MAINTENANCE CHANNEL

Figures

Figures (continued)

Tables

Tables (continued)

Document Organization

Many of the topics described in this document are interrelated. Understanding the general concepts first enables you to more easily comprehend all maintenance channel operations. For this reason, more detailed information describing theory of operation is placed near the end of this document, after the general concepts of sanity code, maintenance commands, route codes, loop controllers, logic monitors, and master clear levels. Block diagrams (11x17 pages) showing signal paths are located at the end of this document.

Maintenance Channel Functions

The maintenance channel provides a channel for performing the following CRAY T90 series system functions:

- Configuring the system
- Master clearing CPUs
- Master clearing memory
- Initializing the system
- Enabling and disabling sections of the system
- Monitoring system activity
- Selecting test points
- Initializing I/O resources
- Sending and controlling sanity codes
- Using the mainframe maintenance environment (MME)

Maintenance Channel Paths

Each half of a CRAY T932 system has a separate and independent maintenance channel. (The CRAY T916 and CRAY T94 systems each have one maintenance channel.) The use of two maintenance channels enables the power-down of half of a CRAY T932 system while the other half is running normal operations.

The maintenance channel ports on two I/O modules in a CRAY T932 system are not used. If one of the I/O maintenance ports fails, the maintenance channel can be recabled to a different I/O module; however, this changes the system configuration.

The maintenance channel begins as a LOSP channel that is routed from the support system VME chassis to a maintenance channel connector on an I/O bulkhead. From the bulkhead, the maintenance channel connects to a maintenance port interface on one of the I/O modules, as shown in

Figure 1. Because the maintenance channel from the VME chassis to an I/O bulkhead is a LOSP channel, it has two cables, one for each channel direction. (There is one bulkhead for each mainframe quadrant.)

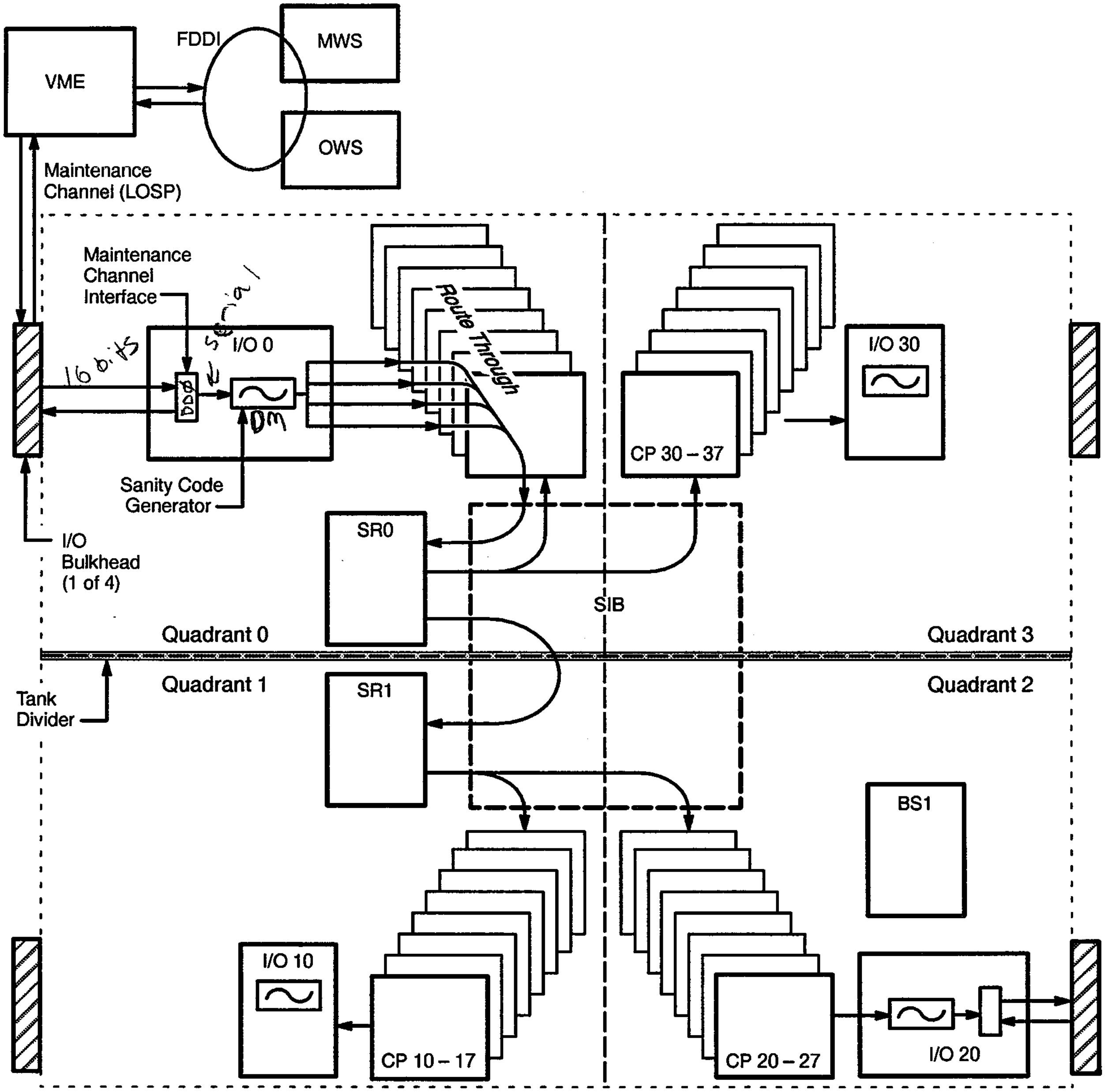


Figure 1. CRAY T932 Maintenance Channel Structure

The six signals that compose the maintenance channel (Figure 2) pass from the maintenance channel interface through the attached CP stack to the local shared module. From the local shared module, the signals branch out to all modules in the system.

The shared modules send signals out to other modules because there is no direct CPU-to-CPU communication; a CPU can access another CPU only through a shared module. The local shared module can access the remote shared module to communicate with modules in the other half of the system.

Maintenance Channel Signals

Starting at the maintenance channel interface (port) on the I/O module, a maintenance channel consists of six signals, as shown in Figure 2.

- Maintenance channel in and out paths are used to send commands and receive data. The maintenance channel path would be better identified as a "command path."

- Sanity code paths are used to send and receive sanity codes between modules to establish module-to-module communication.

- Error logger paths flow in the opposite direction of the maintenance channel and sanity code paths. The Error Logger Acknowledge path provides a return path for responding that one module has received an error from another module and indicating that another error can be sent.

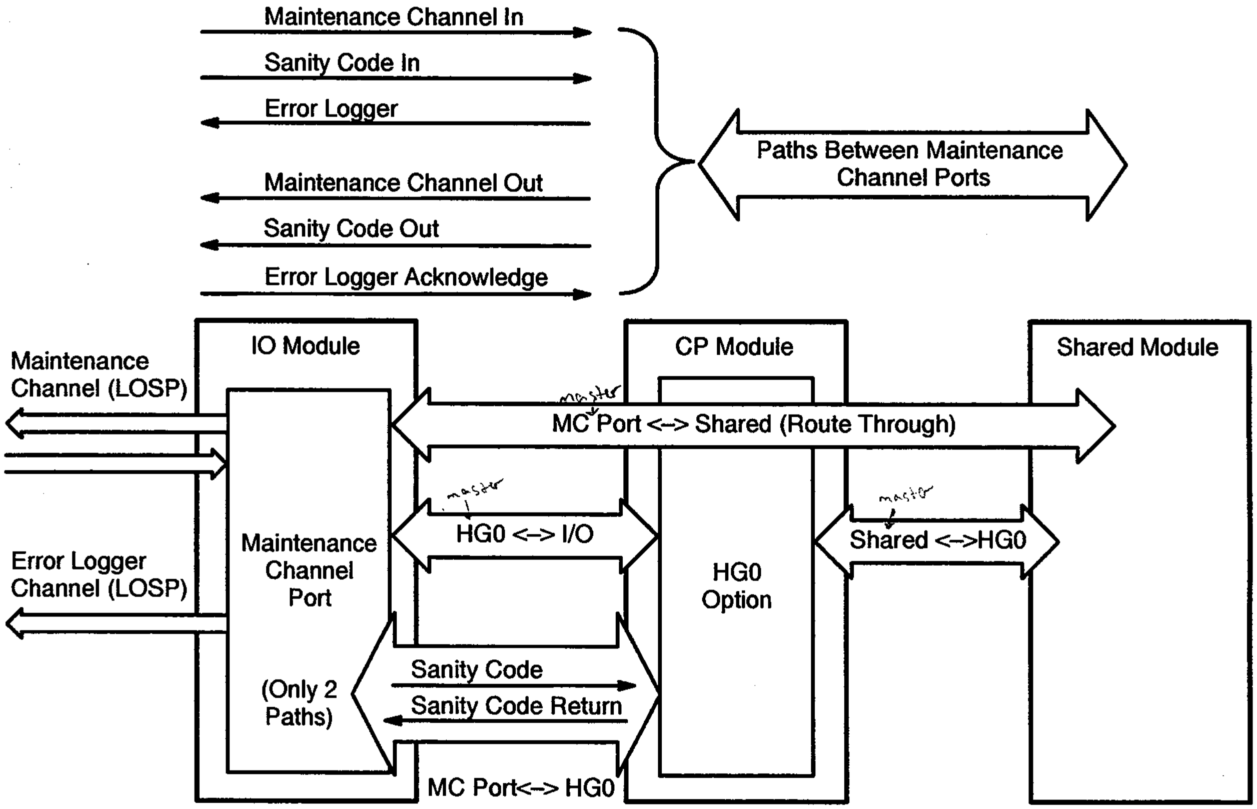


Figure 2. Maintenance Channel Port Signal Paths

Sanity Code

Sanity code is a 6-bit pattern that is used to control data flow and interconnecting processes between modules. Sanity code performs three primary functions:

- Establish paths for sending maintenance commands and data
- Enable module-to-module communication
- Establish a return path for reporting errors

Sanity code is used to ensure that a logical path exists to all configured portions of the system. Sanity codes control all signals that cross module boundaries. A module must receive sanity code through an interface or connector to another module before it can respond to signals or commands from the other module. A module must also return a sanity code to the interface or connector before a module can receive a return response. The interface between modules is either a single connector or a set of serial connectors [a connection between a CP and a network module goes through two connectors, one on each side of a system interconnect board (SIB)].

A module must receive sanity code from another module for 96 consecutive clock periods to establish a connection. Once the connection is established, the sanity code generator rebroadcasts the 6-bit sanity code pattern throughout the tree every six clock periods. If a module loses sanity code from a particular port, it still may be receiving sanity code from another port. A module must continue to receive valid sanity code in order to remain in an active state in a sanity tree structure. This means that the sanity code path is dedicated every 6 clock periods.

The absence of valid sanity code to all ports on a module causes that module to go into a master clear state; all outputs from those modules are then ignored. The "Master Clear Levels" subsection, which begins on page 45, describes the different types of master clear states.

Sanity Trees

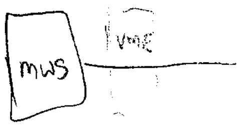
A sanity tree is a logical structure that provides a flexible and reliable control network. A sanity tree is a sanity code distribution network that is defined when the mainframe is configured. The sanity tree is used to establish control of sanity codes and provide fanout networks for sanity codes and maintenance, control, error, and configuration data. (Sanity codes travel on sanity code and sanity code return paths. Maintenance, control, and configuration data travel on maintenance channel and maintenance channel return paths. Error logger data travels on error logger paths.)

A mainframe can support several sanity trees. All directly accessible modules have a node address within the sanity tree (the node address is defined by a route code, as described later).

The actual path or structure of the sanity tree is not rigid, its structure is configured by the operator. Once a tree structure is established, it remains set at the same structure until it is changed or reconfigured.

To configure a sanity tree structure, a function code must first be sent through the maintenance channel to initialize the sanity code generator. Once the sanity code generator is initialized, sanity code can then be sent to the shared module. This sanity code is routed through one of the four CPs connected to the I/O module, as determined by the sanity code arbitration logic on the I/O module. Once the tree is established, configuration information can then be broadcast to individual options on modules.

The first sanity code that a module receives is given a special place in the system configuration structure. The initial sanity codes between each module build a tree structure as more modules are added. Maintenance, control, and configuration data are sent over this initial sanity tree; error log information is returned along this tree. The initial sanity tree structure must be recorded and saved for use in addressing particular nodes in the sanity tree.



Direct and Indirect Sanity Trees

Sanity trees are defined and configured in two different ways:

- Direct sanity trees
- Indirect sanity trees

Specific configuration commands are used to configure direct sanity trees, which result in sanity code being passed on from one module to another. The direct portion of a sanity tree includes CP, I/O, and shared modules.

Indirect portions of a sanity tree include memory modules and network modules. These portions of a sanity tree are called indirect because there are no control or configuration commands that directly access network and memory modules. There are no sanity codes to connect the network module to the memory module. These modules are configured with configuration codes that set memory degrade soft switches. This means that until the memory degrade functions are set, there is no path from CP to memory modules. There are no default paths between these modules; it is possible to configure a system without memory.

Sanity Tree Configuration through Shared Modules

A sanity tree must be configured so that the next node in the tree after the maintenance port on the I/O module is the local shared module. This is necessary because no direct communication occurs between CP modules. A CP must go through a shared module to access another CP module. The shared modules are closer to the logical center of their respective side of the mainframe's data flow.

System Configuration and Sanity Trees

System configuration over the maintenance channel is not the same set of functions and capabilities as sanity codes and sanity trees even though system configuration uses these functions. A system that has two sanity trees can be configured as one continuous system. A system with one sanity tree can be configured into two separate sections in which one section has no communication or shared storage with the other section except for the common sanity tree.

Sanity Codes in Network and Memory Modules

When a network module first receives a sanity code (always from a CP module), it automatically passes the sanity code to its connected memory modules. When a network module receives a sanity code from a connected memory module, it returns a sanity code to any CP module sending it a code. When a memory module receives a sanity code, it returns the code. (A memory module receives sanity codes from a network module in CRAY T932 systems and from a CP module in CRAY T94 systems.)

Sanity codes from CP to memory modules are controlled by memory configuration codes 70 and 71. There are eight sanity code paths from a CPU into the memory system; one for each of the eight memory port paths.

Neither network modules nor memory modules require any logic to be part of a sanity tree. Network and memory modules are never directly addressed by any control or configuration commands. (However, memory master clear function codes can clear request and data queues in network and memory modules.)

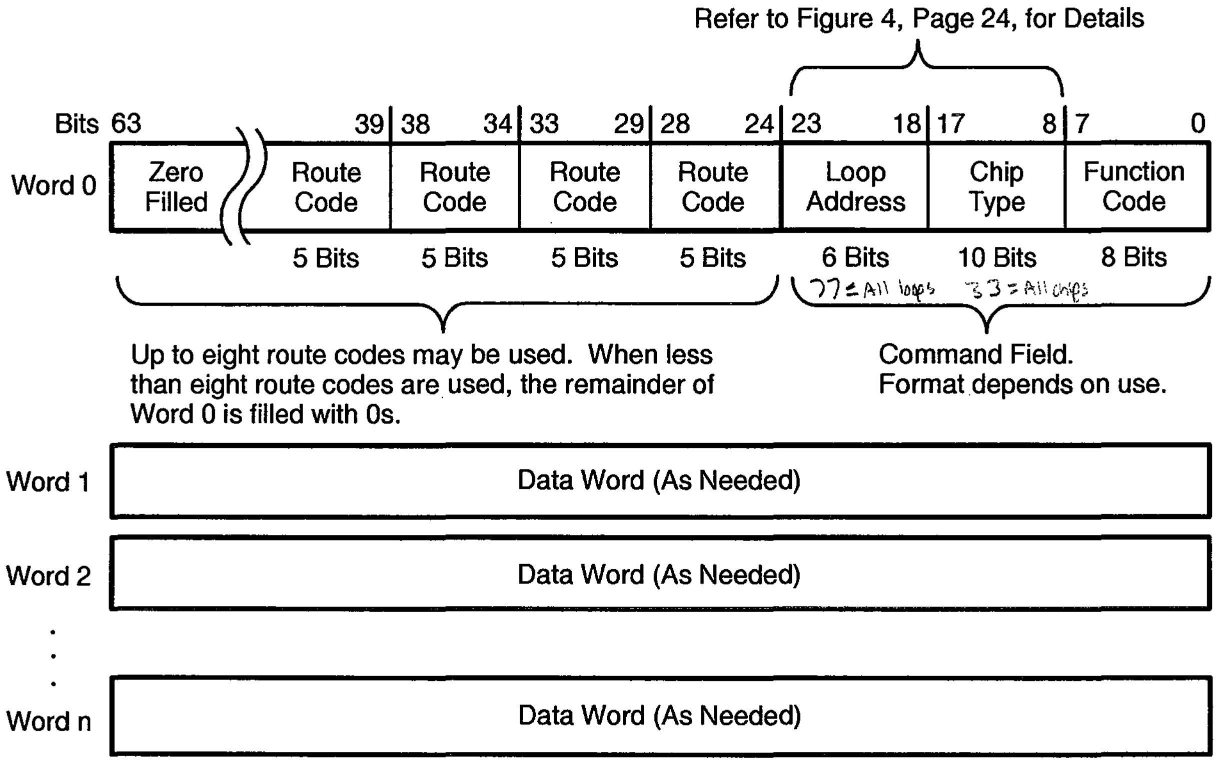
Sanity Code Generator

A sanity code generator on an I/O module creates all sanity codes. The sanity code generator is controlled through the maintenance channel. Only the sanity code generator can create sanity codes. CRAY T932 systems have two maintenance channels, each with its own sanity code generator. Note that logic for the sanity code generator resides on all I/O modules in the system because all I/O modules are the same.

A command is sent over the maintenance channel to initialize the sanity code generator on the I/O module. If the generator fails or produces invalid sanity code, the system will not be able to initialize a working configuration. In this situation, a different sanity code generator could be used to initialize a working configuration.

Maintenance Channel Commands

All the commands and data sent through the maintenance channel to the mainframe are in the form of a message or command word as shown in Figure 3.



Up to eight route codes may be used. When less than eight route codes are used, the remainder of Word 0 is filled with 0s.

Command Field. Format depends on use.

Figure 3. Maintenance Channel Command Word Format

¶
This is a serial data within the machine

Route Codes

A route code is used to select an address for a node in the system. A node is any module in the system that the maintenance channel uses. Each node has several in and out paths. Up to eight 5-bit route codes can be used in a message; each code defines the next node in the sanity tree.

The number of route codes in a message depends on the complete path address of the node that will receive the message. Route codes are right-aligned to bit 24 of word 0 of a maintenance channel message.

As a message reaches each node, the node removes the next route code to establish proper routing of the message. The remainder of the message is then passed to the next node. The system stores the routing sequence once a node path is established. This allows data following a command (in words 1 through n) to be routed without appending the route path to each data word.

NOTE: Route codes are always written in octal format.

Boundary Scan Module Route Codes

Table 1 lists route code addresses on the boundary scan module. Refer to *Boundary Scan Module*, publication number HMM-xxx-0, for detailed information.

Table 1. CPU Module Route Codes

Route Code	Destination on BS Module
20 - 27	Serial maintenance channel (module test only)
30	Loop controller
31 - 32	Undefined
33	Continuity line operation
34 - 36	Undefined
37	Continuation

I/O and CPU Module Route Codes

Table 2 and Table 3 list route code addresses for directly connected modules and addressable components within those modules. The destination CPU and shared module route codes are relative to the I/O module. For example, CP 0 from an I/O module could be physical CP 2, CP 12, CP 22, or CP 32. The logical CPU destinations (0 through 3) are configured locations and do not necessarily correspond to any physical slot position.

Table 2. I/O Module Route Codes

Route Code	Destination
20	Shared via CPU 2 (CPU 0, CRAY T94 system)
21	Shared via CPU 3 (CPU 1, CRAY T94 system)
22	Shared via CPU 4 (CPU 2, CRAY T94 system)
23	Shared via CPU 5 (CPU 3, CRAY T94 system)
30	Loop controller
34	Logic monitor
36 ¶	Sanity code generator
37	Continuation

¶ Available only through a directly connected maintenance channel.

