Maintenance Channel

(CRAY T90™ Series)

HTM-006-B

Cray Research Proprietary

Cray Research, Inc.

Maintenance Channel
-006-B

Record of Revision

REVISION	DESCRIPTION
	April 1995. Original printing.
A	June 1995. In revision A, quadrant and module stack numbers were changed in the chassis map to accommodate changes in the system net list. Corrections were also made to the sanity code description.
B	August 1995. In revision B, quadrant and module stack numbers were changed in the chassis map to accommodate changes in the system net list.

Any shipment to a country outside of the United States requires a letter of assurance from Cray Research, Inc.

Requests for copies of Cray Research, Inc. publications should be directed to:

CRAY RESEARCH, INC.
Logistics
6251 South Prairie View Road
Chippewa Falls, WI 54729

Comments about this publication should be directed to:

CRAY RESEARCH, INC.
Hardware Publications and Training
890 Industrial Blvd.
Chippewa Falls, WI 54729

MAINTENANCE CHANNEL

Figures

Tables

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 that describes 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 (11 × 17 pages) that show 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

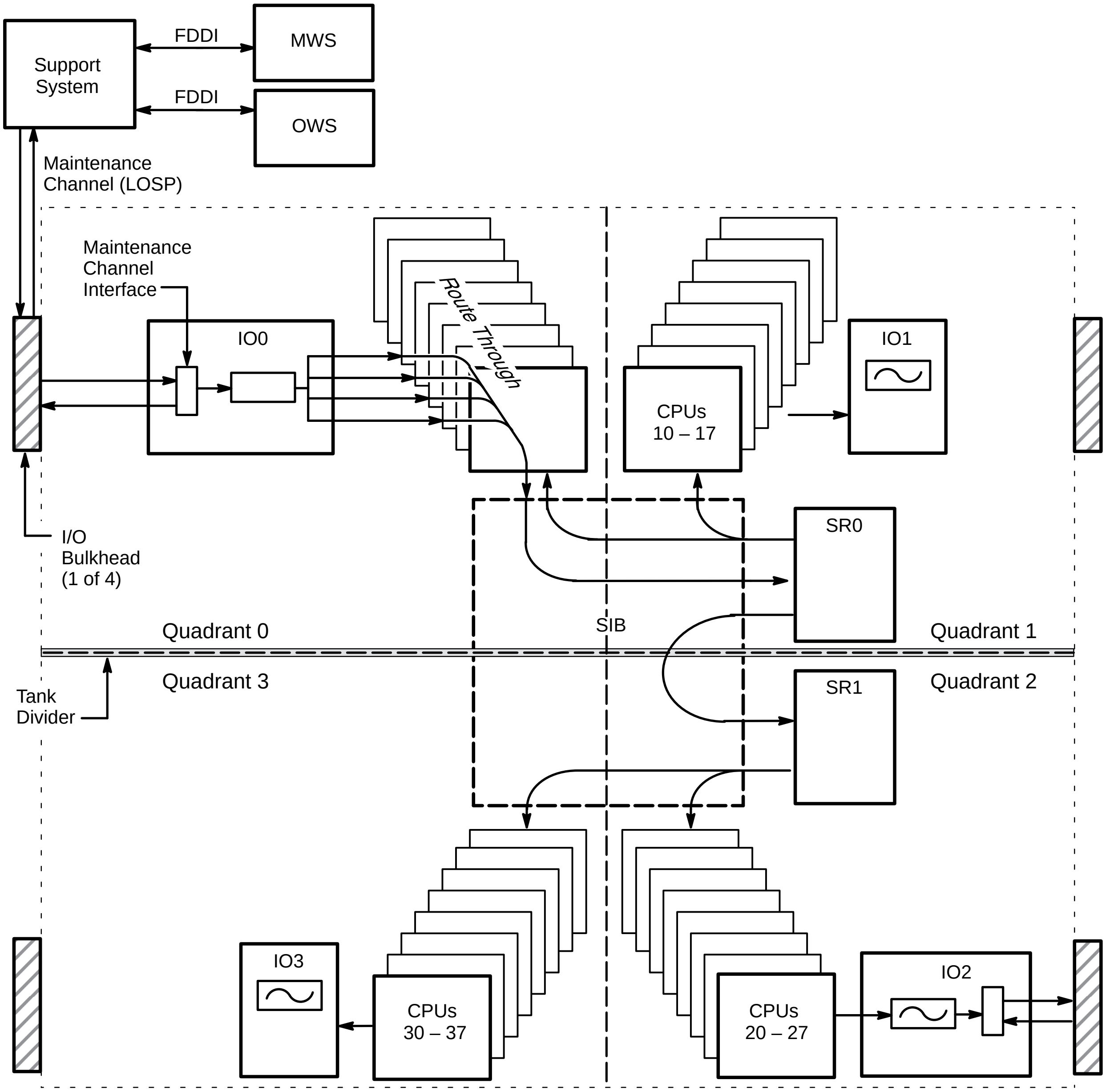
Each half of a CRAY T932 system has a separate and independent maintenance channel. (The CRAY T916 and CRAY T94 systems each have one maintenance channel.) The use of two maintenance channels enables the power-down of half of a CRAY T932 system while the other half is operating normally. Normal system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

The maintenance channel ports on two IO 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 IO module; however, this changes the system configuration.

