Hook feature may be used to limit the events being traced.

If macrocode dies (does not refresh the screen after pressing clear) go to MAINT mode and issue the ACS command:

D PSW

to see if it is in a disabled wait and if a three digit wait state code has been loaded. These are documented in the 580 Macrocode Reference Manual. Do not automatically perform a reset. Recovery may be possible without destroying domain storage.

For some errors which are recovered from macrocode, a complete machine check log may not be taken. These errors are recorded in an Error log. The current error log may be displayed with the ACS command:

Q ELOG

and the contents may be viewed with the command:

D Elog #

Files:

Program files - stored on \sys with the names MACROA and MACROB. Patch files also reside here.

Configuration files - stored on \SYS. The actual filename consists of the name assigned via the CC frame plus the literal C0 and a three digit number indicating the number of times this configuration has been saved.
Example: MVS.C0005

Function Lists - stored on \SYS with a file name consisting of the named assigned on the FC frame plus the literal FL and a three digit number indicating the number of times the file has been saved.
Example: REIPL.FL012

IOCDSs - stored on the \XADATA minidisk with a file name consisting of IOCDS plus the literal A and a single digit identifier.
Example: IOCDS.A1.

Log Files - stored on \SYS with the name EVENT.LOG.

## XA Architecture

The XA architecture permits 256 channels each with 256 devices. This allows 64K hex devices. As implemented, however, only 4K hex devices are allowed. This is limited by the memory space currently allocated to the hardware control blocks. Similarly, XA architecture permits up to 8 channel paths per device. However, a maximum of 4 is all that is currently implemented.

In the IOCP process, a sequential subchannel number is assigned to each device. The subchannel number corresponds to a Subchannel Control Block (SCB). A single device will have only one SCB. If the device is shared by multiple domains a duplicate "related SCB" will be created.

In 370-mode, I/O interrupts are queued on the channel. Thus, there will exist as many queues as there are channels. In XA mode this is different. I/O interrupts have a subclass ranging from Ø to 7. The Interrupt Subclass (IS) is a characteristic of the device and is specified in the SCB. The SCP can change a device's subclass at any time. If an XA I/O interrupt must be queued, it is placed in one of the eight IS queues.

## Control Blocks

Hardware:

SCB - describes a device: unit address, subchannel number, device number, subchannel status, and ISC. The information in the SCB is available to the SCP. The store Subchannel (STSCH) instruction causes Macrocode to format the SCB data into a SCHIB. The contents of the SCB (like the ISC) may be modified by the SCP via the Modify Subchannel (MSCH) instruction.

CCUCB - describes a control unit: a control unit may be attached to four CHPIDs and can service a maximum of 64 devices. The CCUCB list the paths, CU type, and contain usage information. Macrocode examines the CCUCB when making pathing decisions.

QCB - Queue Control Block: A QCB describes an Interrupt subclass for each domain. Thus, each domain could have eight QCBs. In practice, QCBs exist only for the Interrupt subclasses enabled in each domain. The IOP queues interrupts for each domain off an appropriate QCB. In MVS, IS 3 is used for paging devices; IS 5 describes most other I/O devices.

SBS - Subchannel Buffer Storage resides between domain storage and macrocode. In 370-mode, and for XA byte mux channels, the SBS contain status about an ongoing I/O operation, including the executing CCW and the byte count. In XA-mode, the SBS fuctions as a pointer to the SCB.

Software Control Blocks

SCHIB - created by Macrocode for use by the SCP from information in both the SCB and the CCUCB.

ORB - is an operand to the SSCH instruction. The ORB contains the first CCW, interrupt parameter, and the Logical Path Mask (LPM). The interrupt parameter is used to link the SSCH with its later interrupt.

IRB - Interruption Response Block: contains the SCSW and the ESW describing the ending status of the I/O operation. The IRB is delivered in response to a Test Subchannel (TSCH) instruction.

## Displaying Control Blocks

Macrocode DD frame - lists device numbers, device addresses, and subchannel numbers.

D SBS xxxx - where xxxx = device address.

D SCB xxx - where xxx = subchannel number.

## Path Management

When the I-Unit detects an XA I/O instruction it passes control to Macrocode Fast Assist Mode (FAM). Macrocode determines which paths to a device are available and operational, and selects one of these. The IOP is then instructed to perform the I/O. If the device is busy, the IOP will queue on the selected path for 100 ms. After this, the IOP will interrupt Macrocode and Macrocode might try the I/O down another path.

When an interrupt comes back from a device via the IOP, macrocode builds an IRB from information in the SCB.

## Start Subchannel

To begin an I/O operation, the SCP builds the channel program in memory, an ORB containing interrupt parameter, first CCW, and CPM, and places the system ID (SID), containing the subchannel number, into GPR1. Then the SSCH is issued.