Table 3. CP Module Route Codes

Route Code	Destination
20	Memory port 0
21	Memory port 1
22	Memory port 2
23	Memory port 3
24	Memory port 4
25	Memory port 5
26	Memory port 6
27	Memory port 7
30	Loop controller
33	I/O
34	Logic monitor
35	DMA port
37	Continuation

Shared Module Route Codes

Table 4 lists two levels of route codes for shared modules. First level route codes 20 and 21 select a group of CPUs, either CPUs 0 through 7, or 8 through 15. Then the second level route codes 20 through 27 select an individual CPU within the group of eight that were selected with the first level route code.

Table 4. Shared Module Route Codes

Route Code	Destination
First Level	
20	Local CPUs 0 through 7
21	Local CPUs 8 through 15
22	Other shared module
30	Loop controller
31	All attached CP modules
32	Reserved
33	Reserved
34	Logic monitor
37	Continuation
Second Level	
20	CPU 0
21	CPU 1
22	CPU 2 (CPU 0, CRAY T94 system)
23	CPU 3 (CPU 1, CRAY T94 system)
24	CPU 4 (CPU 2, CRAY T94 system)
25	CPU 5 (CPU 3, CRAY T94 system)
26	CPU 6
27	CPU 7
31	Reserved
37	Continuation

Route Code Example

If the maintenance channel in a CRAY T94 system is to access the loop controller in CPU3, the following route code is required: 20/20/25/30. The first 20 addresses the shared module. The second 20 is the first-level shared module route code which selects local CPUs 0 through 7. The 25 is the second-level shared module route code which selects CPU ~~5~~ 3. The 30 is the CPU's route code for the loop controller.

Another example route code for a CRAY T932 is 21/22/21/20/34. The first route code, 21, accesses the shared module via CPU 3. The first-level shared module route code, 22, routes the data to the other shared module. The next code, 21, selects local CPUs 8 through 15 on the other shared module. The 20 selects CPU 0; and the last code, 34, selects the logic monitor in the CPU.

Function Codes

The function code field is used to select a specific function on a chip (option). There can be up to 256 (400_8) function codes which are grouped as follows:

- Configuration Codes (0 – 77_8).
- Maintenance Mode Functions (100 – 177_8)
- Testpoint Selections (200 – 377_8)

NOTE: Refer to the *Triton Maintenance System Engineering Note*, number PRN-0957, for detailed descriptions of each function code; the appendix section includes tables that list testpoints for each option.

Configuration Codes

Configuration codes are forced to default values when sanity code is not available on the module. Master clear function codes do not clear configuration codes; refer to "Master Clear Levels" on page 45. Table 5 through Table 8 list the configuration codes for the CP, I/O, and shared modules.

NOTE: The System Configuration Environment (SCE) program issues configuration and maintenance mode function codes to create and manage the logical configuration of the mainframe.

Table 5. CP Module Configuration Function Codes 0 through 17

Code	Group	Function
0	Section profile	Select 4 section
1		Select 8 section
2	Section start number	Force bit 2, section address
7	Sectional control	Force inverted CPU
10	Subsection profile	Force bit 0, subsection address
11		Force bit 1, subsection address
12		Force bit 2, subsection address
13	Bank profile	Force bit 0, bank address
14		Force bit 1, bank address
15		Force bit 2, bank address
16		Force bit 3, bank address
17	Clear	Clear function codes 10-16

† These default states are forced when no sanity code is available or when a 77 reset code is sent.

Table 6. CP Module Configuration Function Codes 20 through 77

Code	Group	Function
20	Subsection select	Set subsection bit 0
21		Set subsection bit 1
22		Set subsection bit 2
23	Bank select	Set bank bit 0
24		Set bank bit 1
25		Set bank bit 2
26		Set bank bit 3
27	Clear	Clear function codes 20 – 26
30	Memory group profile	Force bit 0, group number
31		Force bit 1, group number
32	Memory group profile	Set group bit 0
33		Set group bit 1
37	Clear	Clear function codes 30 – 33
40	Force CPU to upper 256K	On
41 †		Off
42	Force I/O to upper 256K	On
43 †		Off
50 †	Enable shared access	On
51		Off
52 †	Originate I/O commands	On
53		Off
54 †	Pass-through I/O commands	On
55		Off
60 †	Memory master clear	On
61		Off
62 †	Memory access master clear	On
63		Off
64 †	CPU master clear	On
65		Off
70	Memory sanity code	On
71 †		Off
72	I/O sanity code	On
73 †		Off
76	CPU interrupt request	
77	Reset all configuration codes	

† These default states are forced when no sanity code is available or when a 77 reset code is sent.

Table 7. I/O Module Configuration Codes

Code	Group	Function
10	Support channel path	Set path 2^0
11		Set path 2^1
12 †		Clear path 2^0 , 2^1
16	Channel control	Set loopback, LA-LB pseudo options, (LOSP)
17 †		Clear loopback, LA-LB pseudo options, (LOSP)
20	Memory access	Set group mode
21 †		Clear group mode
22		Set memory group 2^0
23†		Clear memory group 2^0
24		Set memory group 2^1
25 †		Clear memory group 2^1
26	I/O access	Set I/O group 2^0; LA, LB, SU, and VH pseudo options (LOSP and VHISP)
27 †		Clear code 26
30		Set I/O group 2^1; LA, LB, SU, and VH pseudo options (LOSP and VHISP)
31 †		Clear code 30
32	Memory access	Set 256K mode
33 †		Clear 256K mode
34	Channel type	Set MISP mode; LA, LB, and SU pseudo options (LOSP)
35 †		Clear MISP mode; LA, LB, and SU pseudo options (LOSP)
36	Channel control	Channel on
37 †		Channel off
40	System access control	Rearbitrate master
60 †	Quadrant master clear	Quad 0 - on
61		Quad 0 - off
62 †		Quad 1 - on
63		Quad 1 - off
64 †		Quad 2 - on
65		Quad 2 - off
66 †		Quad 3 - on
67		Quad 3 - off
70	Logic monitor	Reset
72	Sanity code to shared	On
73 †		Off
74	Sanity code to CP	On
75 †		Off
77	Configuration code defaults	Reset

† These default states are forced when no sanity code is available or when a 77 reset code is sent.

Table 8. Shared Module Configuration Codes

Code	Group	Function
0	Automatic broadcast detach	On
1 †		Off
10	Support channel 60 – 61	Set bit 0
11		Set bit 1
12 †		Clear bits 0 and 1
14	Support channel 62 – 63	Set bit 0
15		Set bit 1
16 †		Clear bits 0 and 1
20	Support channel 64 – 65	Set bit 0
21		Set bit 1
22 †		Clear bits 0 and 1
24	Support channel 66 – 67	Set bit 0
25		Set bit 1
26 †		Clear bits 0 and 1
40†	Shared module position selection	Hardwired default - On
41		Backup default - On
42		Override default, force 0
43		Override default, force 1
44 †	Enable maintenance mode 100	On
45	Enable data to SR options	On
60 †	Shared module master clear	On
61		Off
66	Enable global CPU sanity code	On
67 †		Off
70	Enable sanity code to remote shared	On
71 †		Off
72	Enable alternate sanity code to remote shared	On
73 †		Off
74	Enable sanity code to CP module	On
75 †		Off
77	Configuration code defaults	Reset

† These default states are forced when no sanity code is available or when a 77 reset code is sent.

Maintenance Mode Function Codes

Maintenance mode functions diagnose or test areas of CP and I/O modules to ensure they are functioning properly. Table 9 lists function codes that are useful in testing areas that are difficult to test using other methods. Some of these areas are:

- Memory error correction SBCDBD
- Instruction stack
- Register parity

Maintenance mode codes are used only by diagnostic programs. Instruction codes can use these maintenance mode codes from a CPU if the CPU is enabled through the maintenance channel.

Maintenance functions can be part of the system instructions or can be run as part of the maintenance channel support system. The maintenance channel can send maintenance functions to the CPU at any time; however, this is not feasible when the system is in use. Forcing a dump of CPU instruction buffers during customer operation crashes the system because the buffer contents are dumped into memory location 1000_8, which is where the operating system resides. The CPU can be configured to reference only the upper 256K of memory before issuing maintenance functions to prevent a system interrupt.

Before a CPU can issue any maintenance mode functions, it must be put into maintenance mode using a 176 function code. This must be done when the CPU is not in a master clear state. After the 176 code is sent, the CPU may issue subsequent maintenance functions. Maintenance mode can be cleared by issuing a 177 (clear maintenance mode) code or by issuing a CPU Master Clear function to the processor. A 177 code also clears any maintenance functions from the system.

Table 9. Maintenance Mode Function Codes

Code	Unit	Function	Code	Unit	Function
100	Vector reg	Force parity	140	Cache	Force parity
101			141	Exchange	Exchange and halt
102			142		
103			143		
104	Mem. access	Reset priority pointers	144		
105	Mem. access	Force inverted CPU	145		
106			146		
107			147		
110	B/T reg	Force parity	150	I/O	Write path SECDED maint.
111			151	I/O	Write SECDED checkbyte
112			152	I/O	Read SECDED checkbyte
113			153	I/O	Disable SECDED correctable
114			154		
115			155		
116			156		
117			157		
120	Inst. stack	Force parity	160		
121	Inst. stack	Load buffer	161		
122	Inst. stack	Store buffer	162	Performance monitor	Enable maintenance
123	Inst. stack	Enter upper 32 bits for store	163		
124	Inst. stack	Sel. buffer Bit 0 for store	164	Memory in ports	4 clock resume delay
125	Inst. stack	Sel. buffer bit 1 for store	165	Memory in ports	16 clock resume delay
126	Inst. stack	Sel. buffer bit 2 for store	166	Memory in ports	63 clock resume delay
127			167		
130	Cache	Do not use!	170	Memory error correction	1771jk gives checkbyte (CP02). code 107 for CP01.
131	Cache	Page reg. load	171	Memory error correction	I/O data gives checkbyte
132	Cache	Page reg. compare	172	Memory error correction	Disable error correction (no log)
133	Cache		173	Memory error correction	For 176ijk instructions, store check byte in Vi instead of normal data.
134			174	Memory error correction	Checkbyte to I/O
135			175	Error logger	Disable error logger
136			176	Set CPU maint.	Maint. channel only.
137	Cache	Clear cache maint.	177	Clear maint. mode	

Loop Controllers

A loop controller serves as a central fan-out point on the module for all the possible 256 (400_8) function codes:

- Configuration parameters
- Maintenance mode functions
- Testpoint selections

Loop controller logic is contained in the following options:

- HG option on CP modules
- DM option on I/O modules
- SM0 and SM1 options on SR modules
- LM0 and LM1 options on NW modules

Memory modules do not contain loop controller logic. However, the network modules pass maintenance function codes to all memory modules that have established sanity codes.

A route code of 30 addresses the loop controllers in all CP, I/O, and SR modules. Testpoint readout and control data from various options are routed to the logic monitor (HM0 and HM1 options) on the respective CP, I/O, or SR module. A description of the logic monitor begins on page 31.

Loop Controller Operation Rules

The following list of rules explains the operation and function of loop controllers.

- A broadcast loop address (77_8) instructs the loop controller to select all loops (16 on CP and shared modules, 32 on I/O modules). Other addresses instruct the loop controller to select individual loops.

- All options on the loop receive the same control function, but only the option that matches the chip type field (Figure 4) accepts the control function. All other options ignore the control function.

- The chip (option) type field can specify an individual option or all options on the loop (all chips = 33_8). Each option on the loop has its own separate input path. That is, each loop has a separate output (pin) for each option on the loop.

- Two options of the same chip type are never on the same loop.
- Function codes are never defined twice; they are all unique.

Loop Controller Command Word Format

All function codes are sent to loop controllers in a maintenance channel command word, as described on page 12. Figure 4 shows bits 0 through 28 of the maintenance channel command word. Bits 8 through 28 route function codes to the loop controllers.

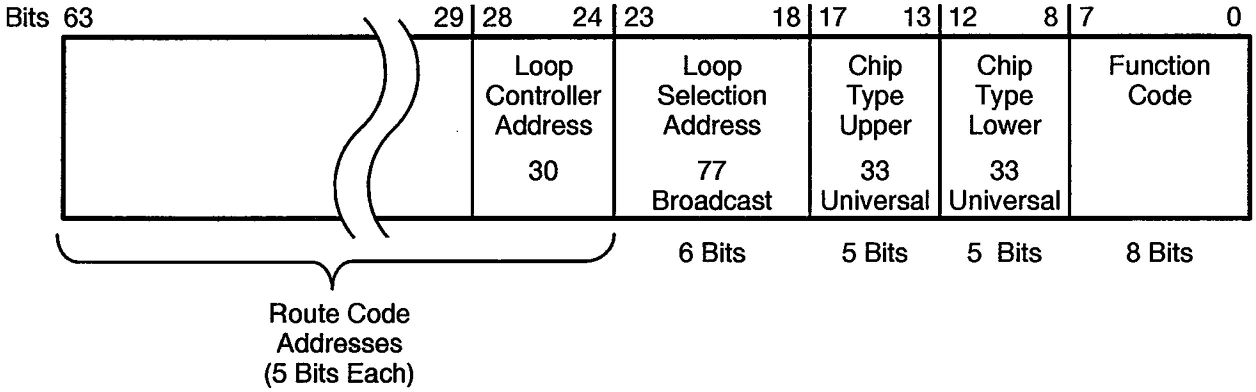


Figure 4. Loop Controller Command Word Format

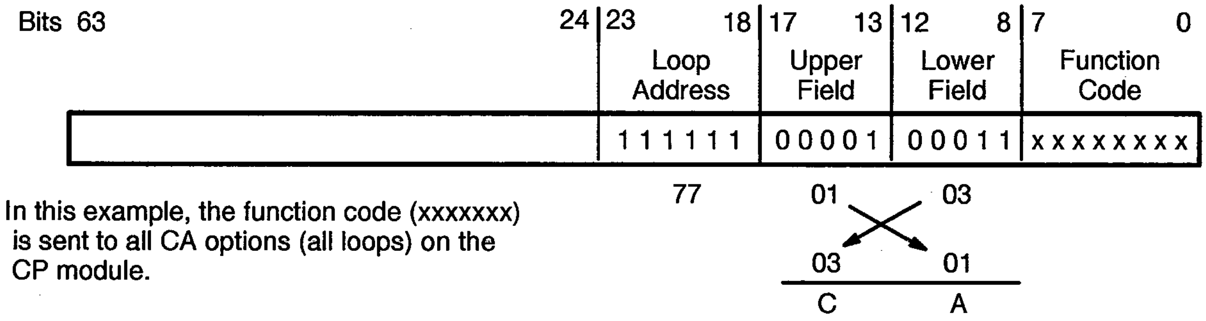
The first route codes in bits 29 through 63 of the command word direct the command to the desired module. The last route code in the command word is the loop controller address, which is 30 for all modules. This code ensures that the command word gets to the loop controller.

The loop selection address defines which loop or set of loops leaving the loop controller will distribute the function code. Only values of 0 through 37_8 for I/O and 77_8 are valid in the loop selection address; other values may give unpredictable results.

Chip Type Fields

The 5 bits in each of the upper and lower chip type fields define the letters of the chip or option. The letter A is encoded as 1, B as 2, and so on; Z is encoded as 32_8. Note that the upper and lower chip type bits are swapped.

For example, as shown in Figure 5, the CA option is encoded as 01,03 in bits 17 through 8. If the function code needs to be sent out the CA01 option (loop 1) on the CP modules, bits 23 through 7 would be 01,01,03. To broadcast to all CA options, the bits would be 77,01,03 as shown in Figure 5. The rule that chip (option) types with the same letter designators cannot reside on the same loop is set because of this routing scheme used in bits 23 through 8.



In this example, the function code (xxxxxxx) is sent to all CA options (all loops) on the CP module.

Figure 5. Example Chip Type Address

Chip type addresses for options that do not reside on a module are ignored. For example a chip address of 02,03 (CB option) sent to the shared module is ignored because there are no CB options on the shared modules.

CP Module Loop Controller

The HG option on each CP module contains the loop controller logic as shown in Figure 6. Each of the 16 HG loop outputs (OHA-OHP) goes to a separate CH option. Each CH option performs a 1 to 10 fanout. These fanouts enable the loop controller to forward maintenance function codes to a maximum of 160 option inputs on the CP module.

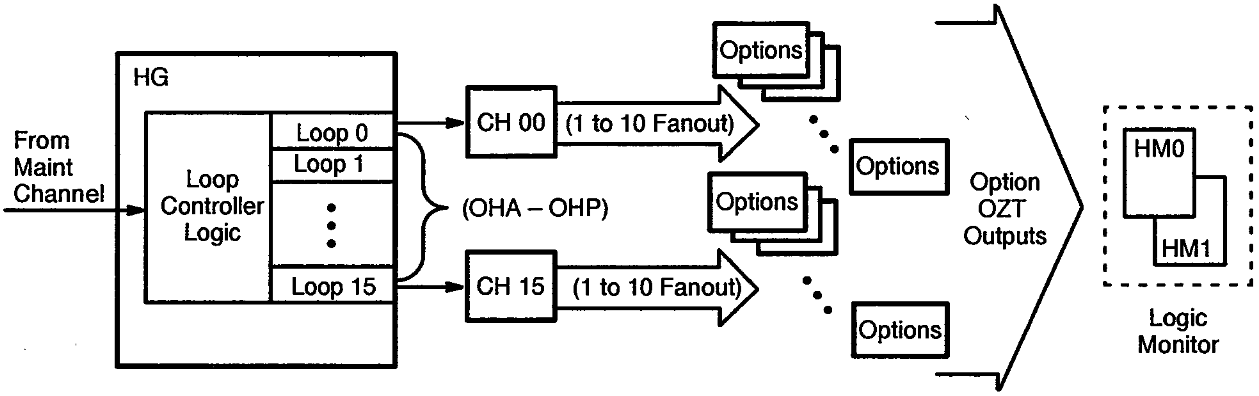


Figure 6. CP Module Loop Controller

Table 10. CP Module Loop Option Assignments

Loop	Options									
Loop 0 (CH00)	VR 00	VM 00	CI 07	CJ 07		CF 00	VF 02	CA 00	CB 00	AR 00
Loop 1 (CH01)	VR 01	VM 01	CI 03	CJ 03		CF 01	VF 03	CA 01	CB 01	AS 00
Loop 2 (CH02)	VR 02	VM 02	CI 00	CJ 00	HB 00		VF 00	CF 02	AU 00	HC 00
Loop 3 (CH03)	VR 03	VM 03	CI 04	CJ 04	HF 00		VF 01	CF 03	AU 01	
Loop 4 (CH04)	VR 04	VM 04	CI 06	CJ 06	VA 00	CG 00	AT 00	CF 04		
Loop 5 (CH05)	VR 05	VM 05	CI 02	CJ 02	VA 01	CG 01	AT 01	CF 05		
Loop 6 (CH06)	VR 06	VM 06	CI 01	CJ 01	HA 01	AS 01	CC 00			
Loop 7 (CH07)	VR 07	VM 07	CI 05	CJ 05	HA 03	AS 02				
Loop 8 (CH08)	VR 08	VM 08	IC 02		CD 00					
Loop 9 (CH09)	VR 09	VM 09	IC 03	SS 00						
Loop 10 (CH10)	VR 10	VM 10	HA 00	BT 00	JA 00					
Loop 11 (CH11)	VR 11	VM 11	HA 02	BT 01	JA 01					
Loop 12 (CH12)	VR 12	VM 12		IC 01	CD 01					
Loop 13 (CH13)	VR 13	VM 13		IC 01						
Loop 14 (CH14)	VR 14	VM 14	HG 00	HD 00						
Loop 15 (CH15)	VR 15	VM 15		HD 01						
CH option outputs (10)	OTO	OTP	OTQ	OTR	OTS	OTT	OTU	OTV	OTW	OTX

I/O Module Loop Controller

Each I/O module has 24 channels: 4 VHISP, 8 HISP, 8 LOSP, and 4 LOSPX channels. Each of these 24 channels can have 10 to 16 separate configurations, which exceeds the limit of 64 configuration codes. Loop controller loops 0 through 3 correspond to one quadrant of an I/O module, except for the DR options. Each quadrant contains 1 VHISP, 2 HISP, 2 LOSP, and routing for the LOSPX support and PINT/MCUI channels. All I/O data in a quadrant is routed through a connected CP module. Figure 7 shows option loop assignments; Table 11 lists these assignments.