The maintenance channel begins as a LOSP channel that is routed from the support system 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 IO modules, as shown in

Figure 1. Because the maintenance channel from the support system 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 serial signals that compose the maintenance channel (Figure 2) pass from the maintenance channel interface through the attached CP stack to the shared (SR) module. From the local SR module, the signals fan out to all modules in the system.

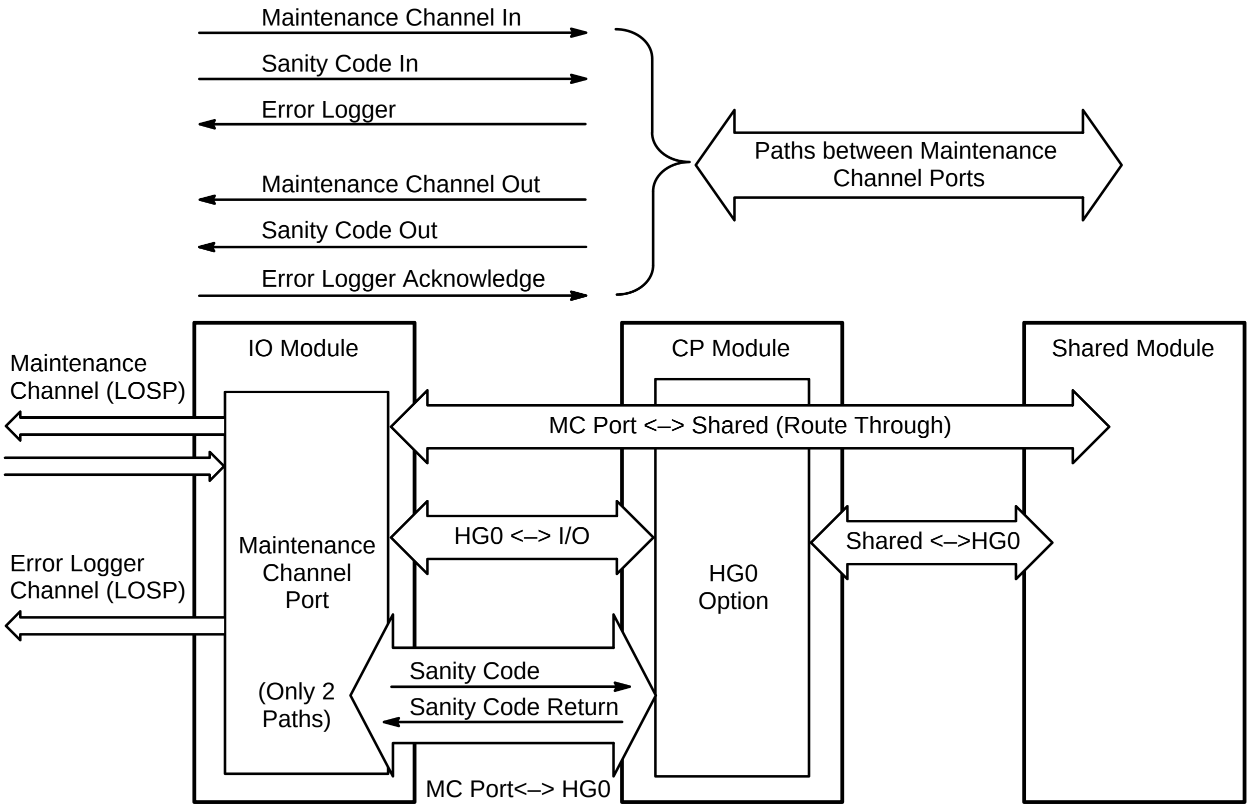
The SR modules send signals to other modules because there is no direct CPU-to-CPU communication; a CPU can access another CPU only through a SR module. The local SR module can access the remote SR module to communicate with modules in the other half of the system.

Maintenance Channel Signals

Starting at the maintenance channel interface (port) on the IO 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 is actually 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 responses from a module that has received an error from another module and another error can be sent.

Figure 2. Maintenance Channel Port Signal Paths



Sanity Code

Sanity code is a 6-bit pattern that controls 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 ensures 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 6 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.

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

Sanity Code Generator

A sanity code generator on an IO 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 can have two maintenance channels, each with its own sanity code generator. Note that logic for the sanity code generator resides on all IO modules in the system because all IO modules are the same.

A command is sent over the maintenance channel to initialize the sanity code generator on the IO 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.

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 establishes control of sanity codes and provides fan-out networks for sanity codes and maintenance, control, error, and configuration data. (Sanity codes travel on sanity code paths and sanity code return paths. Maintenance, control, and configuration data travel on maintenance channel paths 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 operator configures the actual path or structure of the sanity tree. 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; then sanity code can be sent to the SR module. This sanity code is routed through one of the four CP modules connected to the IO module, as determined by the configuration. Once the tree is established, configuration information can 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 logger information is returned along this tree. The initial sanity tree structure is recorded and saved by SCE 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 from one module to another. The direct portion of a sanity tree includes CP, IO, and SR modules.