The I-Unit decodes the SSCH, causing FAM entry into Macrocode. Path selection occurs and a request subchannel bus message is issued. The CPU is not interlocked. Processing may be resumed after SSCH simulation. The IOP receives the bus message via the MBC and fetches the SCB. The CCUCB is checked for control unit busy and if so, the request is queued on the chpid. If the chpid cannot do the I/O, the IOP returns a deferred condition code to macrocode permitting the selection of an alternate path.

## Troubleshooting

Problems are usually detected by IOC microcode which causes a channel logout and notifies Macrocode. There are two types of logouts: SCB/IRB which is usually an interface control check, and a CRW or Extended Logout which may occur when the device address failing is not known.

In addition to these CLOGs, other documentation includes:

Error Log
Macrocode System Log
EREP
Console messages

The IOCDS printout may also be of great value when troubleshooting.

Two types of problems are typical:

1. Start Pendings / Missing Interrupts - when the I/O is initiated a pending bit is set in the SCB. This prevents another operation from being started for the device. When the operation starts, the pending bit is cleared. A start pending conditron exists if the procedure times out for the SCP and the pending bit is still on. This means the I/O has not even started.

   If the operation has timed out, but the pending bit is clear then the operation is not complete. If channel end has not been received then the channel program has not yet completed. If device end is missing, then the channel program completed but the device is not yet done. These errors cause Missing Channel End and Missing Device End messages, respectively.

   In any case, the SCP clears the request and retries the operation. After some number of failures, the chpid may be placed offline.

2. Failing I/O Instruction - this type may cause software abends and event logs, but not machine check logs.

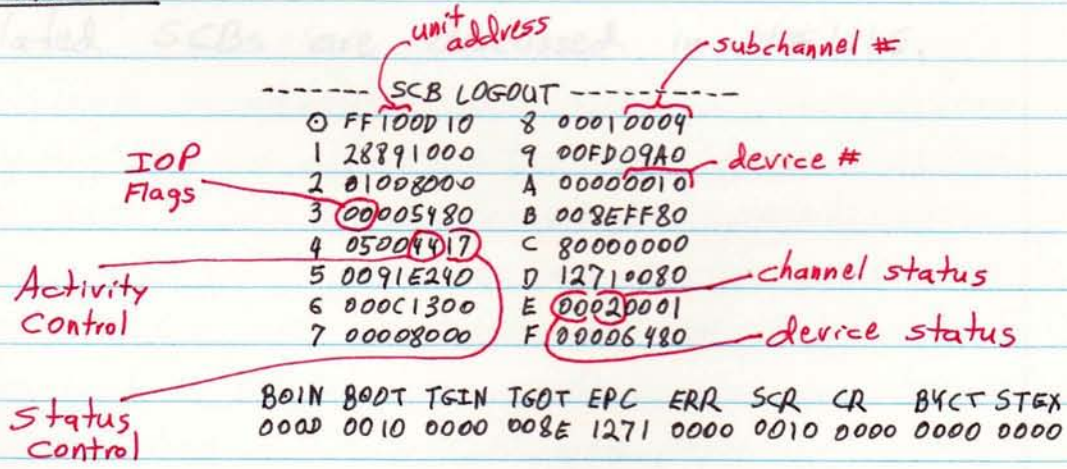
Macrocode itself may be a source of XA I/O errors. Macrocode is a FRU and may be suspect when there are few other symptoms. Before resetting Macrocode (via IML) take a dump. A large number of XA I/O problems can be diagnosed from a Macrocode dump.

Taking a Path Offline: CHPIDs are frequently taken offline to defer working on a failing control unit or device. To do this:

1. Take the channel offline from the SCP. Insure that it really comes offline.
2. Detach the channel through Macrocode (DI frame).

Logouts

SCB/IRB




Extended Logout - this is a copy of IOC working storage and lists (with named fields) slot, Bus Ø in and out, tag in and out, and other information. The EPC is found in word 3F of working storage (the last two bytes of working store). Consult the EPC Dictionary. To decode, see the FERM Working Storage maps.

## XA Configuration

Two types of configuration are necessary:
1. IOCP relating devices to control units to chpids.
2. ACS configuration relating SCA to IOP slot.

The SCA must correspond to the CHPID.
Valid CHPIDs are:

| UP | MP |
|---|---|
| 00-07 | 40-47 |
| 10-17 | 50-57 |
| 20-27 | 60-67 |

Any IOP slot may be assigned to any SCA.
However, a one-to-one correspondance is recommended.

Related SCBs are discussed in H#1116.

# DP and MP Systems

UP - Uniprocessor
DP - extra LSI Stack (also called AP)
MP - Two complete systems

The ACS commands JOIN and PARTITION create and separate multiple processors. After each command ACS will perform a Power On Reset (CRS PO). These commands should only be issued from the Master console.