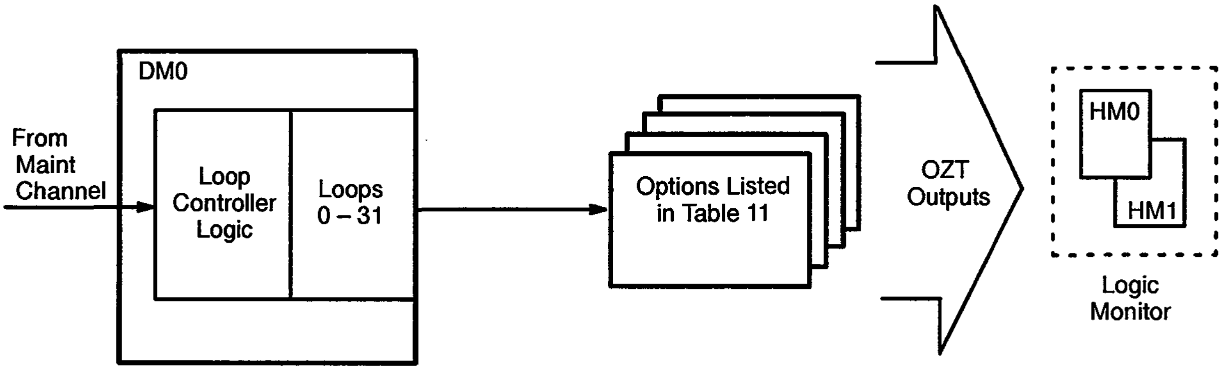


Figure 7. I/O Module Loop Controller

Table 11. I/O Module Loop Option Assignments

Loop	Options					
Loop 0	DR 00	DA 00	DB 00	DC 00	DD 00	DE 00
Loop 1	DR 01	DA 01	DB 01	DC 01	DD 01	DE 01
Loop 2	DR 02	DA 02	DB 02	DC 02	DD 02	DE 02
Loop 3	DR 03	DA 03	DB 03	DC 03	DD 03	DE 03
Loop 4	DR 04					
Loop 5	DR 05					
Loop 6	DR 06					
Loop 7	DR 07					
Loop *n*	DR *n*					
Loop 31	DR 31					

Shared Module Loop Controller

Loop controller logic for the shared modules is contained on the SM00 and SM01 options as shown in Figure 8. Table 12 lists options on each loop.

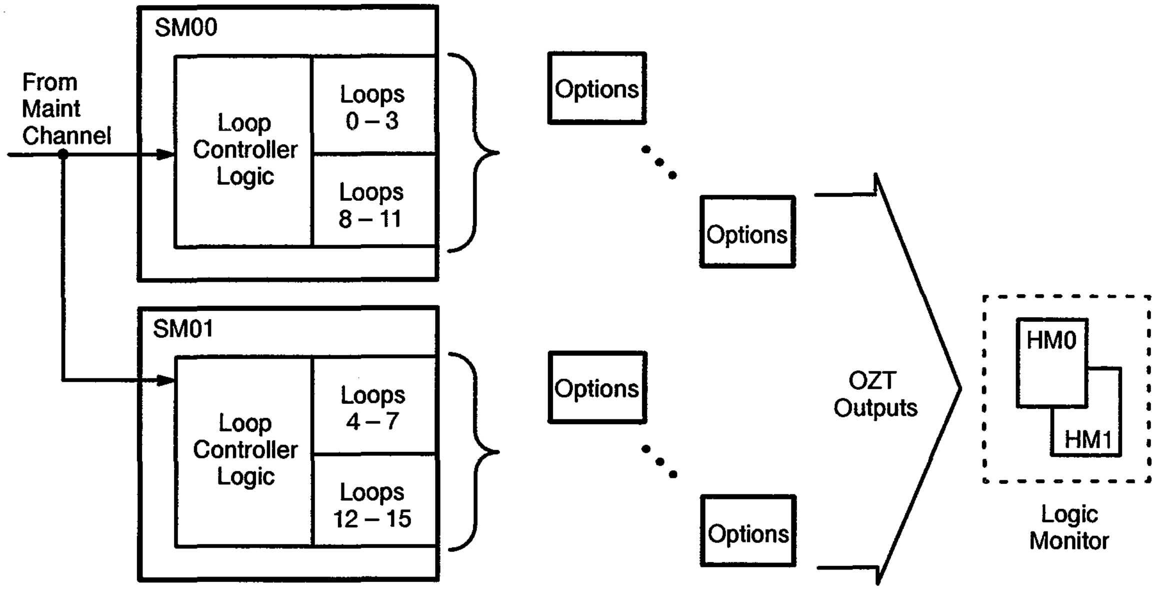


Figure 8. Shared Module Loop Controllers

Table 12. Shared Module Loop Option Assignments

Loop	Options					
Loop 0	SA 00	SB 00	SC 00	SD 00	SR 00 †	SM 00
Loop 1	SA 01	SB 01	SC 01	SD 01	SR 01 †	
Loop 2	SA 02	SB 02	SC 02	SD 02	SR 02 †	
Loop 3	SA 03	SB 03	SC 03	SD 03	SR 03 †	
Loop 4	SA 04	SB 04		SD 04	SR 04 †	
Loop 5	SA 05	SB 05		SD 05		
Loop 6	SA 06	SB 06		SD 06		
Loop 7	SA 07	SB 07		SD 07		
Loop 8	SA 08	SB 08	SC 04	SD 08	SR 05 †	SM 01
Loop 9	SA 09	SB 09	SC 05	SD 09	SR 06 †	
Loop 10	SA 10	SB 10	SC 06	SD 10	SR 07 †	
Loop 11	SA 11	SB 11	SC 07	SD 11	SR 08 †	
Loop 12	SA 12	SB 12		SD 12		
Loop 13	SA 13	SB 13		SD 13		
Loop 14	SA 14	SB 14		SD 14		
Loop 15	SA 15	SB 15		SD 15		

† The SR options can be sent data only if configuration code 45 is active. Current options cannot receive any data.

Network Module Loop Controller

The loop controller logic for the network modules is contained on the LM1 and LM2 options as shown in Figure 9. Table 13 lists the options on each loop.

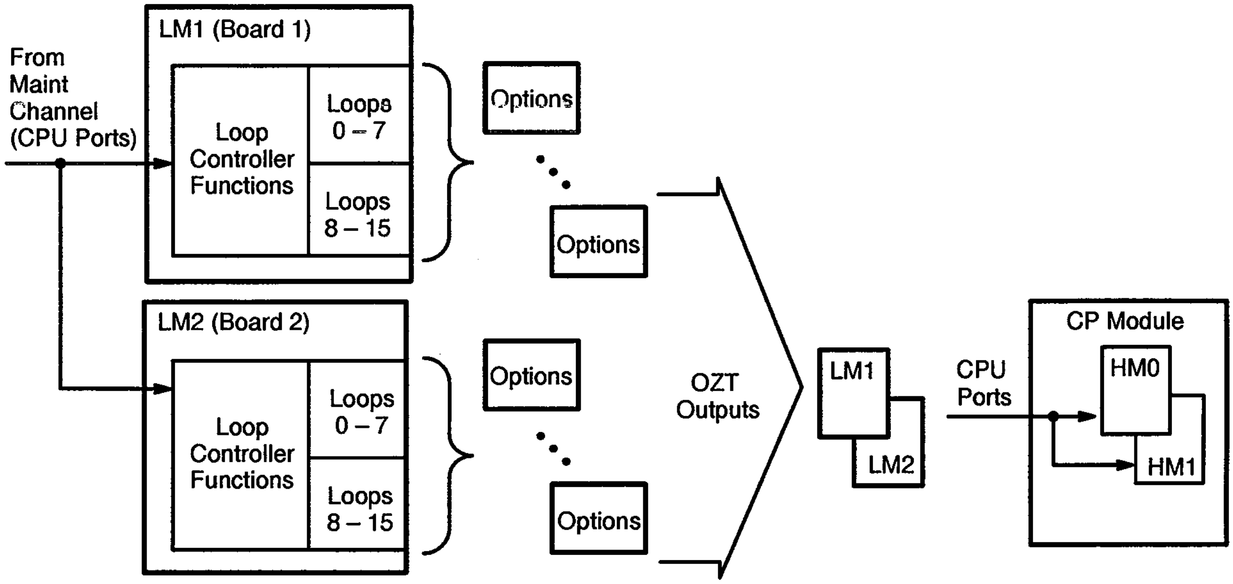


Figure 9. Network Module Loop Controllers

Table 13. Network Module Loop Option Assignments

Loop	Options		
Loop 0	LA 01	LB 01	LC 01
Loop 1	LA 02	LB 02	LC 02
Loop 2	LA 03	LB 03	LC 03
Loop 3	LA 04	LB 04	LC 04
Loop 4	LA 05	LB 05	LC 05
Loop 5	LA 06	LB 06	LC 06
Loop 6	LA 07	LB 07	LC 07
Loop 7	LA 08	LB 08	LC08
Loop 8	LA 09	LB 09	LC 09
Loop 9	LA 10	LB 10	LC 10
Loop 10	LA 11	LB 11	LC 11
Loop 11	LA 12	LB 12	LC 12
Loop 12	LB 13		
Loop 13	LB 14		
Loop 14			
Loop 15			

Boundary Scan Loop Controller

A boundary scan module can be directly connected to a CP module in module checkout situations were there is no I/O module. (This configuration is used only in STCO.) Logic on the boundary scan module provides maintenance channel functions in this configuration.

Table 14. Boundary Scan Loop Controller Function Codes

Code	Destination
50	Primary sanity code - off
51	Primary sanity code - on
52	Secondary sanity code - off
53	Secondary sanity code - on
60	Error logger - off
61	Error logger - on

The primary and secondary sanity code generators enable different sanity code paths, depending on the checkout environment. STCO systems that have a CP and a shared module use the primary sanity code generator. STCO systems that do not have a shared module (only checking a CP module), use the secondary sanity code generator.

Function codes 50 through 53 use a broadcast loop address (77) and universal chip type address (33). A route code of 30 is used for all boundary scan loop controller functions.

Before sending any maintenance channel function that will return data, the error logger must be disabled with a 61 code. This is necessary because the system cannot distinguish a possible error log data word from a requested maintenance channel data word. When the error log is disabled, a channel disconnect is returned; normal error log data does not return disconnects.

Logic Monitor

The logic monitor is used to select and store test-point data and control information from various logic chips to indicate the state of operation of various modules. The logic monitor can also be used to store P-register or instruction parcel data, take snapshots of selected test points or bring selected test-point data back to an external scoping point. The logic monitor can also cause a CPU breakpoint.

The HM0 and HM1 on each CP, I/O, and shared module contain the logic monitor. The network and memory modules do not have HM options, but they do provide logic monitor functions as described later. Each HM option can store test-point data from 64 logic chips. Logic monitors are controlled and programmed through the maintenance channel from the MWS.

The logic monitor can also be used to monitor the CPU(s) as instructions are issuing. The logic monitor can start and stop recording on specific events that occur while instructions are issuing. This allows you to monitor CPU(s) as instructions issue without affecting the CPUs. You can then use the logic monitor to compare actual test-point data with expected data on any of the modules in the system.

Figure 10 illustrates a CP module logic monitor and a loop controller that send function codes to options to select desired test points. Commands selecting the test points to be read are sent to the loop controller. The loop controller sends them to the options. Test point readout data is read by the logic monitor and sent out the maintenance channel return path to the MWS.

Logic Monitor Interface

A stand-alone X Window System based application called the logic monitor environment (LME) provides an interface to CRAY T90 series system logic monitors. You can also start LME from an MME menu. The *Logic Monitor Environment*, publication number HDM-xxx-0, describes how to use LME to operate and control the logic monitors. The information in this *Maintenance Channel* document describes the logic monitor and how it operates and records test-point data.

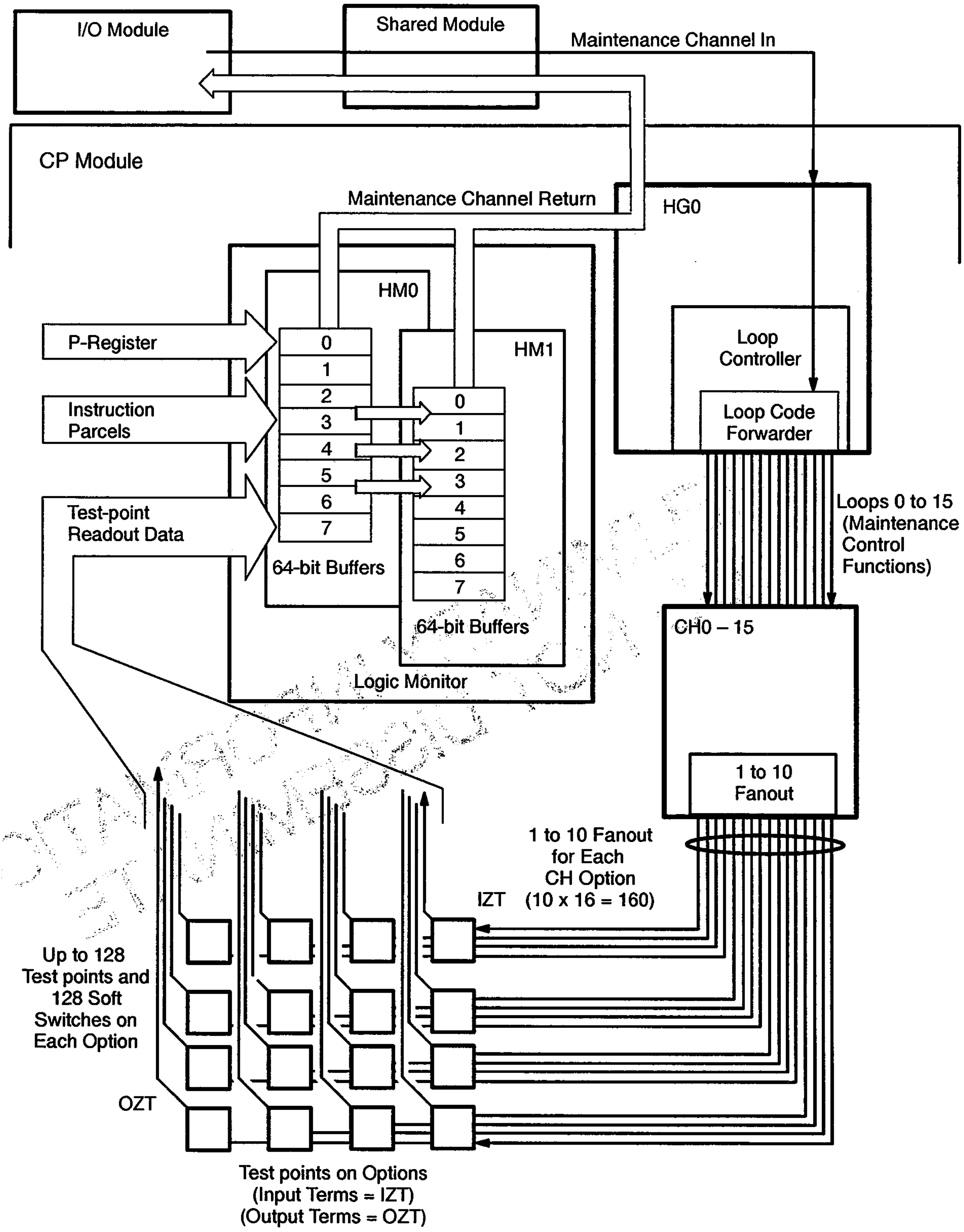

I/O Module
Shared Module
Maintenance Channel In
CP Module
Maintenance Channel Return
HG0
HM0
HM1
P-Register
Instruction Parcels
Test-point Readout Data
0
1
2
3
4
5
6
7
64-bit Buffers
64-bit Buffers
Logic Monitor
Loop Controller
Loop Code Forwarder
Loops 0 to 15 (Maintenance Control Functions)
CH0 – 15
1 to 10 Fanout
1 to 10 Fanout for Each CH Option (10 x 16 = 160)
IZT
Up to 128 Test points and 128 Soft Switches on Each Option
OZT
Test points on Options (Input Terms = IZT) (Output Terms = OZT)

Figure 10. Logic Monitor Control and Data Flow on CP Module

Test Points

Test points are access points into options. Data from each test point is used to verify input or output signals from various options or chips.

NOTE: The logic monitor does not change system configurations or soft switch selections or settings; loop controllers control these functions.

Test point selections within options are performed through loop controllers. On each CP module, 98 options have test points that can be used in logic monitor functions. Each option contains up to 128 test points and 128 soft switches. The recording of test point data is performed on a clock period basis.

Most of the options in the system have a built-in test-point multiplexer that can selectively monitor up to 128 different signals. The option designer determines the specific signals that can be monitored. Each option has its chip type and the signals that can be monitored by test points coded inside the chip.

Testpoints are selected over the maintenance channel using route codes to target a specific chip and function. These test points are then routed to the HM0 and HM1 options on the module where the test-point data is captured or compared against expected results. The HM options control the functions of the logic monitor.

You may use a test point either to signal the start of an event or as a trigger point to end recording, but not both with the same route code command. You can examine the condition of as many as 8, 16, or 32 test-point numbers after a targeted event has occurred.

Logic Monitor Testpoint Pin Assignments

Table 15 through Table 22 list logic monitor input pin assignments for test points from options on each type of module.

NOTE: Each option on an I/O module is connected to two input pins on the logic monitor; this is possible because of the small number of options on I/O modules.

Logic Monitor on Network and Memory Modules

The network and memory modules do not have HM options with 64-bit buffers to store test-point data. However, logic monitor functions do exist for both types of modules; you can read test-point data from options on both modules. Function codes route test-point data from network modules to a connected CP module and record the data in the buffers of the HM options.

The following function codes are available for capturing test-point data from options on the network modules:

- Code 05 - Select Test Points. This code selects an option for test-point data and routes the data to a CP port. If the maintenance return path is enabled, the 05 code also routes test-point data to its port.

- Code 07 - Snapshot Test Points. This code records all test-point data from the network module or a data bit from each connected memory module. This code returns a data word containing the test-point data. If memory module data is requested, the 8 data bits are right-aligned, and the word is filled with zeroes; the bit for memory module 0 is in bit position 0, module two in bit position 2, and so on.

Test-point Data Output from the Logic Monitor

Test-point data output from the logic monitor (HM0 and HM1) is always sent through the maintenance channel in full 64-bit words. Each HM chip sends only 16 bits of data at a time. A 64-bit word contains 32 valid bits from two HM options.

The number of bits of data output in each halfword varies according to the recording mode used (8 x 1024, 16 x 512, and 32 x 256), as shown in Figure 11.