Indirect portions of a sanity tree include memory modules and network modules. 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.

Sanity Tree Configuration through Shared Modules

A sanity tree should be configured so that the next node in the tree after the maintenance port on the IO module is the local SR module. This is necessary because no direct communication occurs between CP modules. A CP module must go through a SR module to access another CP module.

System Configuration and Sanity Trees

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 that sends 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 and CRAY T916 systems and from a CP module in CRAY T94 systems.)

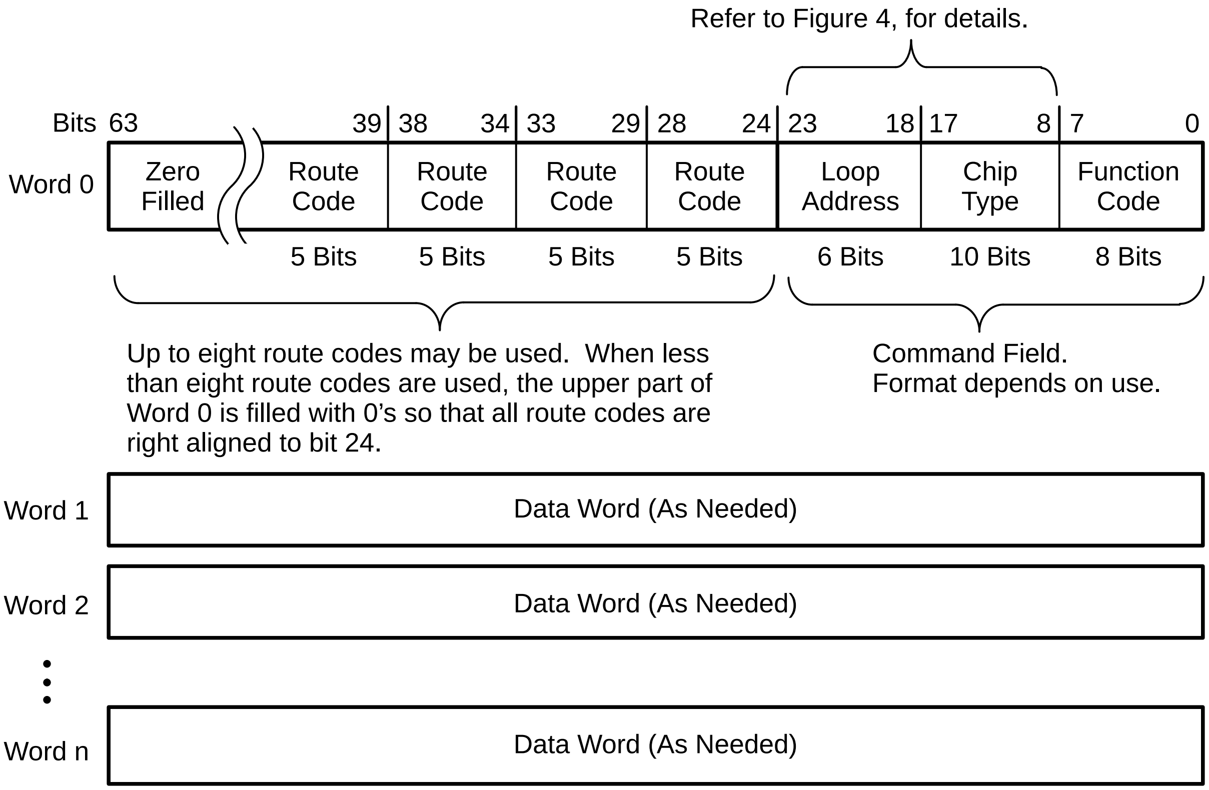
Sanity codes from CP to memory modules are controlled by memory configuration codes 70 and 71. There are eight sanity code paths from a CPU into the memory system; one for each of the eight memory section paths.

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, and network modules can receive commands to select test points.

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.

Figure 3. Maintenance Channel Command Word Format



Refer to Figure 4, for details.

Up to eight route codes may be used. When less than eight route codes are used, the upper part of Word 0 is filled with 0’s so that all route codes are right aligned to bit 24.

Command Field. Format depends on use.

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 path.

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 that follows 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-005-0, for detailed information.

Table 1. Boundary Scan 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

IO 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 SR module route codes are relative to the IO module. For example, CP 0 from an IO 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. IO 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	IO
34	Logic monitor
35	DMA port
37	Continuation

Shared Module Route Codes

Table 4 lists two levels of route codes for SR modules. The 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 was 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

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

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

All the route codes from the master IO module to the final module of a multi-module path must follow the sanity code-established path.

Function Codes

The function code field selects a specific function on an option. There can be up to 256 (400_8) function codes, which are grouped as follows:

- Configuration codes ($0 – 77_8$)
- Maintenance mode functions ($100 – 177_8$)
- Test point selections ($200 – 377_8$)

NOTE: Refer to the *Triton Maintenance System Engineering Note*, Engineering specification number PRN-0957, for detailed descriptions of each function code; the appendix section of this note includes tables that list test points for each option. Configuration and function codes that are not shown in the tables are generally ignored, but should not be used.