In JOINed mode, some ACS commands may be directed at one CPU or another by using the CC0 and CC1 commands. Also, in JOINed mode the ACS status display area includes status of both processors.

scan pages + mch chk logs

## DP Configuration

The second stack is referred to as the remote stack and is addressed as CC 1. It contains no IOP MCCs but it <u>does</u> have a second console MCC. The second console MCC permits error recovery and scan operations to occur. The two console MCCs are connected via the Inter-Console Data Link (ICDL).

The two LSI stacks are physically connected via two bi-directional red ribbon cables to the right side panels. Each stack has Remote A-Bus and Remote B-Bus connectors. Each MBC can receive messages on its A-Bus and direct them via the B-Bus to the other MBC.

One oscillator drives both stacks and requires a cable.

Since in a DP there is only one SSF, BCS must load microcode into both console MCCs. The remote MCC is reset and loaded using the APIMPL1 and APRSPO CPROCs instead of the usual IMPL and RSPO CPROCs. BCS loads console Ø first and then console 1 via ICDL.

Configuration

A new DP should have both UP and DP configuratrons. First, in PARTITIONed mode, create a configuration listing all memory and channels. Then issue the SAVECONF command. This configuration will be used whenever the PARTITION command is issued.

Next, issue the JOIN command (which will fail). Update the QML file to include the second stack and issue SAVECONF. This creates a file which will be used for any subsequent JOIN commands.

During ACS power-up, the system will be place in the same configuration (if possible) as it was in at power down. If it was powered down as a UP, it will come up as a UP.

The system can run without CPU1, but not without CPUØ. Also, the same microcode is loaded into both stacks; therefore, the MCC hardware levels must accomodate the same microcode. The remote stack may NOT be powered down without impacting CPUØ, even when running in partitioned mode.

## MP Configuration

MP systems have a primary CPU and an alternate CPU as well as a primary and alternate ACS. The primary ACS is customarily assigned to the primary CPU. The ACS master console is always on the primary ACS. An MP has two MOCs with two active ACSs; however, the primary ACS is responsible for error logging while the secondary ACS is available as backup to the first. The primary and alternate CPUs are designated by jumpers on the CPU ID card.

The ACS status display, in the upper left corner, indicates whether a console is Master or secondary, and to which processor it is attached.

An MP can run with only one ACS active. The only impact will be on the OPER mode of the consoles connected to the failing side.

Control may be passed to any console:

1. A display station on the primary side may be given control with the command: XFER CONTROL TO DSx
2. A display station on the primary side may seize control with the command: REQ
3. A display station on the secondary ACS may be given control with the commands: CC x / XFER CONTROL TO DSy
   where: x = other CPU
   y = Display Station #
4. A display station on the secondary ACS may seize control with the command: REQ FORCE
   This last method should be avoided.

Resets : ACS uses the configuration files on the Master console to perform 580 resets. Therefore, both hard disks must duplicate all files. After updating the system configuration on the master console, issue the SYNCCONF command. ICDL will be used to the necessary files from the master to the secondary.

Logs - Machine Check logs will be written to the hard disk on which ever side has the primary ACS. This log will contain scan pages for both systems. Use the CCØ and CC1 commands to view each set while looking at the logs.

BCS - An MP has two SSFs and therefore two BCSs. Each resets and loads its own console MCC. The MPIMPL CPROC is used instead of the ordinary IMPL CPROC. This sets the MP mode bit in the console MCC ODR register. The two BCS systems do not communicate or effect the opposite systems.

Power-up - the secondary CPU should be powered up first, followed by the primary within one minute.

Configuration : each Processor should have its own UP configuration in order to run partitioned. After creating the JOINed configuration, issue the SYNCCONF command.

Memory - the memory address assigned in the configuration must be sequential. The second CPU must begin where the first CPU's ended. This is done with the conf mix commands.

## 580 Troubleshooting

Diagnostics:

1. Atlas floppy-based
2. BCS CDM floppy-based
3. System floppy diagnostics - may be copied to the hard disk /diag directory. These are used to diagnose very specific parts of the 580.
4. Tape based - ADAM
5. System Level - simulate the load of a SCP and should be run in check stop. DIRT and ALPHA are most commonly used.

DIRT is the most popular diagnostic. Read the user's guide section in the Diagnostic Reference Manual. DIRT usually lives on the /diag or /user minidisks and is loaded into a domain (XA or 370) via the LDMEM command. A PSW restart begin execution. DIRT may also run in native mode (without Macrocode) but only on a 370-processor.

Scan Pages - Some new scan pages are available. Pages 692, 693, 694, and 695 offer a high level summary of system error latches. Review HWS for more information.

To master diagnostic techniques, become thoroughly familiar with:

1. BUGPROCs Manual (page 5-1).
2. Scan Page Reference Manual.
3. Diagnostic Reference Manual (2 vols).

# Table of Contents

## Vol. I

Vol. II