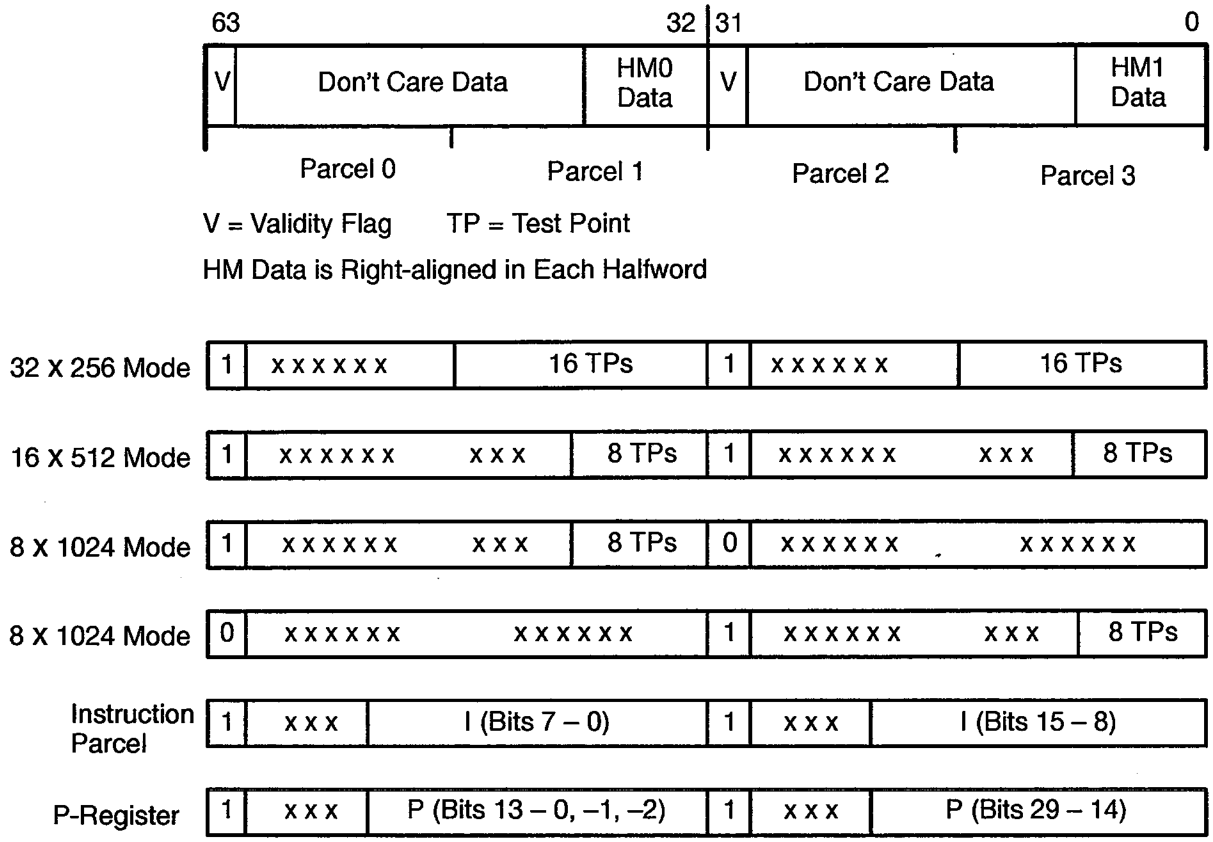


Figure 11. Logic Monitor Data Word Output Formats

Logic Monitor Modes for Recording Test Point Data

The logic monitor has several modes of recording test-point data; the configuration of the eight register buffers determines these modes. Each buffer is 64 by 8 for a total of 512 storage locations per HM or a total of 1024 per CPU. Configuring buffers depends on the number of test points captured and the number of clock periods to record. The more test points you want to capture, the fewer the number of clock periods available to capture the data.

The two HM options work in parallel to capture the following number of test points (TPs):

- Mode 1 = 8 TPs for 1,024 CPs (May be noted as 8 x 1024 mode)
  8 groups x 32 bits
- Mode 2 = 16 TPs for 512 CPs (May be noted as 16 x 512 mode)
  16 groups x 16 bits
- Mode 3 = 32 TPs for 256 CPs (May be noted as 32 x 256 mode)
  32 groups x 8 bits

NOTE: In Modes 2 and 3, each HM option records data separately; however, in mode 1 each HM option records data for 512 clock periods.

There are eight test-point **groups** on each CPU, four test-point groups per HM option. Up to 32 test points can be selected (mode 3). Table 15 and Table 16 list all the possible test points, the Boolean terms, and the option types.

Table 15. HM0 Test-point Assignments on CP Modules

Input Bit	Primary		Secondary		Input Bit	Primary		Secondary	
	Term	Chip Type	Term	Chip Type		Term	Chip Type	Term	Chip Type
0	IAA	VR00	IAH		32	IEA	VR04	IAD	CK00
1	IAB	VM00	IBG	AS00	33	IEB	VM04	IBC	CI01
2	IAC	CI00	ICF	VF01	34	IEC	CI04	ICB	VM02
3	IAD	CK00	IDE	HD00	35	IED	CA00	IDA	VR03
4	IAE	HA00	IED	CA00	36	IEE	HF00	IEH	
5	IAF	VA00	IFC	CI05	37	IEF	IC01	IFG	
6	IAG	VR00	IGB	VM06	38	IEG	AS01	IGF	
7	IAH		IHA	VR07	39	IEH		IHE	BT00
8	IBA	VR01	IAA	VR00	40	IFA	VR05	IAE	HA00
9	IBB	VM01	IBH		41	IFB	VM05	IBD	CK01
10	IBC	CI01	ICG	AT00	42	IFC	CI05	ICC	CI02
11	IBD	CK01	IDF	IC00	43	IFD	CB00	IDB	VM03
12	IBE	HA01	IEE	HF00	44	IFE	HB00	IEA	VR04
13	IBF	VF00	IFD	CB00	45	IFF		IFH	
14	IBG	AS00	IGC	CI06	46	IFG		IGG	
15	IBH		IHB	VM07	47	IFH		IHF	
16	ICA	VR02	IAB	VM00	48	IGA	VR06	IAF	VA00
17	ICB	VM02	IBA	VR01	49	IGB	VM06	IBE	HA01
18	ICC	CI02	ICH		50	IGC	CI06	ICD	CF00
19	ICD	CF02	IDG	AU00	51	IGD	CG00	IDC	CI03
20	ICE		IEF	IC01	52	IGE	JA00	IEB	VM04
21	ICF	VF01	IFE	HB00	53	IGF		IFA	VR05
22	ICG	AT00	IGD	CG00	54	IGG		IGH	
23	ICH		IHC	CI07	55	IGH		IHG	
24	IDA	VR03	IAC	CI00	56	IHA	VR07	IAG	AR00
25	IDB	VM03	IBB	VM01	57	IHB	VM07	IBF	VF00
26	IDC	CI03	ICA	VR02	58	IHC	CI07	ICE	
27	IDD	CF01	IDH		59	IHD	CC00	IDD	CF01
28	IDE	HD00	IEG	AS01	60	IHE	BT00	IEC	CI04
29	IDF	IC00	IFF		61	IHF		IFB	VM05
30	IDG	AU00	IGE	JA00	62	IHG		IGA	VR06
31	IDH		IHD	CC00	63	IHH		IHH	

Table 16. HM1 Test-point Assignments on CP Modules

Input Bit	Primary		Secondary		Input Bit	Primary		Secondary	
	Term	Chip Type	Term	Chip Type		Term	Chip Type	Term	Chip Type
0	IAA	VR08	IAH		32	IEA	VR12	IAD	CK02
1	IAB	VM08	IBG	AS02	33	IEB	VM12	IBC	CJ01
2	IAC	CJ00	ICF	VF03	34	IEC	CJ04	ICB	VM10
3	IAD	CK02	IDE	HD01	35	IED	CA01	IDA	VR11
4	IAE	HA02	IED	CA01	36	IEE	HG00	IEH	
5	IAF	VA00	IFC	CJ05	37	IEF	IC03	IFG	
6	IAG		IGB	VM14	38	IEG		IGF	
7	IAH		IHA	VR07	39	IEH		IHE	BT01
8	IBA	VR09	IAA	VR15	40	IFA	VR13	IAE	HA02
9	IBB	VM09	IBH		41	IFB	VM13	IBD	CK03
10	IBC	CJ01	ICG	AT01	42	IFC	CJ05	ICC	CJ01
11	IBD	CK03	IDF	IC02	43	IFD	CB01	IDB	VM11
12	IBE	HA03	IEE	HG00	44	IFE		IEA	VR12
13	IBF	VF02	IFD	CB01	45	IFF		IFH	
14	IBG	AS02	IGC	CJ06	46	IFG		IGG	
15	IBH		IHB	VM15	47	IFH		IHF	
16	ICA	VR10	IAB	VM08	48	IGA	VR14	IAF	VA01
17	ICB	VM10	IBA	VR09	49	IGB	VM14	IBE	HA03
18	ICC	CJ02	ICH		50	IGC	CJ06	ICD	CF02
19	ICD	CF02	IDG	AU01	51	IGD	CG01	IDC	CJ03
20	ICE	HC00	IEF	IC03	52	IGE	JA01	IEB	VM12
21	ICF	VF03	IFE		53	IGF		IFA	VR13
22	ICG	AT01	IGD	CG01	54	IGG		IGH	
23	ICH		IHC	CJ07	55	IGH		IHG	
24	IDA	VR11	IAC	CJ00	56	IHA	VR15	IAG	
25	IDB	VM11	IBB	VM01	57	IHB	VM15	IBF	VF02
26	IDC	CJ03	ICA	VR10	58	IHC	CJ07	ICE	HC00
27	IDD	CF03	IDH		59	IHD	SS00	IDD	CF03
28	IDE	HD01	IEG		60	IHE	BT01	IEC	CJ04
29	IDF	IC02	IFF		61	IHF		IFB	VM13
30	IDG	AU01	IGE	JA01	62	IHG		IGA	VR14
31	IDH		IHD	SS00	63	IHH		IHH	

Table 17. HM0 Test-point Assignments on Shared Modules

Input Bit	Primary		Secondary		Input Bit	Primary		Secondary	
	Term	Chip Type	Term	Chip Type		Term	Chip Type	Term	Chip Type
0	IAA	SA00	IAH		32	IEA	SA08	IAD	SD00
1	IAB	SB00	IBG		33	IEB	SB08	IBC	SC01
2	IAC	SC00	ICF		34	IEC		ICB	SB02
3	IAD	SD00	IDE	SR03	35	IED	SD04	IDA	SA03
4	IAE	SR00	IED	SD04	36	IEE	SR04	IEH	
5	IAF	SM00	IFC		37	IEF		IFG	
6	IAG		IGB	SB10	38	IEG		IGF	
7	IAH		IHA	SA11	39	IEH		IHE	
8	IBA	SA01	IAA	SA00	40	IFA	SA09	IAE	SR00
9	IBB	SB01	IBH		41	IFB	SB09	IBD	SD01
10	IBC	SC01	ICG		42	IFC		ICC	SC02
11	IBD	SD01	IDF		43	IFD	SD05	IDB	SB03
12	IBE	SR01	IEE	SR04	44	IFE		IEA	SA08
13	IBF		IFD	SD05	45	IFF		IFH	
14	IBG		IGC		46	IFG		IGG	
15	IBH		IHB	SB11	47	IFH		IHF	
16	ICA	SA02	IAB	SB00	48	IGA	SA10	IAF	SM00
17	ICB	SB02	IBA	SA01	49	IGB	SB10	IBE	SR01
18	ICC	SC02	ICH		50	IGC		ICD	SD02
19	ICD	SD02	IDG		51	IGD	SD06	IDC	SC03
20	ICE	SR02	IEF		52	IGE		IEB	SB08
21	ICF		IFE		53	IGF		IFA	SA09
22	ICG		IGD	SD06	54	IGG		IGH	
23	ICH		IHC		55	IGH		IHG	
24	IDA	SA03	IAC	SC00	56	IHA	SA11	IAG	
25	IDB	SB03	IBB	SB01	57	IHB	SB11	IBF	
26	IDC	SC03	ICA	SA02	58	IHC		ICE	SR02
27	IDD	SD03	IDH		59	IHD	SD07	IDD	SD03
28	IDE	SR03	IEG		60	IHE		IEC	
29	IDF		IFF		61	IHF		IFB	SB09
30	IDG		IGE		62	IHG		IGA	SA10
31	IDH		IHD	SD07	63	IHH		IHH	

Table 18. HM1 Test-point Assignments on Shared Modules

Input Bit	Primary		Secondary		Input Bit	Primary		Secondary	
	Term	Chip Type	Term	Chip Type		Term	Chip Type	Term	Chip Type
0	IAA	SA04	IAH		32	IEA	SA12	IAD	SD08
1	IAB	SB04	IBG		33	IEB	SB12	IBC	SC05
2	IAC	SC04	ICF		34	IEC		ICB	SB06
3	IAD	SD08	IDE	SR08	35	IED	SD12	IDA	SA07
4	IAE	SR05	IED	SD12	36	IEE		IEH	
5	IAF		IFC		37	IEF		IFG	
6	IAG		IGB	SB14	38	IEG		IGF	
7	IAH	SA05	IHA	SA15	39	IEH		IHE	
8	IBA	SB05	IAA	SA04	40	IFA	SA13	IAE	SR05
9	IBB	SC05	IBH		41	IFB	SB13	IBD	SD09
10	IBC	SD09	ICG		42	IFC		ICC	SC06
11	IBD	SR06	IDF		43	IFD	SD13	IDB	SB07
12	IBE		IEE		44	IFE		IEA	SA12
13	IBF		IFD	SD13	45	IFF		IFH	
14	IBG		IGC		46	IFG		IGG	
15	IBH		IHB	SB15	47	IFH		IHF	
16	ICA	SA06	IAB	SB04	48	IGA	SB14	IAF	SM01
17	ICB	SB06	IBA	SA05	49	IGB	SB14	IBE	SR06
18	ICC	SD10	ICH		50	IGC		ICD	SD10
19	ICD	SR07	IDG		51	IGD	SD14	IDC	SC07
20	ICE		IEF		52	IGE		IEB	SB12
21	ICF		IFE		53	IGF		IFA	SA13
22	ICG		IGD	SD14	54	IGG		IGH	
23	ICH		IHC		55	IGH		IHG	
24	IDA	SA07	IAC	SC04	56	IHA	SA15	IAG	
25	IDB	SB07	IBB	SB05	57	IHB	SB15	IBF	
26	IDC	SC07	ICA	SA06	58	IHC		ICE	SR07
27	IDD	SD11	IDH		59	IHD	SD15	IDD	SD11
28	IDE	SR08	IEG		60	IHE		IEC	
29	IDF		IFF		61	IHF		IFB	SB13
30	IDG		IGE		62	IHG		IGA	SA14
31	IDH		IHD	SD15	63	IHH		IHH	

Table 19. HM0 Test-point Assignments on I/O Modules

Input Bit	Primary		Secondary		Input Bit	Primary		Secondary	
	Term	Chip Type	Term	Chip Type		Term	Chip Type	Term	Chip Type
0	IAA	DR00	IAH	DR19	32	IEA	DA00	IAD	DR03
1	IAB	DR01	IBG	DR22	33	IEB	DA01	IBC	DR06
2	IAC	DR02	ICF	DM00	34	IEC	DE00	ICB	DB00
3	IAD	DR03	IDE	DE01	35	IED	DE01	IDA	DA01
4	IAE	DR16	IED	DE01	36	IEE	DR00	IEH	DR20
5	IAF	DR17	IFC	DM00	37	IEF	DR04	IFG	DR17
6	IAG	DR18	IGB	DC01	38	IEG	DR16	IGF	DR06
7	IAH	DR19	IHA	DD00	39	IEH	DR20	IHE	DR03
8	IBA	DR04	IAA	DR00	40	IFA	DB00	IAE	DR16
9	IBB	DR05	IBH	DR23	41	IFB	DB01	IBD	DR07
10	IBC	DR06	ICG		42	IFC	DM00	ICC	DC00
11	IBD	DR07	IDF		43	IFD		IDB	DB01
12	IBE	DR20	IEE	DR00	44	IFE	DR01	IEA	DA00
13	IBF	DR21	IFD		45	IFF	DR05	IFH	DR21
14	IBG	DR22	IGC		46	IFG	DR17	IGG	DR18
15	IBH	DR23	IHB	DD01	47	IFH	DR21	IHF	DR07
16	ICA	DA00	IAB	DR01	48	IGA	DC00	IAF	DR17
17	ICB	DB00	IBA	DR04	49	IGB	DC01	IBE	DR20
18	ICC	DC00	ICH		50	IGC		ICD	DD00
19	ICD	DD00	IDG		51	IGD		IDC	DC01
20	ICE	DE00	IEF	DR04	52	IGE	DR02	IEB	DA01
21	ICF	DM00	IFE	DR01	53	IGF	DR06	IFA	DB00
22	ICG		IGD		54	IGG	DR18	IGH	DR22
23	ICH		IHC		55	IGH	DR22	IHG	DR19
24	IDA	DA01	IAC	DR02	56	IHA	DD00	IAG	DR18
25	IDB	DB01	IBB	DR05	57	IHB	DD01	IBF	DR21
26	IDC	DC01	ICA	DA00	58	IHC		ICE	DE00
27	IDD	DD01	IDH		59	IHD		IDD	DD01
28	IDE	DE01	IEG	DR16	60	IHE	DR03	IEC	DE00
29	IDF		IFF	DR05	61	IHF	DR07	IFB	DB01
30	IDG		IGE	DR02	62	IHG	DR19	IGA	DC00
31	IDH		IHD		63	IHH	DR23	IHH	DR23

Table 20. HM1 Test-point Assignments on I/O Modules

Input Bit	Primary		Secondary		Input Bit	Primary		Secondary	
	Term	Chip Type	Term	Chip Type		Term	Chip Type	Term	Chip Type
0	IAA	DR08	IAH	DR27	32	IEA	DA02	IAD	DR11
1	IAB	DR09	IBG	DR30	33	IEB	DA03	IBC	DR14
2	IAC	DR10	ICF		34	IEC	DE02	ICB	DB02
3	IAD	DR11	IDE		35	IED	DE03	IDA	DA03
4	IAE	DR24	IED	DE03	36	IEE	DR08	IEH	DR28
5	IAF	DR25	IFC		37	IEF	DR12	IFG	DR25
6	IAG	DR26	IGB	DC03	38	IEG	DR24	IGF	DR14
7	IAH	DR27	IHA	DD02	39	IEH	DR28	IHE	DR11
8	IBA	DR12	IAA	DR08	40	IFA	DB02	IAE	DR24
9	IBB	DR13	IBH	DR31	41	IFB	DB03	IBD	DR15
10	IBC	DR14	ICG		42	IFC		ICC	DC02
11	IBD	DR15	IDF		43	IFD		IDB	DB03
12	IBE	DR28	IEE	DR08	44	IFE	DR09	IEA	DA02
13	IBF	DR29	IFD		45	IFF	DR13	IFH	DR29
14	IBG	DR30	IGC		46	IFG	DR25	IGG	DR26
15	IBH	DR31	IHB	DD03	47	IFH	DR29	IHF	DR15
16	ICA	DA02	IAB	DR09	48	IGA	DC02	IAF	DR25
17	ICB	DB02	IBA	DR12	49	IGB	DC03	IBE	DR28
18	ICC	DC02	ICH		50	IGC		ICD	DD02
19	ICD	DD02	IDG		51	IGD		IDC	DC03
20	ICE		IEF	DR12	52	IGE	DR10	IEB	DA03
21	ICF		IFE	DR09	53	IGF	DR14	IFA	DB02
22	ICG		IGD		54	IGG	DR26	IGH	DR30
23	ICH		IHC		55	IGH	DR30	IHG	DR27
24	IDA	DA03	IAC	DR10	56	IHA	DD02	IAG	DR26
25	IDB	DB03	IBB	DR13	57	IHB	DD03	IBF	DR29
26	IDC	DC03	ICA	DA02	58	IHC		ICE	
27	IDD	DD03	IDH		59	IHD		IDD	DD03
28	IDE		IEG	DR24	60	IHE	DR11	IEC	DE02
29	IDF		IFF	DR13	61	IHF	DR25	IFB	DB03
30	IDG		IGE	DR10	62	IHG	DR27	IGA	DC02
31	IDH		IHD		63	IHH	DR31	IHH	DR31

Table 21. LM1 Option Test-point Assignments for Network/Memory Modules

Input Bit	Term	Option	Input Bit	Term	Option	Input Bit	Term	Option	Memory Port
0	IQA	LA01	32	ISA	LC1	64	IBQ	LC1	0
1	IQB	LA02	33	ISB	LC2	65	IBR	LC3	1
2	IQC	LA03	34	ISC	LC3	66	IBS	LC4	2
3	IQD	LA04	35	ISD	LC4	66	IBT	LC6	3
4	IQE	LA05	36	ISE	LC5	67	IBU	LC7	4
5	IQF	LA06	37	ISF	LC6	68	IBV	LC9	5
6	IQG	LA07	38	ISG	LC7	69	IBW	LC10	6
7	IQH	LA08	39	ISH	LC8	70	IBX	LC12	7
8	IQI	LA09	40	ISI	LC9				
9	IQJ	LA10	41	ISJ	LC10				
10	IQK	LA11	42	ISK	LC11				
11	IQL	LA12	43	ISL	LC12				
12			44						
13			45						
14			46						
15			47						
16	IRA	LB01	48	ITA	LM01				
17	IRB	LB02	49						
18	IRC	LB03	50						
19	IRD	LB04	51						
20	IRE	LB05	52						
21	IRF	LB06	53						
22	IRG	LB07	54						
23	IRH	LB08	55						
24	IRI	LB09	56						
25	IRJ	LB10	57						
26	IRK	LB11	58						
27	IRL	LB12	59						
28	IRM	LB13	60						
29	IRN	LB14	61						
30			62						
31			63						

Table 22. LM2 Option Test-point Assignments for Network/Memory Modules

Input Bit	Term	Option	Input Bit	Term	Option	Input Bit	Term	Option	Memory Port
0	IQA	LA13	32	ISA	LC13	64	IBQ	LC13	0
1	IQB	LA14	33	ISB	LC14	65	IBR	LC15	1
2	IQC	LA15	34	ISC	LC15	66	IBS	LC16	2
3	IQD	LA16	35	ISD	LC16	66	IBT	LC18	3
4	IQE	LA17	36	ISE	LC17	67	IBU	LC19	4
5	IQF	LA18	37	ISF	LC18	68	IBV	LC21	5
6	IQG	LA19	38	ISG	LC19	69	IBW	LC22	6
7	IQH	LA20	39	ISH	LC20	70	IBX	LC24	7
8	IQI	LA21	40	ISI	LC21				
9	IQJ	LA22	41	ISJ	LC122				
10	IQK	LA23	42	ISK	LC123				
11	IQL	LA24	43	ISL	LC124				
12			44						
13			45						
14			46						
15			47						
16	IRA	LA15	48	ITA	LM02				
17	IRB	LA16	49						
18	IRC	LA17	50						
19	IRD	LA18	51						
20	IRE	LA19	52						
21	IRF	LA20	53						
22	IRG	LA21	54						
23	IRH	LA22	55						
24	IRI	LA23	56						
25	IRJ	LA24	57						
26	IRK	LA25	58						
27	IRL	LA26	59						
28	IRM	LA27	60						
29	IRN	LA28	61						
30			62						
31			63						

Master Clear Levels

CRAY T90 series systems do not have a broadcast global master clear switch. Instead, six levels of master clear functions are performed through the maintenance channel:

2
- Memory master clear
- Memory access master clear
- CPU master clear
- I/O quadrant master clear
- Shared master clear
- System master clear (the highest level master clear which drops sanity code)

level of severity

Master clear functions initialize and restore the mainframe system and return CPUs to previously set states. Master clear functions and sanity code functions are separate functions, yet they are integrated. The lack of sanity code forces a master clear on all CPUs and all other system modules. Table 23 lists master clear and sanity code functions.