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 . Table 5 through Table 8 list the configuration codes for the CP, IO, and SR 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

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. IO Module Configuration Codes

Code	Group	Function
10	Support channel path	Set path bit 0
11		Set path bit 1
12 †		Clear path bits 0 and 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 bit 0
23 †		Clear memory group bit 0
24		Set memory group bit 1
25 †		Clear memory group bit 1
26	I/O access	Set I/O group bit 0; LA, LB, SU, and VH pseudo-options (LOSP and VHISP)
27 †		Clear code 26
30		Set I/O group bit 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 cluster 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 IO modules to ensure they are functioning properly. Table 9 lists function codes that are useful in testing areas that are difficult to test with 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 it issues maintenance functions to prevent a system interrupt.

Before a CPU can issue any maintenance mode functions, it must receive a 176 function code to place the CPU into maintenance mode. 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 CPU. 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 correction
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 storage	163		
124	Inst stack	Sel buffer bit 0 for storage	164	Memory in ports	4-clock-period resume delay
125	Inst stack	Sel buffer bit 1 for storage	165	Memory in ports	16-clock-period resume delay
126	Inst stack	Sel buffer bit 2 for storage	166	Memory in ports	63-clock-period resume delay
127			167		
130	Cache	Do not use	170	Memory error correction	1771*jk* 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 176i*jk* instructions, store checkbyte in V*i* 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 is a logical grouping of options on a module. A loop controller serves as a central fan-out point on the module for all the possible 256 (400_8) function codes:

- Configuration codes
- Maintenance mode functions
- Test-point selections

Loop controller logic is contained in the following options:

- HG option on CP modules
- DM option on IO 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, IO, and SR modules. Test-point information and control data from various options are routed to the logic monitor (HM0 and HM1 options) on the respective CP, IO, or SR module. A description of the logic monitor begins on page 30.

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 SR modules, 32 on IO 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.

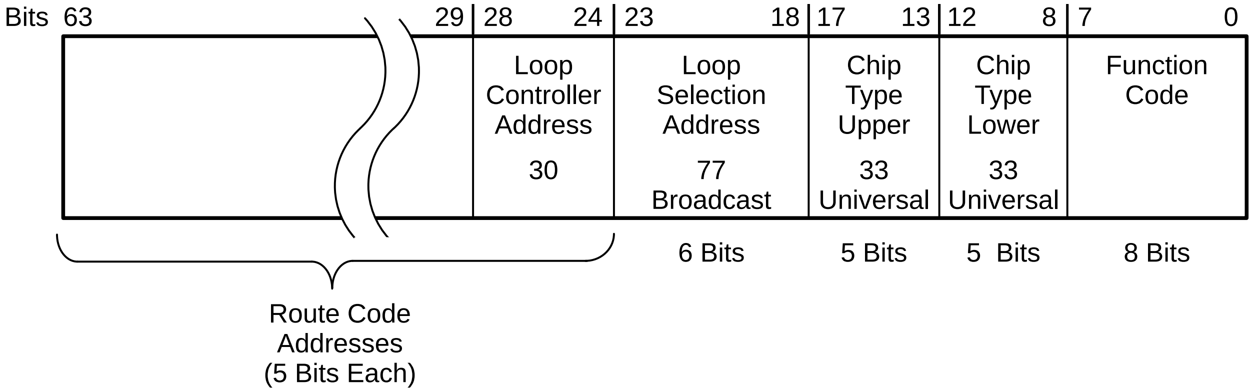
- The chip type field can specify an individual option or all options on the loop (all chips = 33_8). Each option on the loop has its own separate input path. That is, each loop has a separate output (pin) for each option on the loop.

- Two options of the same type are never on the same loop.
- Function codes are never used twice for one module type; they are all unique.

Loop Controller Command Word Format

All function codes are sent to loop controllers in a maintenance channel command word. Figure 4 shows bits 0 through 28 of the maintenance channel command word. Bits 24 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 goes to the loop controller.

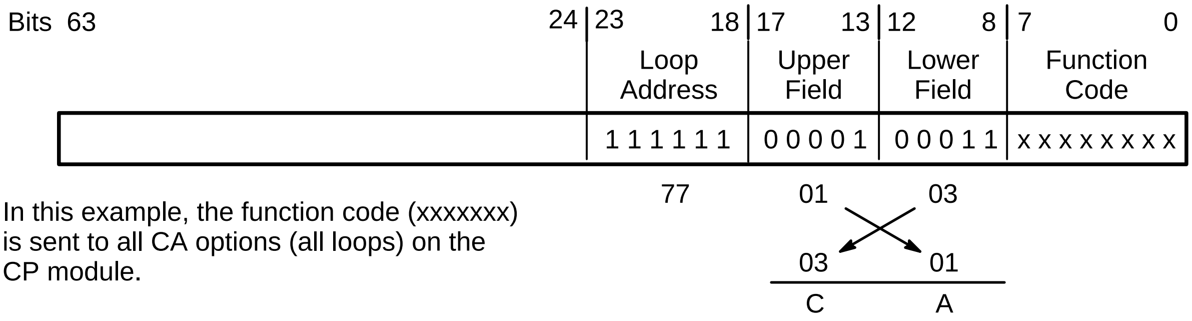
The loop-selection address defines which loop or set of loops that leaves the loop controller will distribute the function code. Only values of 0 through 37_8 and 77_8 are valid for IO modules, and 0 through 17_8 and 77_8 are valid for CP, SR, and network modules in the loop-selection address. Other values are undefined.

Chip Type Fields

The 5 bits in each of the upper and lower chip type fields define the letters of the option. The letter A is encoded as 1, B as 2, and so on; Z is encoded as 32. Note that the upper and lower option 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 to 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. Chip (option) types with the same letter designators cannot reside on the same loop because of the routing scheme that is used in bits 23 through 8.

Figure 5. Example Chip Type Address



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

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

CP Module Loop Controller

The HG option on each CP module contains the loop controller logic as shown in Figure 6. Table 10 shows the option assignments to each loop. Each of the 16 HG loop outputs (OHA through 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

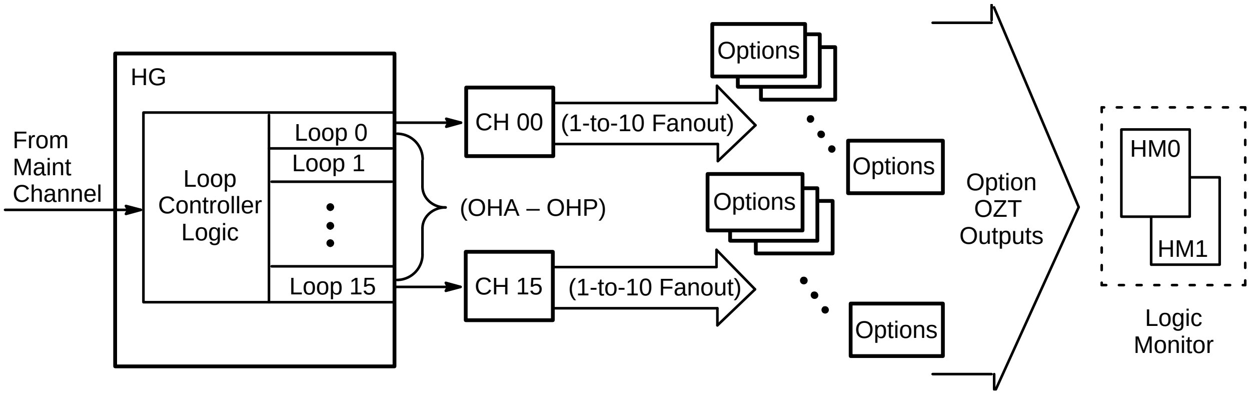


Table 10. CP Module Loop Controller Option Assignments

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

IO Module Loop Controller

Each IO 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 IO module, except for the DR options. Each quadrant contains 1 VHISP, 2 HISP, 2 LOSP channels, and routing for the LOSPX support and MCUI channels. All I/O data in a quadrant is routed through a connected CP module. Figure 7 shows the loop option assignments; Table 11 lists these assignments.

Figure 7. IO Module Loop Controller

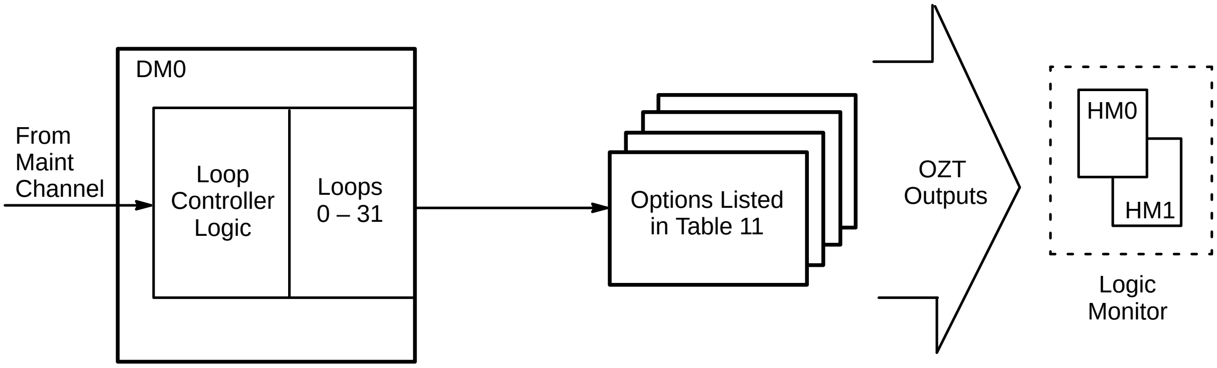


Table 11. IO Module Loop Option Assignments

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 SR modules is contained on the SM00 and SM01 options as shown in Figure 8. Table 12 lists the options on each loop.