Table 23. Master Clear and Sanity Code Functions

Code	Function†
60	Memory master clear on
61	Memory master clear off
62	Memory access master clear on
63	Memory access master clear off
64	CPU master clear on
65	CPU master clear off
70	Memory sanity code on
71	Memory sanity code off
72	I/O sanity code on
73	I/O sanity code off

† Master clear on and sanity code off functions are default states that are forced when no sanity code is present or when a reset configuration code (77) is sent.

Configuration codes routed to CP modules control the passing of sanity codes to modules that do not receive sanity code enable commands. Shared modules are instructed to send sanity code to specific CP modules. Network modules have no loop controllers and therefore must pass sanity enable codes to connected memory modules. Sanity code is sent from a CP module to a network module under control of configuration codes.

The codes select the appropriate memory sections and enable memory to receive sanity code from CP modules. For these reasons, sanity code must precede any configuration or master clear sequence.

System Master Clear

This is the highest level of master clear. This signal enters the I/O module on the LOSP maintenance channel. The system master clear signal resets the LOSPX interface for quadrant 0, which is the maintenance channel. It also resets the sanity code generator, which collapses the entire system.

Memory Master Clear

This is the second highest level of master clear. When activated from a CP module, this function clears all network and memory module request and data queues. It also resets the control logic of the network and memory modules that receive it. The memory master clear function does not affect the contents of memory and does not affect the SBCDBD logic.

The memory master clear function affects only modules that are receiving valid sanity code. If a network or memory module is not receiving sanity code, it will be forced into this master clear state.

The memory master clear function, like all other configuration functions, must be cleared. For example, if another CPU tries to access a memory module that is receiving master clear, the CPU reference will hang until the master clear is cleared or released.

In the CPU, the memory master clear function also forces memory access master clear and CPU master clear active. Therefore, all memory logic is reset. No configuration functions are affected, but all soft switches are cleared.

I/O Quadrant Master Clear

There are four independent I/O master clear signals, one for each quadrant on the I/O module. Each of these signals halt all channel activity and clear channel control, memory access controls, and data buffer control for the quadrant. It also clears diagnostic modes for the channels in the quadrant.

Shared Master Clear

This signal clears all the logic on the shared module. There shouldn't be a need for this signal very often because the flush command from the CPU clears all logic on the shared module associated with a particular CPU.

Memory Access Master Clear

This level of master clear is used to clear the memory interface logic in the CPU. This clears the priority and request logic and blocks any memory-to-I/O references that are in the CPU. If a request has already left and is in the network or memory logic, it will complete. The memory access master clear function does not affect the memory modules.

A reset configuration code (77) or a memory master clear on (60) code activates the memory access master clear function. Loss of sanity code to the CP module also activates this function. Once activated, this memory access master clear must be cleared with a 63 configuration code (memory access master clear off).

CPU Master Clear

This is the lowest-level master clear function. This function halts the CPU and, upon restart, causes the CPU to do an initial exchange sequence. This function does not destroy A and S register values, so a drop of CPU master clear and a restart causes the initial exchange package to resume execution.

When the CPU master clear is activated, all CPU maintenance codes, 100 through 178, are cleared. The CPU master clear does not clear any of the CPU configuration codes (00 through 78).

A reset configuration code (77), a memory master clear on (60) code, or a memory access master clear on (62) code activates the CPU master clear function. Loss of sanity code to the CP module also activates the CPU master clear function. Once activated, the CPU master clear must be cleared with a 65 configuration code (CPU master clear off).

Memory and I/O Sanity Code Functions

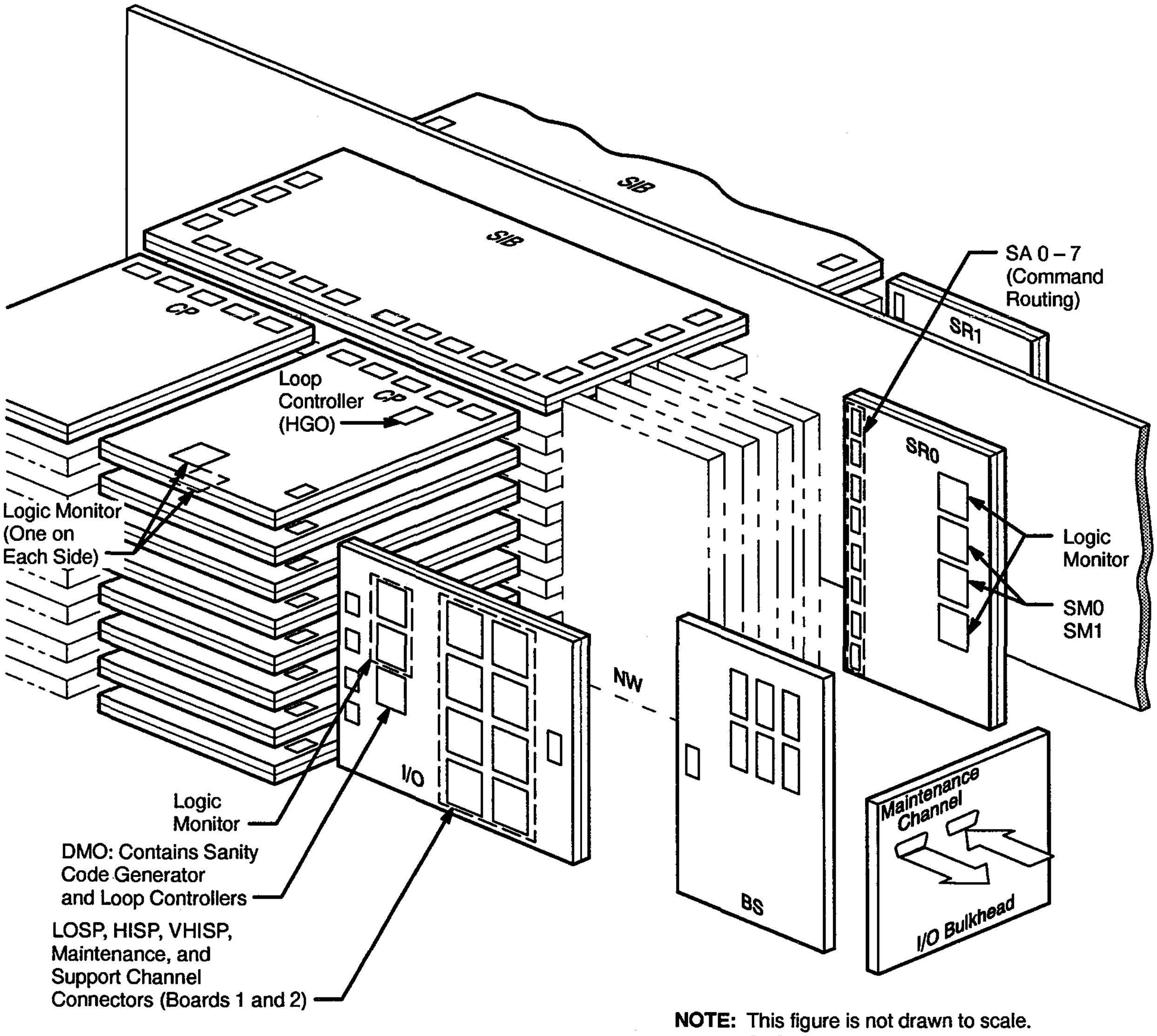
Sanity code is not sent directly from CP modules into memory. From CP modules, sanity code sent to network and memory modules is controlled by memory module configuration codes and the memory sanity code on and off codes (70 and 71). When a network module receives sanity code from any connected CP module, it passes the sanity code to all connected memory modules. When a memory module receives a sanity code, it automatically returns the sanity code to the network module that sent the code. The network module passes the sanity code to the connected CP modules. There are 8 test points in each CP module that allow returned sanity codes to be verified.

There is a separate path for sanity code from a CP module into memory, one for each of the 8 memory ports. Sanity code is routed into memory when memory sanity code on (code 70) is ANDed with one of the memory module function codes (00, 01, 02) that selects the number of memory sections.

Configuration codes that define subsections and banks have no effect on memory sanity codes. If memory is degraded (the number of subsections or banks is taken out of the configuration), CP modules still send sanity codes to the degraded memory modules. Even though a CP module will never send a memory request to degraded memory modules, it can still send them sanity code.

Maintenance Channel Theory of Operations

This section provides more detailed information on the maintenance channel and sanity code operations in CRAY T90 series systems. Figure 12 illustrates some of the components used to configure, perform, and control maintenance activities. It shows the location of logic monitors, loop controllers, the sanity code generator, and other components. Refer to this illustration as you read the theory of operation in the rest of this document. Also refer to the 11x17 in. block diagrams at the end of this document for more detail.

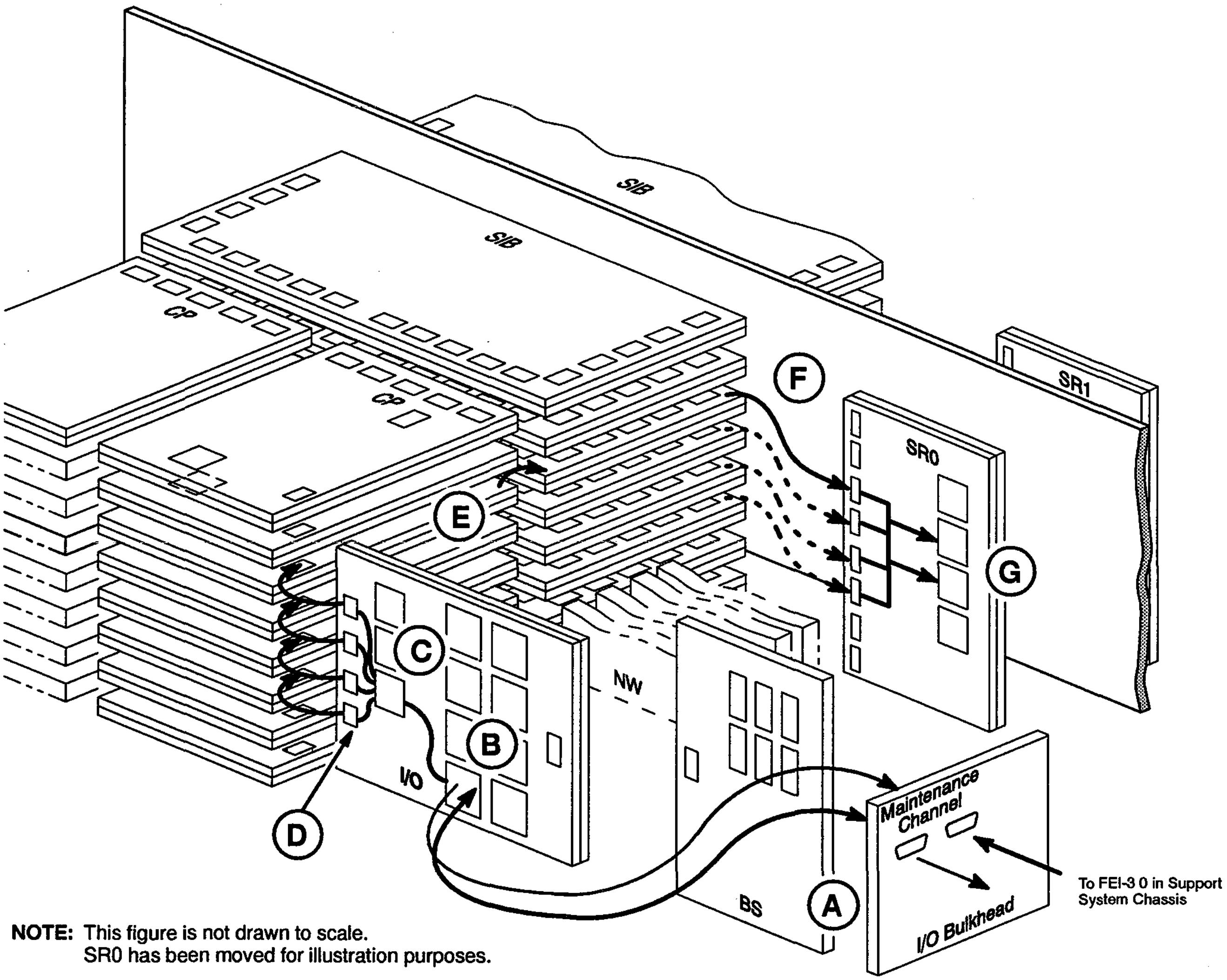

SIB
SIB
CP
CP
Loop Controller (HGO)
Logic Monitor (One on Each Side)
SA 0 – 7 (Command Routing)
SR1
SR0
Logic Monitor
SM0
SM1
NW
I/O
BS
Maintenance Channel
I/O Bulkhead
Logic Monitor
DMO: Contains Sanity Code Generator and Loop Controllers
LOSP, HISP, VHISP, Maintenance, and Support Channel Connectors (Boards 1 and 2)

NOTE: This figure is not drawn to scale. SR0 has been moved for illustration purposes.

Figure 12. Related Maintenance Channel Components

Initial Sanity Tree Configuration

The following process establishes an initial sanity tree. The letters correspond to the circled letters in Figure 13.



NOTE: This figure is not drawn to scale.
SR0 has been moved for illustration purposes.

Figure 13. Initial Sanity Tree Path to Shared Module

A. The LOSP maintenance channel leaving the VME in the support system chassis is connected to the maintenance channel port on the I/O bulkhead.

B. The maintenance channel connects to a LOSP I/O channel connector on the I/O module. The maintenance channel then enters a channel controller (DD0). A continuation route code (37) puts the maintenance channel in data mode.

In data mode, the sanity code generator (DM0 option) must handshake (exchange ready/resume signals) with the maintenance channel upstream before replying to the channel controller on the I/O module (DD option). A disconnect signal from upstream (route code 35) or an attention signal from the channel controller will take the maintenance channel out of data mode.

C. A value of 46464646_8 is sent in a command, along with a 36 route code, through the maintenance channel to initialize the sanity code generator. The I/O module has not received any sanity code at this time.

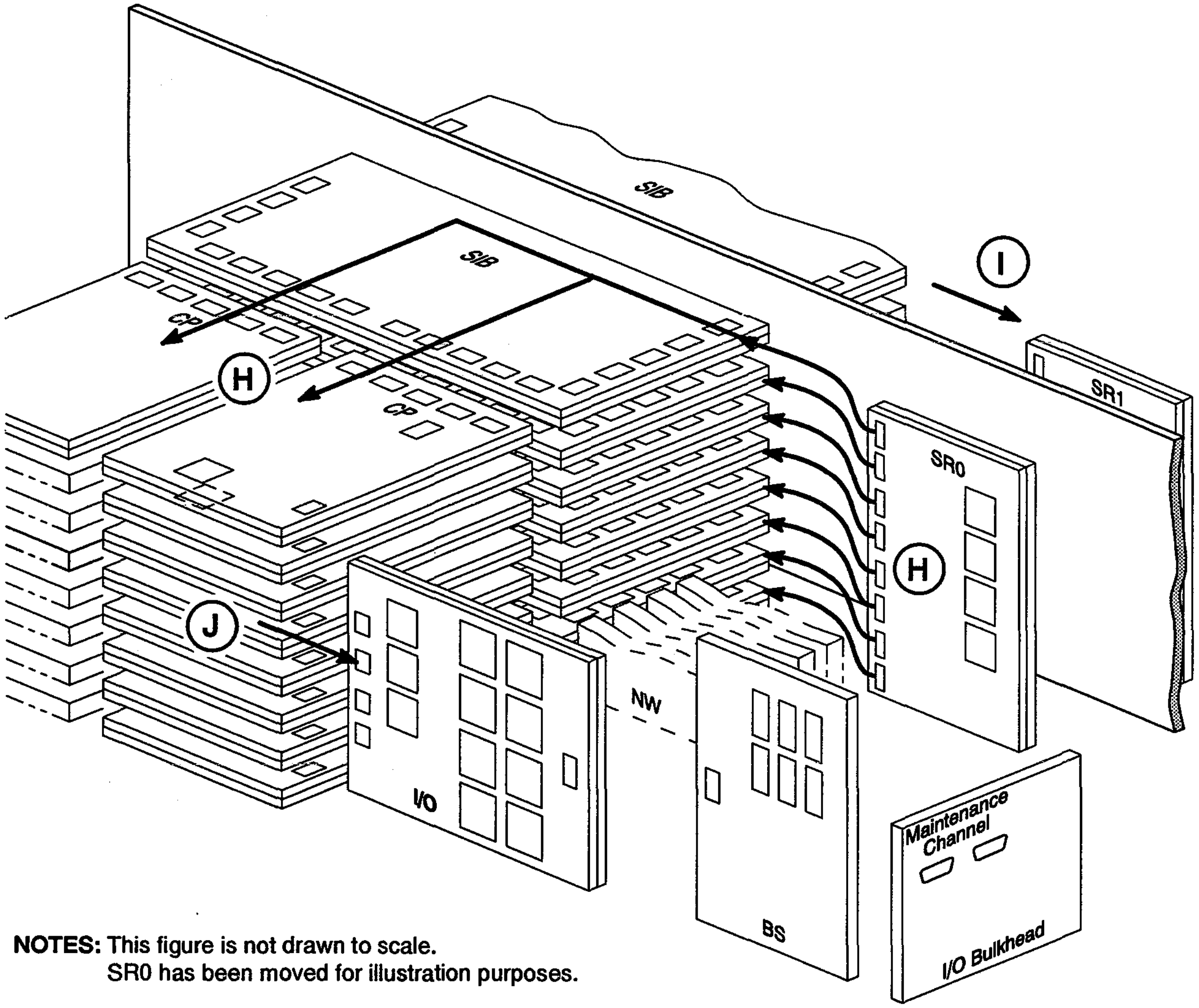
D. Sanity code to the shared module. Sanity code is typically sent to DC0, which routes the sanity code through the attached CP module and on to the shared module.