Figure 8. Shared Module Loop Controllers

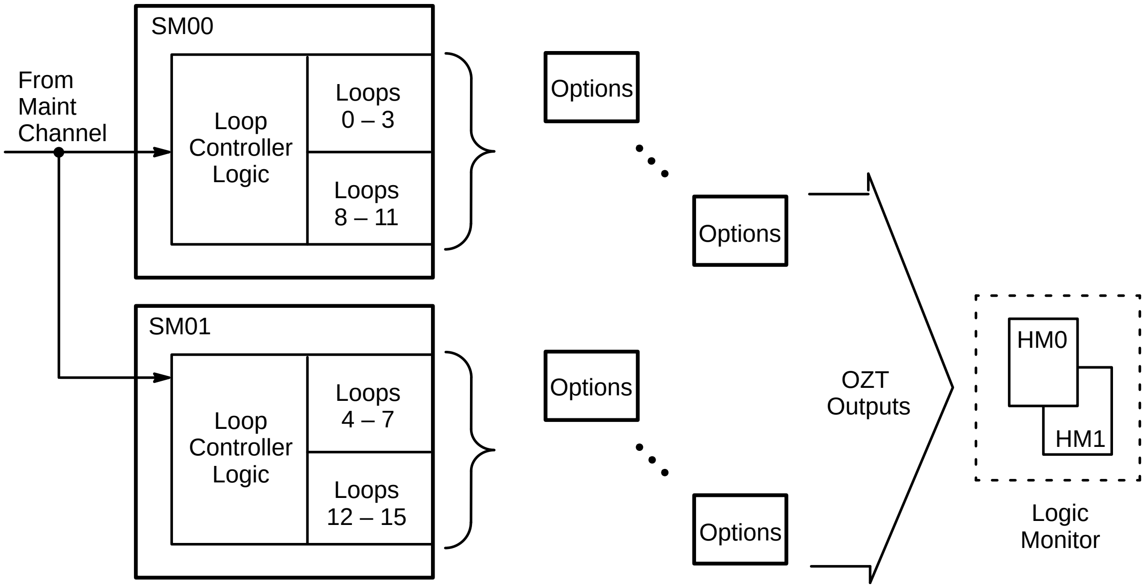


Table 12. Shared Module Loop Option Assignments

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

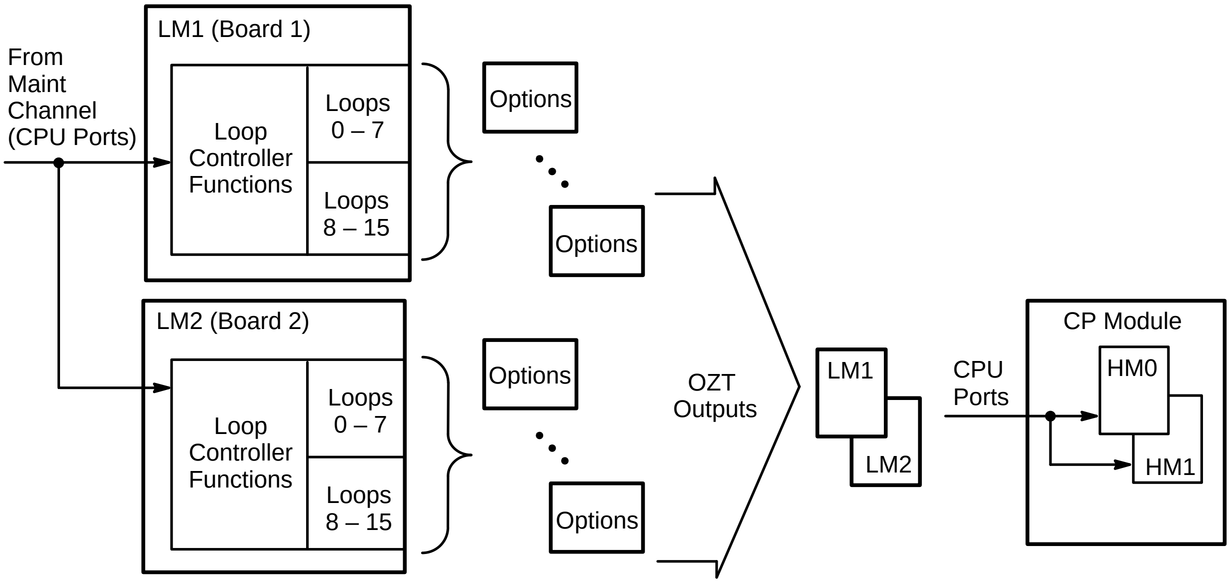


Table 13. Network Module Loop Option Assignments

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

During system checkout, the boundary scan module can be connected directly to a CP module when no IO module is present. The logic on the boundary scan module provides maintenance channel functions. This procedure is not used in the field. Table 14 lists the function codes that are utilized by the boundary scan module.

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 module and a SR module use the primary sanity code path. STCO systems that do not have a SR 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 you send any maintenance channel function that will return data, you must disable the error logger with a 61 code. This is necessary because the system cannot distinguish a possible error logger data word from a requested maintenance channel word. When the error logger is disabled, a channel disconnect is returned; normal error logger data does not return disconnects.