E. Any of the DC options and CP modules can be used; however, the SCE configuration program will normally try DC0 first. If this path is not successful, the next DC option/CP module is used. Note that this sanity code is routed through the CP module and that the CP module does not recognize the sanity code. All error log and maintenance channel traffic is routed in the same direction.

F. The sanity code enters the local shared module through the SA option that corresponds to the CP port. The dotted lines indicate that it can be received through any of the four ports.

G. Sanity code on the shared module is routed to the SM0 and SM1 options.

Refer to Figure 14 for the remaining processes of the initial sanity tree configuration process.



NOTES: This figure is not drawn to scale.
SR0 has been moved for illustration purposes.

Figure 14. Initial Sanity Tree Path from Local Shared Module

H. The shared module forwards the sanity code to the CPUs that are to be configured, which can be any combination that the user wants.

I. If the CP and I/O modules on the other side of the system are to be included in the sanity tree, the I/O module then sends sanity code to the remote shared module. The remote shared module then initializes its attached CP modules.

J. The CP module that was used as a route-through path sends sanity code to initialize the I/O module.

The path the sanity code takes to build the tree is the same as the return path the system takes to communicate with the maintenance channel. It is also possible that in a CRAY T932 system, the two halves of the system could be getting sanity code from two different maintenance channels and send responses back only to the channel that initiated the code. The sanity codes could not overlap one and other; there is a sanity code arbitrator associated with each module that performs a first-come, first-serve arbitration.

It can also be noted that on a large system, a shared module should serve as a central fanout point for sanity code after the I/O module. This centralized fanout point can be used because both of the shared modules are connected. By using both shared modules, one distribution level should be removed from the tree structure.

Swapping Sanity Code Generators

In a system with more than one I/O module, only one I/O module can be the master sanity code generator. It is possible to designate a master sanity code generator on a different I/O module without dropping sanity code to the entire system, which may be useful when one half of the system must be powered down for maintenance without powering down the other half.

After the initial sanity tree is built from one I/O module, the sanity code generator on a second I/O module is started. Using the maintenance channel on the first I/O module, sanity codes are enabled to build a complete new sanity tree. At this point, two separate sanity trees are enabled though only one is actually being used.

A configuration code of 40 is sent to the DM option on the second I/O module, which causes the DM option to rearbitrate where master sanity code is coming from. It will now be coming from the sanity code generator within the DM option instead of from a CP module. At this point, communications cannot take place between the original maintenance channel and the second I/O module.

The sanity code generator on the first I/O module can now be turned off (by loading an invalid word into the sanity code generator). As the first sanity tree collapses, a new one is built without losing sanity code at any point.

Because the maintenance channel and the error logger follow the sanity tree, any messages in these paths may be corrupted and/or lost during reconfiguration of the sanity tree. Once the new sanity tree is established, all error log and maintenance channel traffic follows this new path.

Maintenance Channel Operation on CP Modules

The following subsections describe the maintenance channel operations that the HB, HC, and HG options on the CP modules provide. The block diagrams on pages 65 and 67 show related signal paths for these options.

HB and HC Options

The HB option controls the I/O references, acts as the interface from the I/O module to the shared module interface, and sums the SECDED errors from the I/O data buffers.

The HB option contains the majority of the maintenance channel logic. The HC option handles only a few of the tasks, namely capturing any data following a 37 (continuation) route code and capturing read data from memory. Every data word following a 37 route code can be captured, even though the data words may be part of a continuation sequence or a starting address. The ability to capture every data word following a 37 route code is done to enable the desired data word to eventually be captured. The HB option signals the HC option when to send out data to memory (through the HA0/1 options). When the HB option sends read data on the return channel, the HC option monitors the return channel. When the HC option detects a 36 route code, it inserts every other bit into the return stream.

Maintenance channel data is sent out from the HG to the HB and the HC at the same time. The return path starts on the HB option and is passed through the HC option. If memory data is being read, the HC inserts the even data bits into the data stream.

Maintenance Channel DMA Controller

Maintenance channel direct memory access (DMA) interface logic performs the following operations:

- Write Memory command. Data is transferred into main memory using the I/O memory port.

- Read Memory command. Data is transferred from main memory using the I/O memory port.

- Send Function to I/O Module command. The maintenance channel is used to send a function to the I/O module.

- Read I/O Status command. The maintenance channel is used to read status from the I/O module.

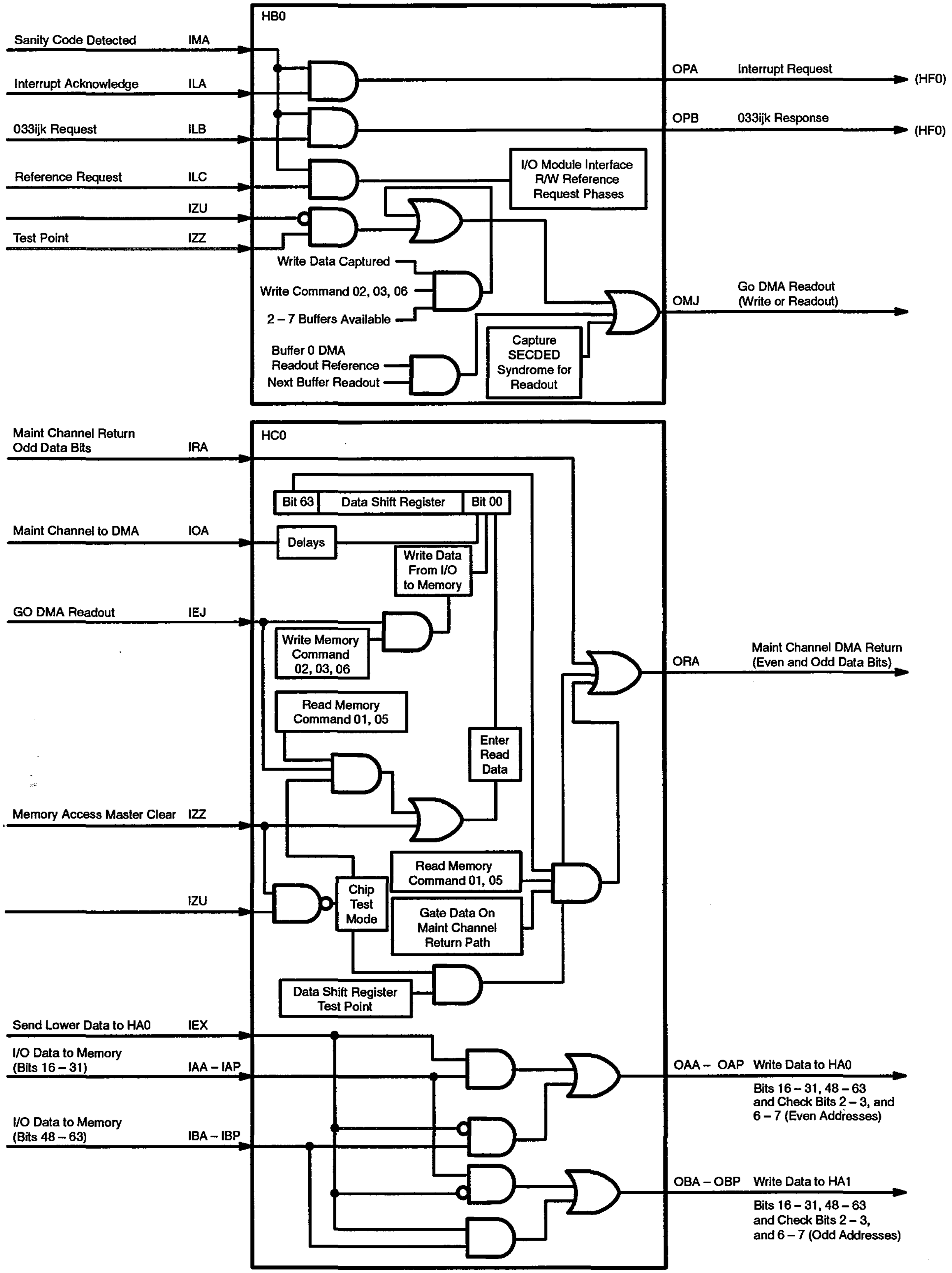

HB0
Sanity Code Detected
IMA
Interrupt Acknowledge
ILA
033ijk Request
ILB
Reference Request
ILC
IZU
Test Point
IZZ
OPA
Interrupt Request
(HF0)
OPB
033ijk Response
(HF0)
I/O Module Interface R/W Reference Request Phases
Write Data Captured
Write Command 02, 03, 06
2 – 7 Buffers Available
Buffer 0 DMA Readout Reference
Next Buffer Readout
Capture SECDED Syndrome for Readout
OMJ
Go DMA Readout (Write or Readout)
HC0
Maint Channel Return Odd Data Bits
IRA
Bit 63
Data Shift Register
Bit 00
Maint Channel to DMA
IOA
Delays
Write Data From I/O to Memory
GO DMA Readout
IEJ
Write Memory Command 02, 03, 06
Read Memory Command 01, 05
Enter Read Data
ORA
Maint Channel DMA Return (Even and Odd Data Bits)
Memory Access Master Clear
IZZ
IZU
Chip Test Mode
Read Memory Command 01, 05
Gate Data On Maint Channel Return Path
Data Shift Register Test Point
Send Lower Data to HA0
IEX
I/O Data to Memory (Bits 16 – 31)
IAA – IAP
I/O Data to Memory (Bits 48 – 63)
IBA – IBP
OAA – OAP
Write Data to HA0
Bits 16 – 31, 48 – 63 and Check Bits 2 – 3, and 6 – 7 (Even Addresses)
OBA – OBP
Write Data to HA1
Bits 16 – 31, 48 – 63 and Check Bits 2 – 3, and 6 – 7 (Odd Addresses)

Figure 15. Data Paths Through HB and HC Options

HG Option

The HG option provides the following maintenance functions for the CP module. Also refer to Figure 16 and Figure 17 for more information.

- Sanity code recognizers and the home port arbiter for first sanity code detection.
- Loop forwarders to send out maintenance channel functions to options on this module.
- DMA control forwards all maintenance channel DMA functions to the HB and HC options.
- Maintenance channel functions forwarded to the I/O module. Logic monitor functions are also forwarded from the HG option to the I/O module.
- Maintenance channel return messages are forwarded from this option.
- Error logger channel functions for the CP module are performed on this option.

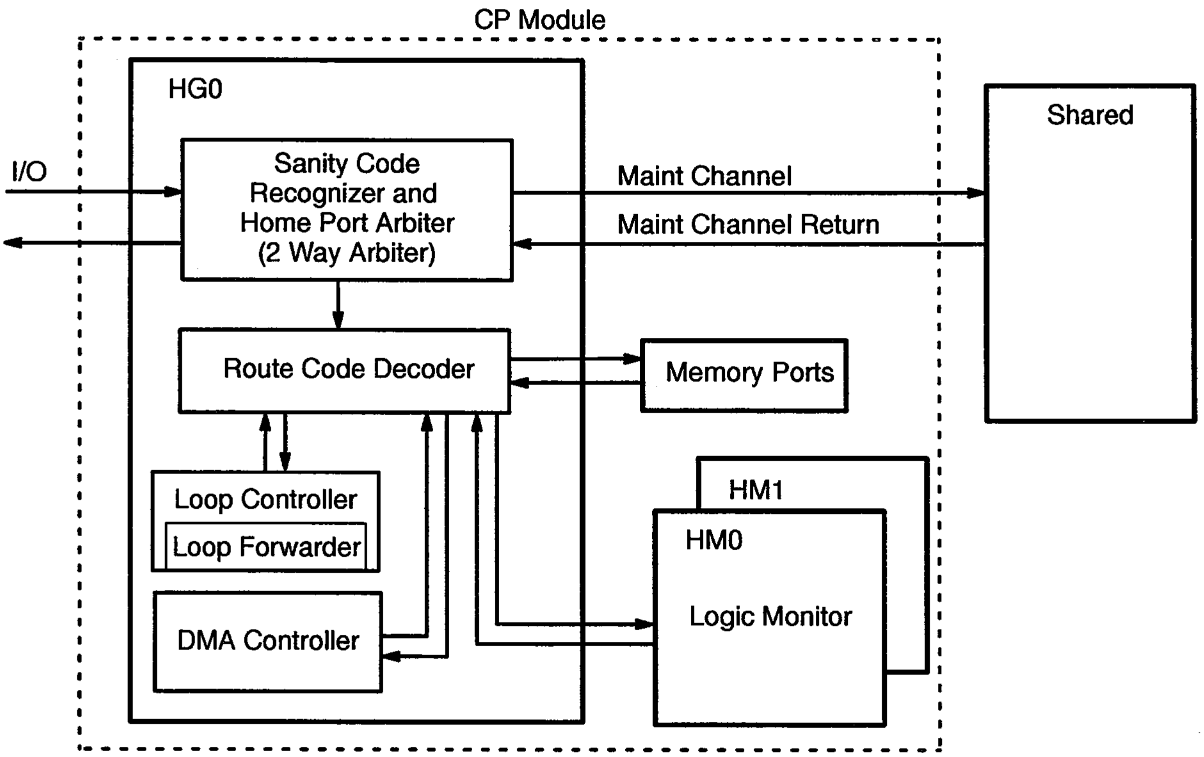


Figure 16. Maintenance Function on the HG Option

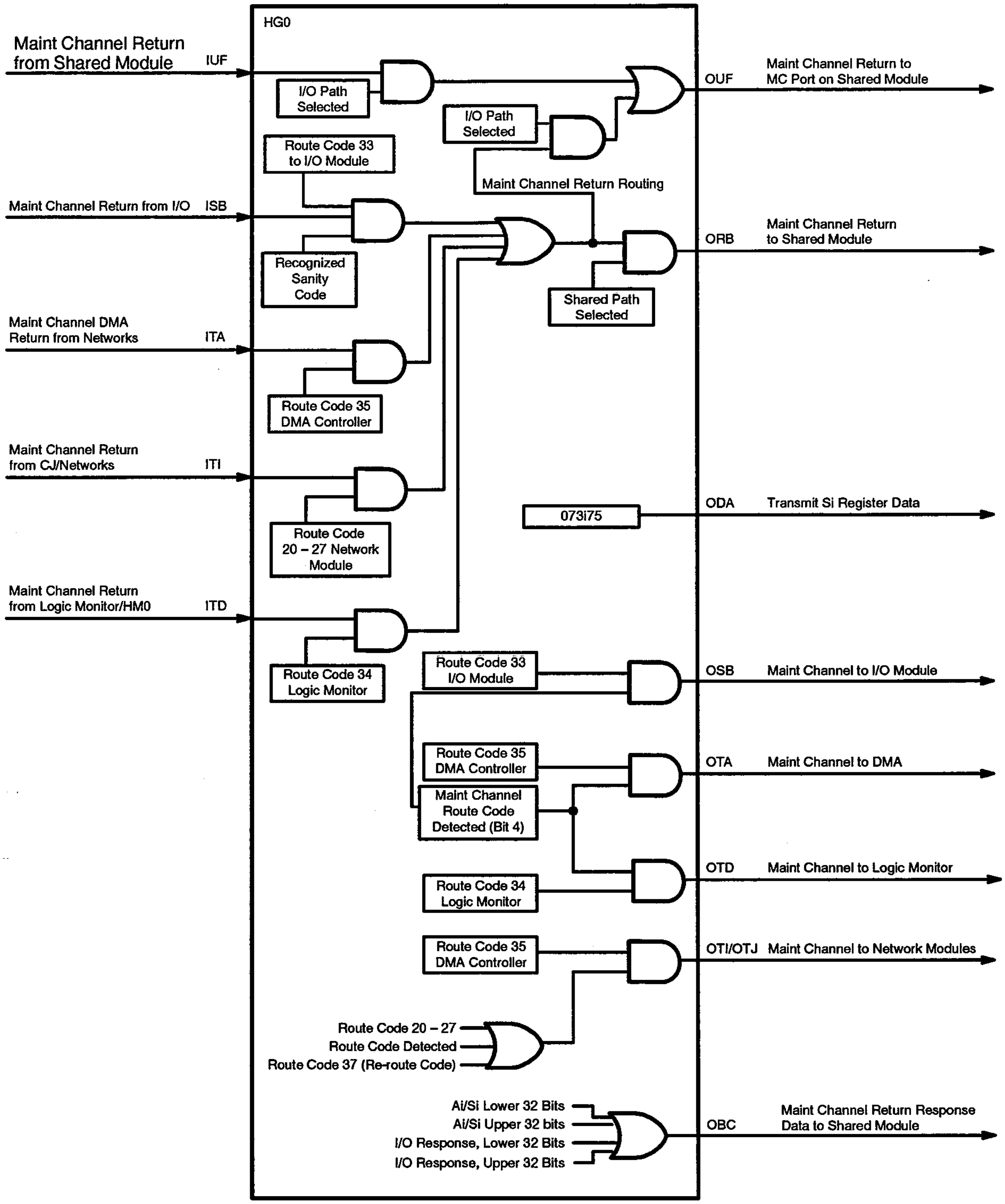

HG0
Maint Channel Return from Shared Module
IUF
I/O Path Selected
I/O Path Selected
OUF
Maint Channel Return to MC Port on Shared Module
Route Code 33 to I/O Module
Maint Channel Return Routing
Maint Channel Return from I/O
ISB
Recognized Sanity Code
ORB
Maint Channel Return to Shared Module
Shared Path Selected
Maint Channel DMA Return from Networks
ITA
Route Code 35 DMA Controller
Maint Channel Return from CJ/Networks
ITI
073i75
ODA
Transmit Si Register Data
Route Code 20 – 27 Network Module
Maint Channel Return from Logic Monitor/HM0
ITD
Route Code 34 Logic Monitor
Route Code 33 I/O Module
OSB
Maint Channel to I/O Module
Route Code 35 DMA Controller
OTA
Maint Channel to DMA
Maint Channel Route Code Detected (Bit 4)
OTD
Maint Channel to Logic Monitor
Route Code 34 Logic Monitor
Route Code 35 DMA Controller
OTI/OTJ
Maint Channel to Network Modules
Route Code 20 – 27
Route Code Detected
Route Code 37 (Re-route Code)
Ai/Si Lower 32 Bits
Ai/Si Upper 32 bits
I/O Response, Lower 32 Bits
I/O Response, Upper 32 Bits
OBC
Maint Channel Return Response Data to Shared Module

Figure 17. Maintenance Channel Paths from HGO Option

Maintenance Channel Operation on the Shared Modules

The SM option on the shared modules provides sanity code, maintenance channel, error logger, and master clear functions. There are two SM options per shared module; they act identically and simultaneously for all functions.

Sanity Code

The two SM chips may receive sanity codes from I/O modules through eight CP module paths and two paths from the other (remote) shared module. The path that first supplies a valid sanity code becomes the controlling port. I/O modules have priority over remote shared modules, and SM0 inputs have priority over SM1 inputs.

Sanity code is returned to any path supplying sanity code, regardless of whether or not it is the controlling port. This means that it is not possible for a sender of sanity code to determine whether the shared module has provided the master (controlling) sanity code.

If sanity code is lost from the controlling port while another port is sending a sanity code, the other port becomes the controlling port without resetting the shared module or disrupting the sanity codes being sent to the CP modules.

Absence of any sanity code to the SM options causes the shared module master clear and logic monitor master clear to be asserted. When a sanity code is recognized, the logic monitor master clear is dropped, but the shared module master clear remains until turned off by a 66 function code. A loop command of 67 turns on shared module master clear and asserts logic monitor master clear for exactly 1 clock period.

Sanity code is forwarded to the SA options only after a 66 configuration code has been received. (A 67 code turns this state off.) Each of the SA options forwards the sanity code to a CP module after a 074 code has been received. Both codes must be used before sanity code can be sent to a CP module. This is because the SA options do not clear the 074 state on reset or power-up, so the SM option has a master shut-off valve, which is turned off by lack of incoming sanity code.

Another function of the SM options is to determine whether a valid remote shared module exists. The remote shared module exists if it sends a sanity code return after the local shared module sends it sanity code. (If a shared module sends sanity code to the remote shared module, the remote shared module always sends sanity code return.)

Sanity connection between two shared modules can use any of four paths. Normally, the SM0 option sends sanity codes to the other SM0 chip. Using a configuration mode, an alternate path between SM1 chips can be used instead. If sanity had started at the other board, there would be two additional independent paths to choose from.

Maintenance Channel Data to all CP Modules

Maintenance channel data is sent to CP modules and to the remote shared module regardless of sanity codes or sanity code returns. However, maintenance channel return and error logger messages are gated with sanity code return from that CP module.

A route code of 31 performs two functions to ensure that a message simultaneously reaches all connected CP modules in the system. First, the 31 route code sends a 033 route code to the remote shared module (if it is sending a sanity code return). The local shared module then sends a message to all of its connected CP modules after a 12-clock-period delay. The 33 route code causes the remote shared module to send the message to all CPUs attached to that module, without the 12-clock-period delay. The 12-clock-period delay for the local shared modules compensates for the delay in sending the message to the remote shared module.

Defining Local and Remote Shared Modules

Two inputs per SM option are strapped (hardwired) to the SIB to determine whether the shared module is the local or remote (0 or 1). Normally the IWA input (connector za010) to SM0 is used for this purpose. Using a configuration code, the IWA input of SM1 (connector za012) can be used instead. Refer to the block diagram, Figure 22, on page 69. The IWE inputs are not used for this function.

Configuration codes can be used to override these inputs altogether and force the shared module position to 0 or 1. You need to be careful when using configuration codes to ensure that both shared modules are not set to the same number. The SM chips ensure that both SM chips on the same module each recognize the module position of the other; this is done by combining the codes from the two options with an OR gate. Override codes should be broadcast to both SM options to get expected results.

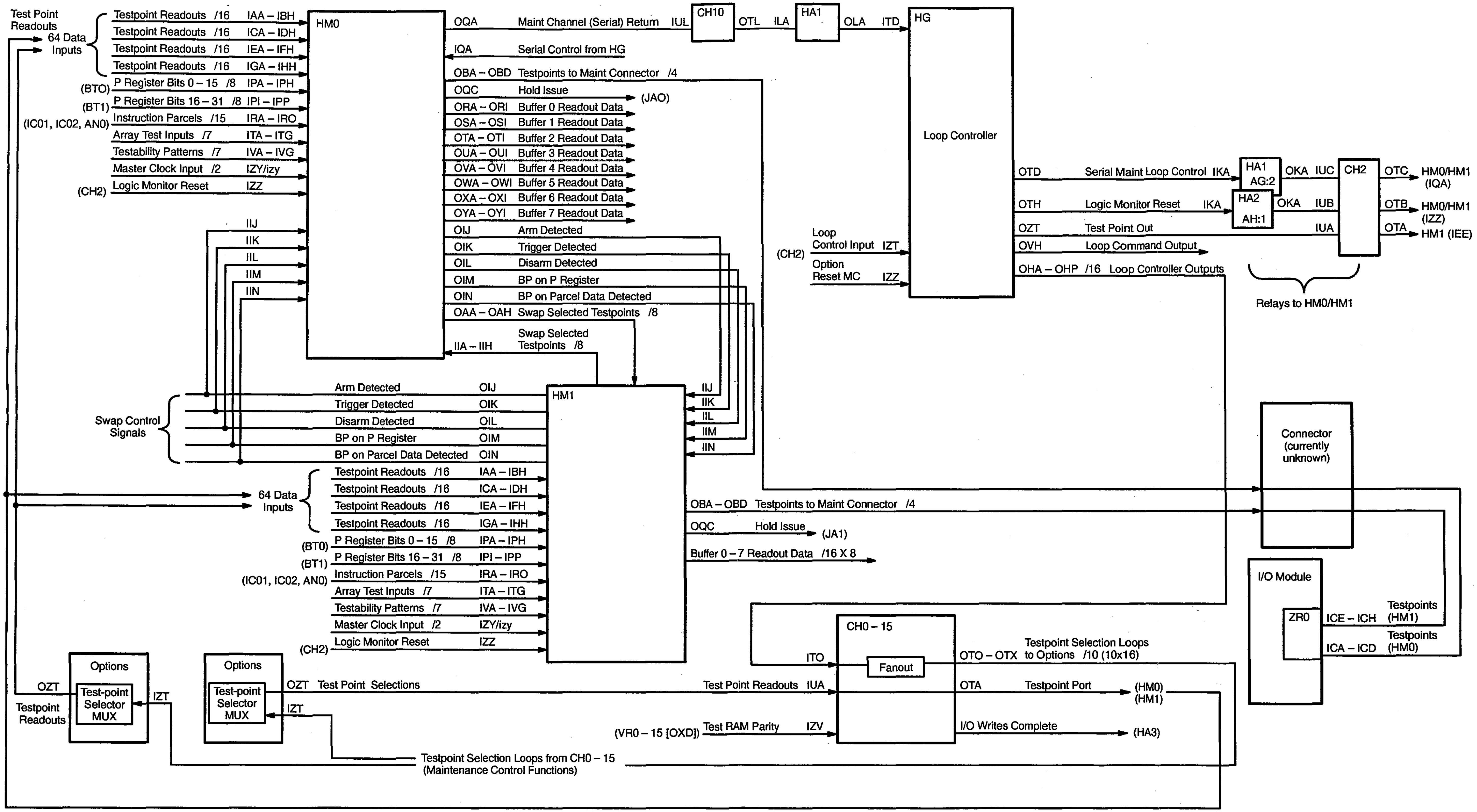

Test Point Readouts
64 Data Inputs
Testpoint Readouts /16 IAA – IBH
Testpoint Readouts /16 ICA – IDH
Testpoint Readouts /16 IEA – IFH
Testpoint Readouts /16 IGA – IHH
(BTO) P Register Bits 0 – 15 /8 IPA – IPH
(BT1) P Register Bits 16 – 31 /8 IPI – IPP
(IC01, IC02, AN0) Instruction Parcels /15 IRA – IRO
Array Test Inputs /7 ITA – ITG
Testability Patterns /7 IVA – IVG
Master Clock Input /2 IZY/izy
(CH2) Logic Monitor Reset IZZ
IIJ
IIK
IIL
IIM
IIN
HM0
OQA Maint Channel (Serial) Return IUL
IQA Serial Control from HG
OBA – OBD Testpoints to Maint Connector /4
OQC Hold Issue
(JAO)
ORA – ORI Buffer 0 Readout Data
OSA – OSI Buffer 1 Readout Data
OTA – OTI Buffer 2 Readout Data
OUA – OUI Buffer 3 Readout Data
OVA – OVI Buffer 4 Readout Data
OWA – OWI Buffer 5 Readout Data
OXA – OXI Buffer 6 Readout Data
OYA – OYI Buffer 7 Readout Data
OIJ Arm Detected
OIK Trigger Detected
OIL Disarm Detected
OIM BP on P Register
OIN BP on Parcel Data Detected
OAA – OAH Swap Selected Testpoints /8
IIA – IIH Swap Selected Testpoints /8
CH10
OTL ILA
HA1
OLA ITD
HG
Loop Controller
(CH2) Loop Control Input IZT
Option Reset MC IZZ
OTD Serial Maint Loop Control IKA
OTH Logic Monitor Reset IKA
OZT Test Point Out
OVH Loop Command Output
OHA – OHP /16 Loop Controller Outputs
HA1 AG:2
HA2 AH:1
OKA IUC
OKA IUB
IUA
CH2
OTC HM0/HM1 (IQA)
OTB HM0/HM1 (IZZ)
OTA HM1 (IEE)
Relays to HM0/HM1
Swap Control Signals
Arm Detected OIJ
Trigger Detected OIK
Disarm Detected OIL
BP on P Register OIM
BP on Parcel Data Detected OIN
HM1
IIJ
IIK
IIL
IIM
IIN
64 Data Inputs
Testpoint Readouts /16 IAA – IBH
Testpoint Readouts /16 ICA – IDH
Testpoint Readouts /16 IEA – IFH
Testpoint Readouts /16 IGA – IHH
(BT0) P Register Bits 0 – 15 /8 IPA – IPH
(BT1) P Register Bits 16 – 31 /8 IPI – IPP
(IC01, IC02, AN0) Instruction Parcels /15 IRA – IRO
Array Test Inputs /7 ITA – ITG
Testability Patterns /7 IVA – IVG
Master Clock Input /2 IZY/izy
(CH2) Logic Monitor Reset IZZ
OBA – OBD Testpoints to Maint Connector /4
OQC Hold Issue
(JA1)
Buffer 0 – 7 Readout Data /16 X 8
Connector (currently unknown)
I/O Module
ZR0
ICE – ICH Testpoints (HM1)
ICA – ICD Testpoints (HM0)
CH0 – 15
Fanout
ITO
OTO – OTX Testpoint Selection Loops to Options /10 (10x16)
Test Point Readouts IUA
OTA Testpoint Port
(HM0) (HM1)
(VR0 – 15 [OXD]) Test RAM Parity IZV
I/O Writes Complete
(HA3)
Options
Test-point Selector MUX
OZT
Testpoint Readouts
IZT
Options
Test point Selector MUX
OZT Test Point Selections
IZT
Testpoint Selection Loops from CH0 – 15 (Maintenance Control Functions)

Figure 18. CP Module Logic Monitor Block Diagram

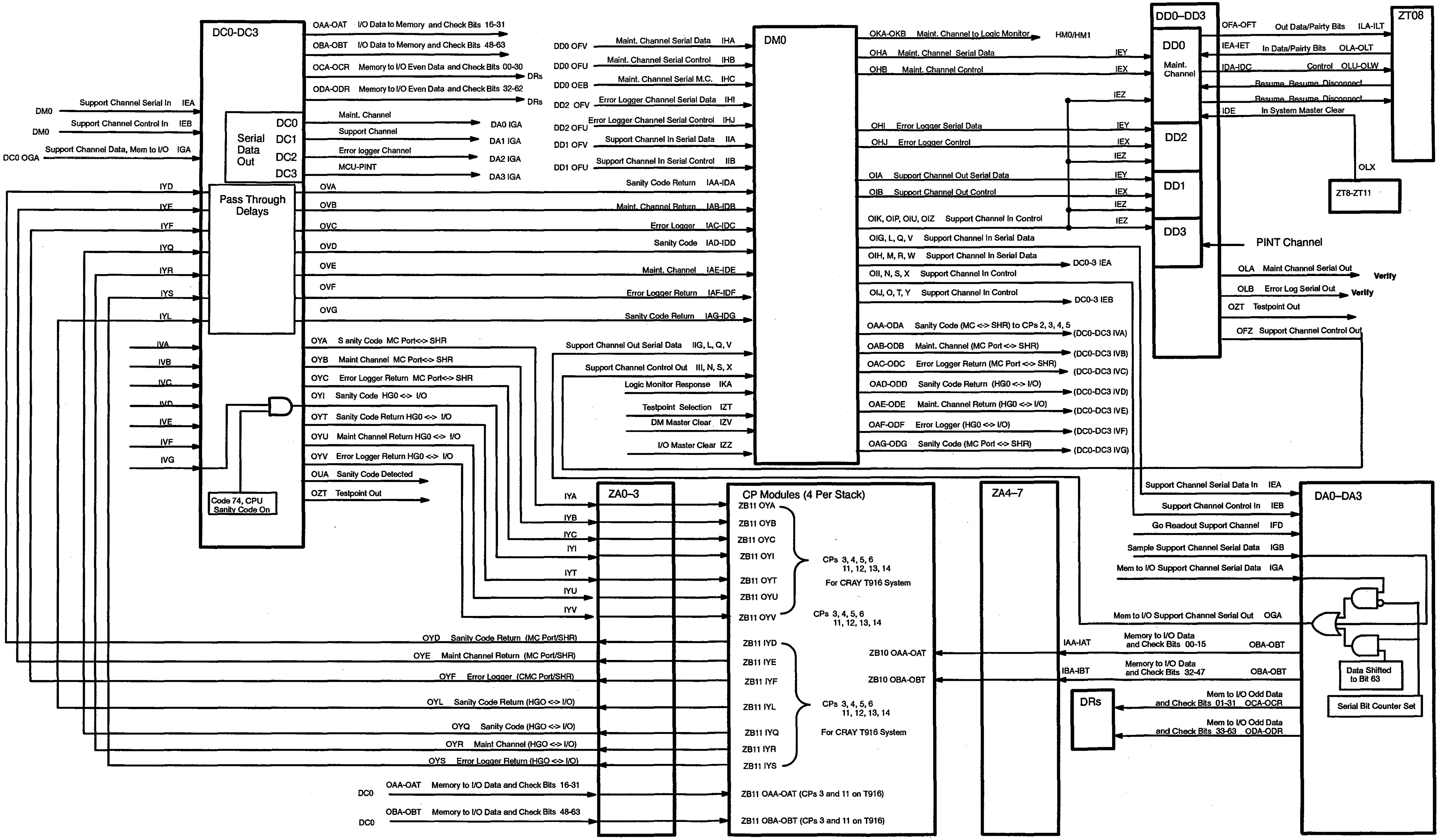

DC0-DC3
OAA-OAT I/O Data to Memory and Check Bits 16-31
OBA-OBT I/O Data to Memory and Check Bits 48-63
OCA-OCR Memory to I/O Even Data and Check Bits 00-30
DRs
ODA-ODR Memory to I/O Even Data and Check Bits 32-62
DRs
DM0 Support Channel Serial In IEA
DM0 Support Channel Control In IEB
DC0 OGA Support Channel Data, Mem to I/O IGA
Serial Data Out
DC0
DC1
DC2
DC3
Maint. Channel
DA0 IGA
Support Channel
DA1 IGA
Error logger Channel
DA2 IGA
MCU-PINT
DA3 IGA
Pass Through Delays
IYD
IYE
IYF
IYQ
IYR
IYS
IYL
IVA
IVB
IVC
IVD
IVE
IVF
IVG
OVA
OVB
OVC
OVD
OVE
OVF
OVG
OYA Sanity Code MC Port<-> SHR
OYB Maint Channel MC Port<-> SHR
OYC Error Logger Return MC Port<-> SHR
OYI Sanity Code HG0 <-> I/O
OYT Sanity Code Return HG0 <-> I/O
OYU Maint Channel Return HG0 <-> I/O
OYV Error Logger Return HG0 <-> I/O
OUA Sanity Code Detected
OZT Testpoint Out
Code 74, CPU Sanity Code On
DD0 OFV Maint. Channel Serial Data IHA
DD0 OFU Maint. Channel Serial Control IHB
DD0 OEB Maint. Channel Serial M.C. IHC
DD2 OFV Error Logger Channel Serial Data IHI
DD2 OFU Error Logger Channel Serial Control IHJ
DD1 OFV Support Channel In Serial Data IIA
DD1 OFU Support Channel In Serial Control IIB
Sanity Code Return IAA-IDA
Maint. Channel Return IAB-IDB
Error Logger IAC-IDC
Sanity Code IAD-IDD
Maint. Channel IAE-IDE
Error Logger Return IAF-IDF
Sanity Code Return IAG-IDG
Support Channel Out Serial Data IIG, L, Q, V
Support Channel Control Out III, N, S, X
Logic Monitor Response IKA
Testpoint Selection IZT
DM Master Clear IZV
I/O Master Clear IZZ
DM0
OKA-OKB Maint. Channel to Logic Monitor
HM0/HM1
OHA Maint. Channel Serial Data
IEY
OHB Maint. Channel Control
IEX
IEZ
OHI Error Logger Serial Data
IEY
OHJ Error Logger Control
IEX
IEZ
OIA Support Channel Out Serial Data
IEY
OIB Support Channel Out Control
IEX
IEZ
OIK, OIP, OIU, OIZ Support Channel In Control
IEZ
OIG, L, Q, V Support Channel In Serial Data
OIH, M, R, W Support Channel In Serial Data
DC0-3 IEA
OII, N, S, X Support Channel In Control
OIJ, O, T, Y Support Channel In Control
DC0-3 IEB
OAA-ODA Sanity Code (MC <-> SHR) to CPs 2, 3, 4, 5
(DC0-DC3 IVA)
OAB-ODB Maint. Channel (MC Port <-> SHR)
(DC0-DC3 IVB)
OAC-ODC Error Logger Return (MC Port <-> SHR)
(DC0-DC3 IVC)
OAD-ODD Sanity Code Return (HG0 <-> I/O)
(DC0-DC3 IVD)
OAE-ODE Maint. Channel Return (HG0 <-> I/O)
(DC0-DC3 IVE)
OAF-ODF Error Logger (HG0 <-> I/O)
(DC0-DC3 IVF)
OAG-ODG Sanity Code (MC Port <-> SHR)
(DC0-DC3 IVG)
DD0–DD3
DD0
Maint. Channel
DD2
DD1
DD3
OFA-OFT Out Data/Pairty Bits ILA-ILT
IEA-IET In Data/Pairty Bits OLA-OLT
IDA-IDC Control OLU-OLW
Resume Resume Disconnect
Resume Resume Disconnect
IDE In System Master Clear
ZT08
OLX
ZT8-ZT11
PINT Channel
OLA Maint Channel Serial Out
Verify
OLB Error Log Serial Out
Verify
OZT Testpoint Out
OFZ Support Channel Control Out
IYA
IYB
IYC
IYI
IYT
IYU
IYV
ZA0–3
OYD Sanity Code Return (MC Port/SHR)
OYE Maint Channel Return (MC Port/SHR)
OYF Error Logger (CMC Port/SHR)
OYL Sanity Code Return (HGO <-> I/O)
OYQ Sanity Code (HGO <-> I/O)
OYR Maint Channel (HGO <-> I/O)
OYS Error Logger Return (HGO <-> I/O)
DC0 OAA-OAT Memory to I/O Data and Check Bits 16-31
DC0 OBA-OBT Memory to I/O Data and Check Bits 48-63
CP Modules (4 Per Stack)
ZB11 OYA
ZB11 OYB
ZB11 OYC
ZB11 OYI
ZB11 OYT
ZB11 OYU
ZB11 OYV
CPs 3, 4, 5, 6 11, 12, 13, 14
For CRAY T916 System
CPs 3, 4, 5, 6 11, 12, 13, 14
ZB11 IYD
ZB11 IYE
ZB11 IYF
ZB11 IYL
ZB11 IYQ
ZB11 IYR
ZB11 IYS
ZB10 OAA-OAT
ZB10 OBA-OBT
CPs 3, 4, 5, 6 11, 12, 13, 14
For CRAY T916 System
ZB11 OAA-OAT (CPs 3 and 11 on T916)
ZB11 OBA-OBT (CPs 3 and 11 on T916)
ZA4–7
IAA-IAT
IBA-IBT
DRs
Support Channel Serial Data In IEA
Support Channel Control In IEB
Go Readout Support Channel IFD
Sample Support Channel Serial Data IGB
Mem to I/O Support Channel Serial Data IGA
Mem to I/O Support Channel Serial Out OGA
Memory to I/O Data and Check Bits 00-15 OBA-OBT
Memory to I/O Data and Check Bits 32-47 OBA-OBT
Mem to I/O Odd Data and Check Bits 01-31 OCA-OCR
Mem to I/O Odd Data and Check Bits 33-63 ODA-ODR
DA0-DA3
Data Shifted to Bit 63
Serial Bit Counter Set