Logic Monitor

The logic monitor selects and stores test-point data and control information from various logic chips to indicate the state of operation of various modules. The logic monitor can also 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 options on each CP, IO, and SR 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.

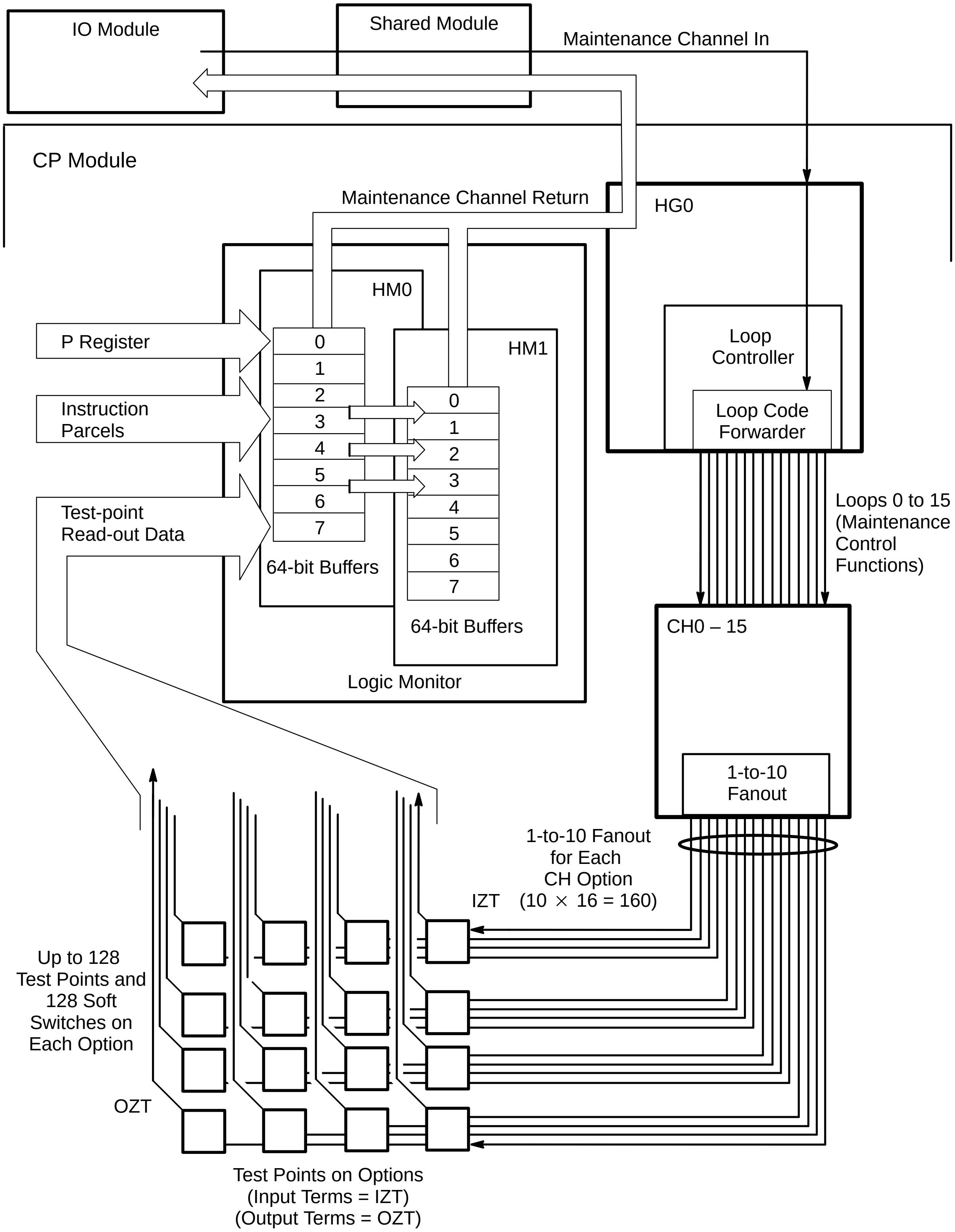
The logic monitor can also be used to monitor the CPU(s) as instructions are issuing. The logic monitor can start and stop recording on specific events that occur while instructions are issuing. This allows you to monitor CPU(s) as instructions issue without affecting the CPUs. You can then use the logic 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 that select the test points to be read are sent to the loop controller. The loop controller sends them to the options. Test-point read-out data is read by the logic monitor and is sent through 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 *LME User Guide* describes how to use LME to operate and control the logic monitors.

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


IO Module
Shared Module
Maintenance Channel In
CP Module
Maintenance Channel Return
HG0
HM0
HM1
P Register
Instruction Parcels
Test-point Read-out 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 × 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)

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.

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.

Test points are selected through the maintenance channel with 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 you cannot do 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.

Master Clear Levels

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

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

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.

Configuration codes routed to CP modules control the passing of sanity codes to modules that do not receive sanity code enable commands. SR 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 the 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 IO 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 sanity tree.