Figure 19. Maintenance Channel and Sanity Code Paths on I/O Module

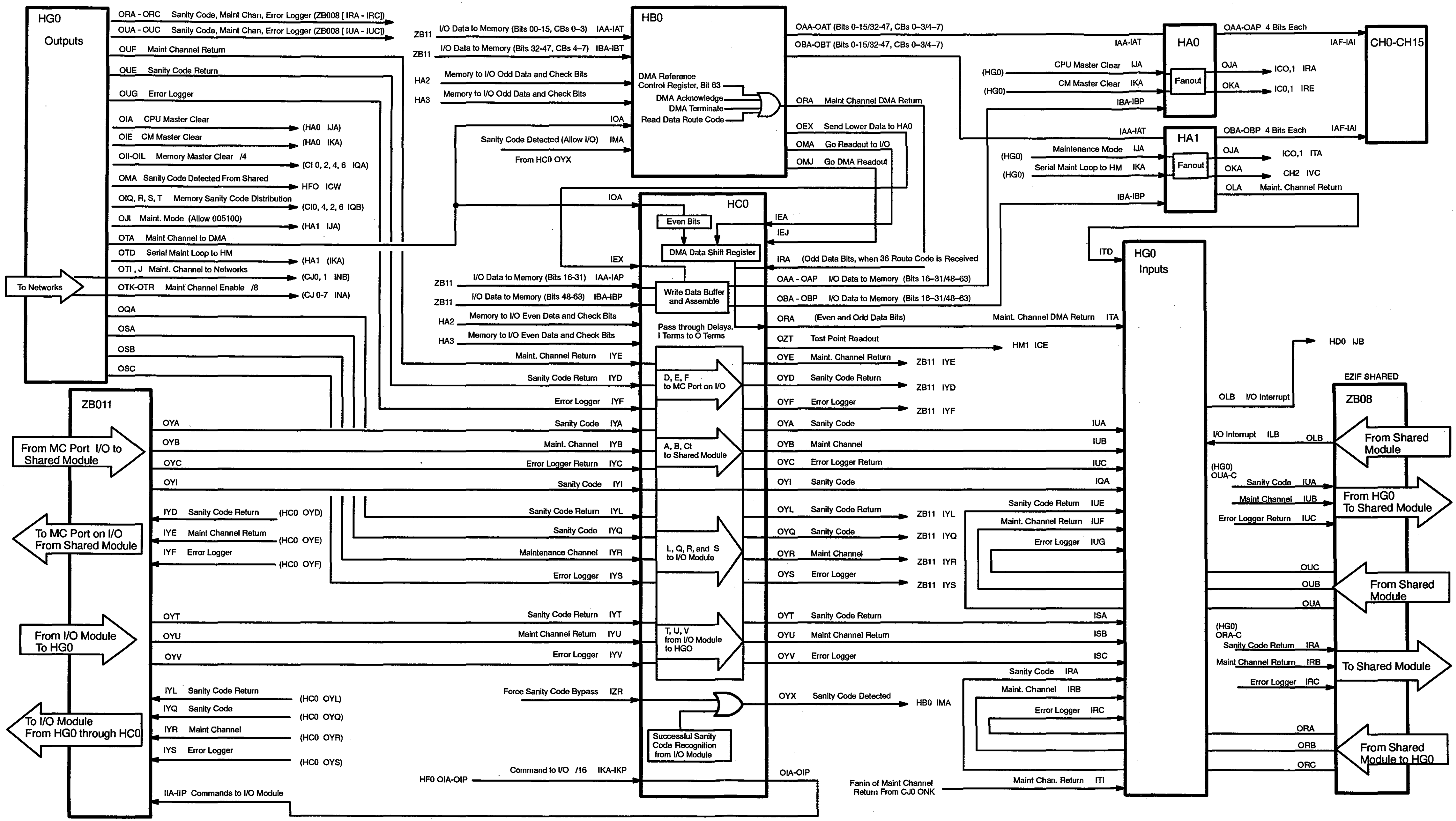

HG0
Outputs
ORA - ORC Sanity Code, Maint Chan, Error Logger (ZB008 [IRA - IRC])
OUA - OUC Sanity Code, Maint Chan, Error Logger (ZB008 [IUA - IUC])
OUF Maint Channel Return
OUE Sanity Code Return
OUG Error Logger
OIA CPU Master Clear
(HA0 IJA)
OIE CM Master Clear
(HA0 IKA)
OII-OIL Memory Master Clear /4
(CI 0, 2, 4, 6 IQA)
OMA Sanity Code Detected From Shared
HFO ICW
OIQ, R, S, T Memory Sanity Code Distribution
(CI0, 4, 2, 6 IQB)
OJI Maint. Mode (Allow 005100)
(HA1 IJA)
OTA Maint Channel to DMA
OTD Serial Maint Loop to HM
(HA1 (IKA)
OTI, J Maint. Channel to Networks
(CJ0, 1 INB)
To Networks
OTK-OTR Maint Channel Enable /8
(CJ 0-7 INA)
OQA
OSA
OSB
OSC
ZB011
From MC Port I/O to Shared Module
To MC Port on I/O From Shared Module
From I/O Module To HG0
To I/O Module From HG0 through HC0
OYA
OYB
OYC
OYI
IYD Sanity Code Return (HC0 OYD)
IYE Maint Channel Return (HC0 OYE)
IYF Error Logger (HC0 OYF)
OYT
OYU
OYV
IYL Sanity Code Return (HC0 OYL)
IYQ Sanity Code (HC0 OYQ)
IYR Maint Channel (HC0 OYR)
IYS Error Logger (HC0 OYS)
IIA-IIP Commands to I/O Module
ZB11 I/O Data to Memory (Bits 00-15, CBs 0-3) IAA-IAT
ZB11 I/O Data to Memory (Bits 32-47, CBs 4-7) IBA-IBT
HA2 Memory to I/O Odd Data and Check Bits
HA3 Memory to I/O Odd Data and Check Bits
IOA
Sanity Code Detected (Allow I/O) IMA
From HC0 OYX
HB0
DMA Reference Control Register, Bit 63
DMA Acknowledge
DMA Terminate
Read Data Route Code
OAA-OAT (Bits 0-15/32-47, CBs 0-3/4-7)
OBA-OBT (Bits 0-15/32-47, CBs 0-3/4-7)
ORA Maint Channel DMA Return
OEX Send Lower Data to HA0
OMA Go Readout to I/O
OMJ Go DMA Readout
IOA
HC0
Even Bits
DMA Data Shift Register
IEA
IEJ
IEX
IRA (Odd Data Bits, when 36 Route Code is Received
ZB11 I/O Data to Memory (Bits 16-31) IAA-IAP
ZB11 I/O Data to Memory (Bits 48-63) IBA-IBP
HA2 Memory to I/O Even Data and Check Bits
HA3 Memory to I/O Even Data and Check Bits
Write Data Buffer and Assemble
Pass through Delays. I Terms to O Terms
OAA - OAP I/O Data to Memory (Bits 16-31/48-63)
OBA - OBP I/O Data to Memory (Bits 16-31/48-63)
ORA (Even and Odd Data Bits)
Maint. Channel DMA Return ITA
OZT Test Point Readout
HM1 ICE
Maint. Channel Return IYE
Sanity Code Return IYD
Error Logger IYF
D, E, F to MC Port on I/O
OYE Maint. Channel Return ZB11 IYE
OYD Sanity Code Return ZB11 IYD
OYF Error Logger ZB11 IYF
Sanity Code IYA
Maint. Channel IYB
Error Logger Return IYC
Sanity Code IYI
A, B, Ct to Shared Module
OYA Sanity Code IUA
OYB Maint Channel IUB
OYC Error Logger Return IUC
OYI Sanity Code IQA
Sanity Code Return IYL
Sanity Code IYQ
Maintenance Channel IYR
Error Logger IYS
L, Q, R, and S to I/O Module
OYL Sanity Code Return ZB11 IYL
OYQ Sanity Code ZB11 IYQ
OYR Maint Channel ZB11 IYR
OYS Error Logger ZB11 IYS
Sanity Code Return IYT
Maint Channel Return IYU
Error Logger IYV
T, U, V from I/O Module to HG0
OYT Sanity Code Return ISA
OYU Maint Channel Return ISB
OYV Error Logger ISC
Force Sanity Code Bypass IZR
OYX Sanity Code Detected
HB0 IMA
Successful Sanity Code Recognition from I/O Module
HF0 OIA-OIP
Command to I/O /16 IKA-IKP
OIA-OIP
Fanin of Maint Channel Return From CJ0 ONK
Maint Chan. Return ITI
HA0
IAA-IAT
(HG0) CPU Master Clear IJA
(HG0) CM Master Clear IKA
IBA-IBP
Fanout
OAA-OAP 4 Bits Each
IAF-IAI
OJA ICO,1 IRA
OKA ICO,1 IRE
CH0-CH15
HA1
IAA-IAT
(HG0) Maintenance Mode IJA
(HG0) Serial Maint Loop to HM IKA
IBA-IBP
Fanout
OBA-OBP 4 Bits Each
IAF-IAI
OJA ICO,1 ITA
OKA CH2 IVC
OLA Maint. Channel Return
ITD
HG0
Inputs
HD0 IJB
EZIF SHARED
ZB08
OLB I/O Interrupt
I/O Interrupt ILB
OLB
From Shared Module
(HG0) OUA-C
Sanity Code IUA
Maint Channel IUB
Error Logger Return IUC
From HG0 To Shared Module
Sanity Code Return IUE
Maint. Channel Return IUF
Error Logger IUG
OUC
OUB
OUA
From Shared Module
(HG0) ORA-C
Sanity Code Return IRA
Maint Channel Return IRB
Error Logger IRC
To Shared Module
Sanity Code IRA
Maint. Channel IRB
Error Logger IRC
ORA
ORB
ORC
From Shared Module to HG0

Figure 20. Maintenance Channel and Sanity Code Paths on Processor Module

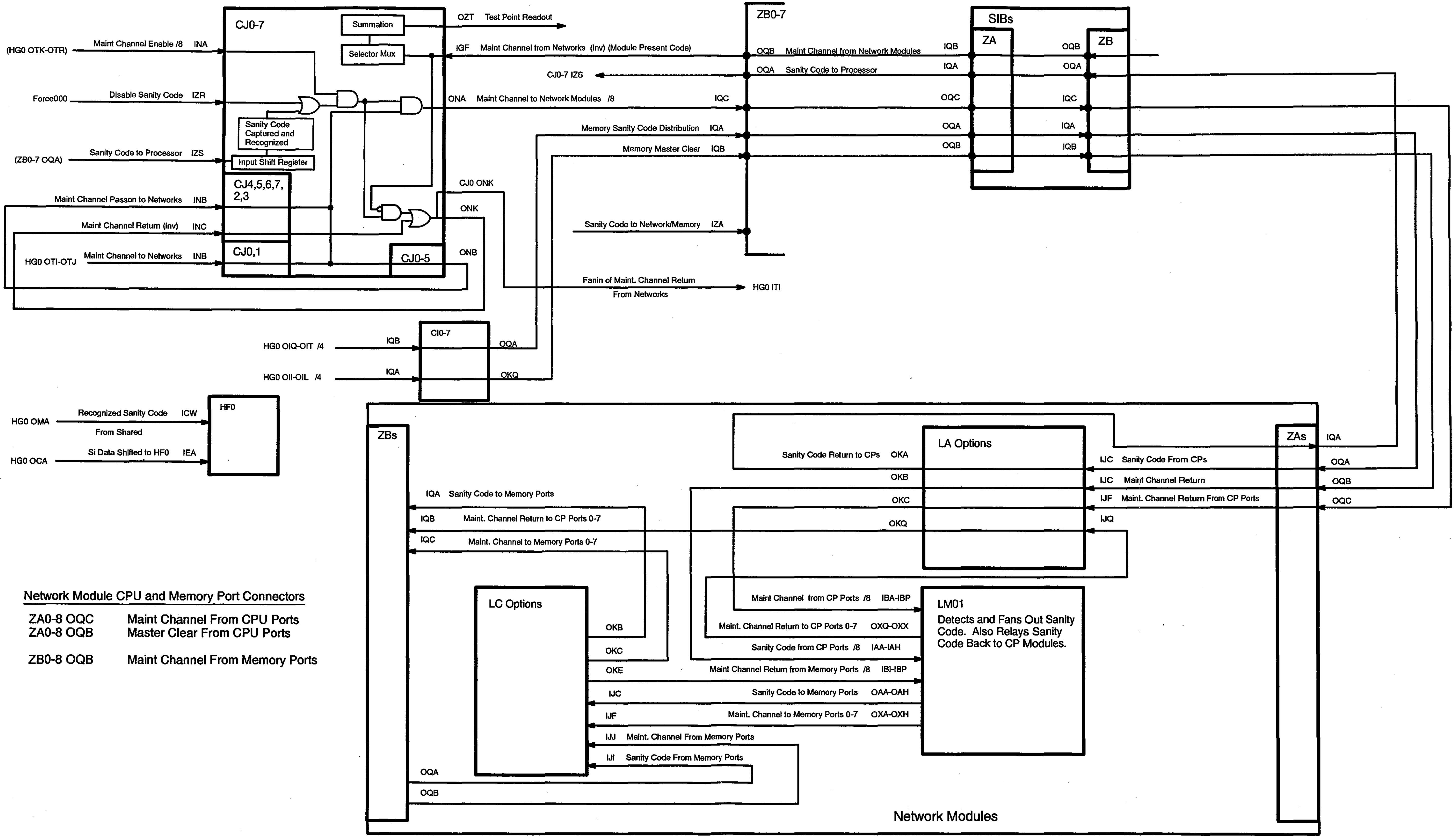

CJ0-7
Summation
Selector Mux
OZT Test Point Readout
IGF Maint Channel from Networks (inv) (Module Present Code)
(HG0 OTK-OTR)
Maint Channel Enable /8 INA
Force000
Disable Sanity Code IZR
Sanity Code Captured and Recognized
Input Shift Register
(ZB0-7 OQA)
Sanity Code to Processor IZS
CJ4,5,6,7, 2,3
Maint Channel Passon to Networks INB
Maint Channel Return (inv) INC
HG0 OTI-OTJ
Maint Channel to Networks INB
CJ0,1
CJ0-5
CJ0-7 IZS
ONA Maint Channel to Network Modules /8
IQC
Memory Sanity Code Distribution IQA
Memory Master Clear IQB
CJ0 ONK
ONK
ONB
Sanity Code to Network/Memory IZA
Fanin of Maint. Channel Return From Networks
HG0 ITI
ZB0-7
OQB Maint Channel from Network Modules
OQA Sanity Code to Processor
SIBs
ZA
ZB
IQB
IQA
OQC
OQA
OQB
OQB
OQA
IQC
IQA
IQB
HG0 OIQ-OIT /4
IQB
HG0 OII-OIL /4
IQA
CI0-7
OQA
OKQ
HG0 OMA
Recognized Sanity Code ICW
From Shared
HG0 OCA
Si Data Shifted to HF0 IEA
HF0
ZBs
ZAs
IQA
OQA
OQB
OQC
LA Options
Sanity Code Return to CPs OKA
OKB
OKC
OKQ
IJC Sanity Code From CPs
IJC Maint Channel Return
IJF Maint. Channel Return From CP Ports
IJQ
IQA Sanity Code to Memory Ports
IQB Maint. Channel Return to CP Ports 0-7
IQC Maint. Channel to Memory Ports 0-7
LC Options
OKB
OKC
OKE
IJC
IJF
IJJ Maint. Channel From Memory Ports
IJI Sanity Code From Memory Ports
OQA
OQB
Maint Channel from CP Ports /8 IBA-IBP
Maint. Channel Return to CP Ports 0-7 OXQ-OXX
Sanity Code from CP Ports /8 IAA-IAH
Maint Channel Return from Memory Ports /8 IBI-IBP
Sanity Code to Memory Ports OAA-OAH
Maint. Channel to Memory Ports 0-7 OXA-OXH
LM01
Detects and Fans Out Sanity Code. Also Relays Sanity Code Back to CP Modules.
Network Modules
Network Module CPU and Memory Port Connectors
ZA0-8 OQC Maint Channel From CPU Ports
ZA0-8 OQB Master Clear From CPU Ports
ZB0-8 OQB Maint Channel From Memory Ports

Figure 21. Maintenance Channel and Sanity Code Paths, CP-to-Network Modules

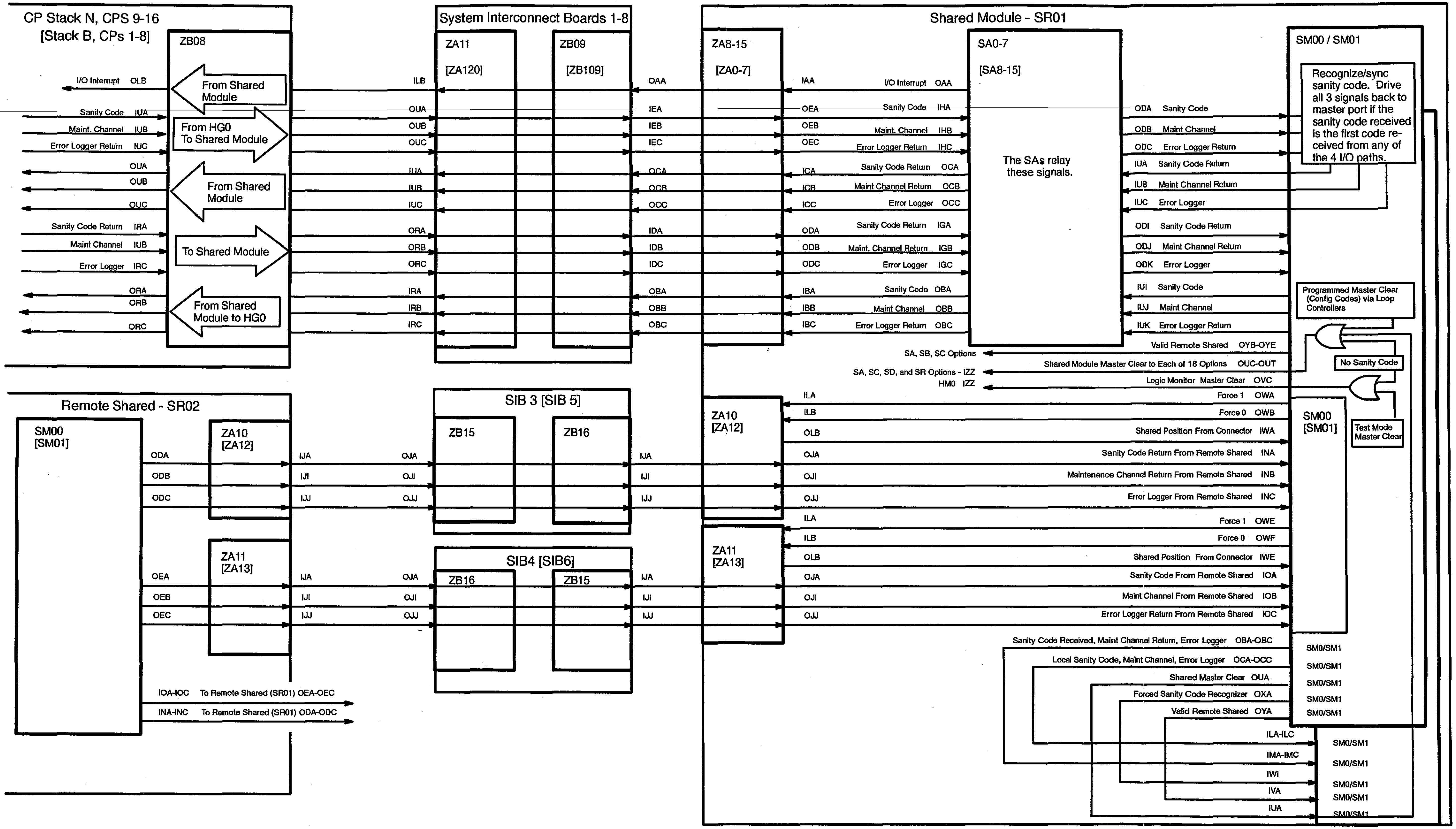


Figure 22. Maintenance Channel and Sanity Code Paths From CPU, Through SIB, to the Shared Module

Reader Comment Form

Title: Maintenance Channel
Preliminary Information

Number: HTM-xxx-x
December 1994

Your feedback on this publication will help us provide better documentation in the future. Please take a moment to answer the few questions below.

For what purpose did you primarily use this document?

______Troubleshooting
______Tutorial or introduction
______Reference information
______Classroom use
______Other - please explain ________________________________

Using a scale from 1 (poor) to 10 (excellent), please rate this document on the following criteria and explain your ratings:

______Accuracy ________________________________
______Organization ________________________________
______Readability ________________________________
______Physical qualities (binding, printing, page layout) ________________
______Amount of diagrams and photos ________________________
______Quality of diagrams and photos ________________________

Completeness (Check one)

______Too much information ________________________________

______Too little information ________________________________

______Just the right amount of information

Your comments help Hardware Publications and Training improve the quality and usefulness of your publications. Please use the space provided below to share your comments with us. When possible, please give specific page and paragraph references. We will respond to your comments in writing within 48 hours.

NAME ________________________________
JOB TITLE ____________________________
FIRM ________________________________
ADDRESS _____________________________
CITY ________________STATE________ZIP__________
DATE ________________________________
[or attach your business card]

CRAY
RESEARCH, INC.

CUT ALONG THIS LINE

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO 6184 ST. PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY RESEARCH, INC.

Attn: Hardware Publications and Training
890 Industrial Boulevard
Chippewa Falls, WI 54729

Fold

STAPLE