Memory Master Clear

This is the second highest level of master clear. When this function is activated from a CP module, it 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 in a master clear state, the CPU reference will hang until the master clear state is cleared or released.

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

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 run to completion. The memory access master clear function does not affect the memory modules.

A reset configuration code (77) or a memory master clear on code (60) 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 of the master clear functions. This function halts the CPU and, upon restart, causes the CPU to do an initial exchange sequence. The CPU master clear does not destroy A and S register values, so a drop of CPU master clear causes an exchange to the initial exchange package.

When the CPU master clear is activated, all CPU maintenance codes, 100 through 177, are cleared. The CPU master clear does not clear any of the CPU configuration codes (00 through 77).

A reset configuration code (77), a memory master clear on code (60), or a memory access master clear on code (62) 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).

IO Quadrant Master Clear

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

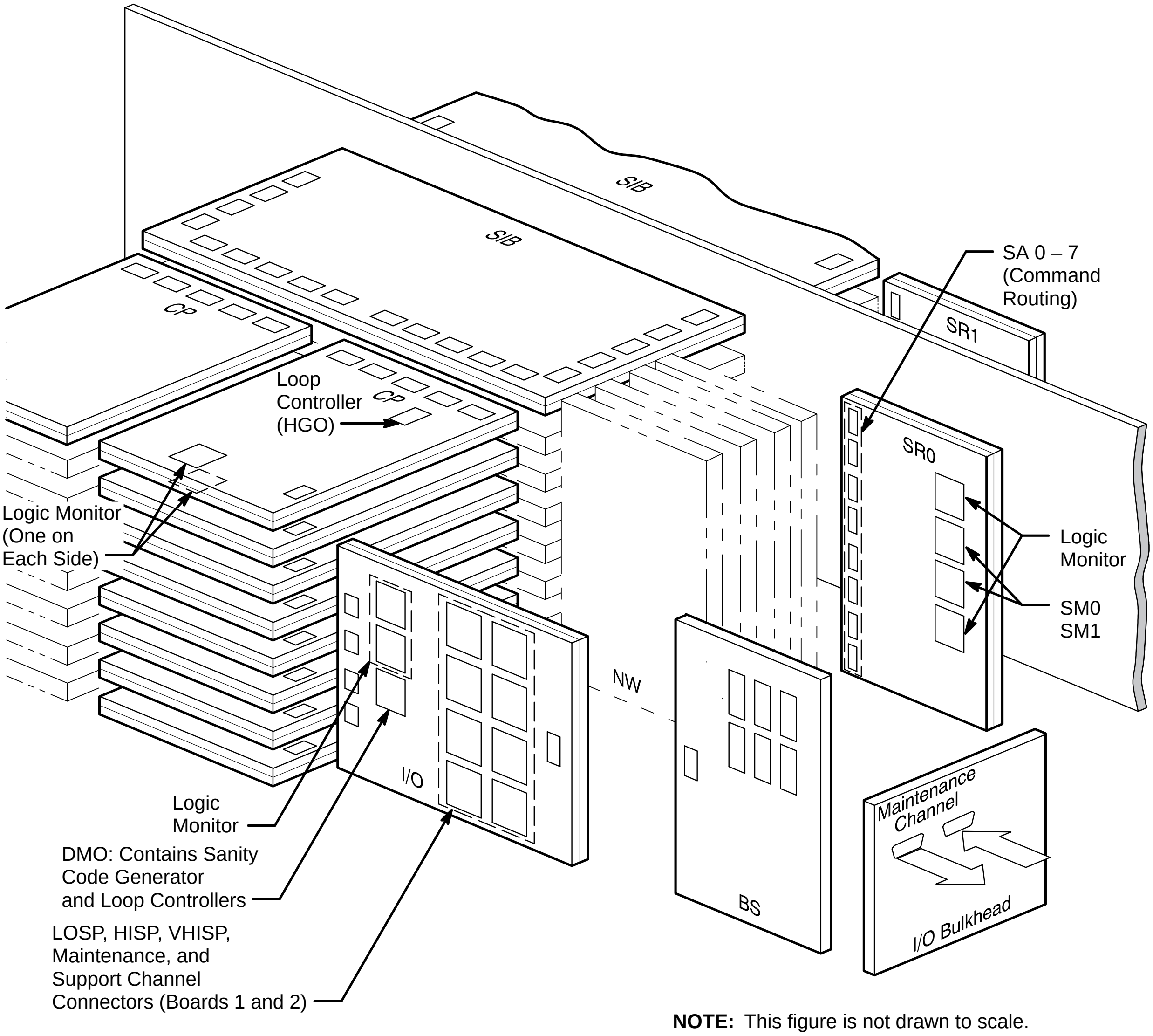
Shared Master Clear

This signal clears all the logic on the SR module. This signal should not be needed very often because the flush command from the CPU clears all logic on the SR module associated with a particular CPU.

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 11 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 11 × 17 in. block diagrams at the end of this document for more detail.

Figure 11. Related Maintenance Channel Components

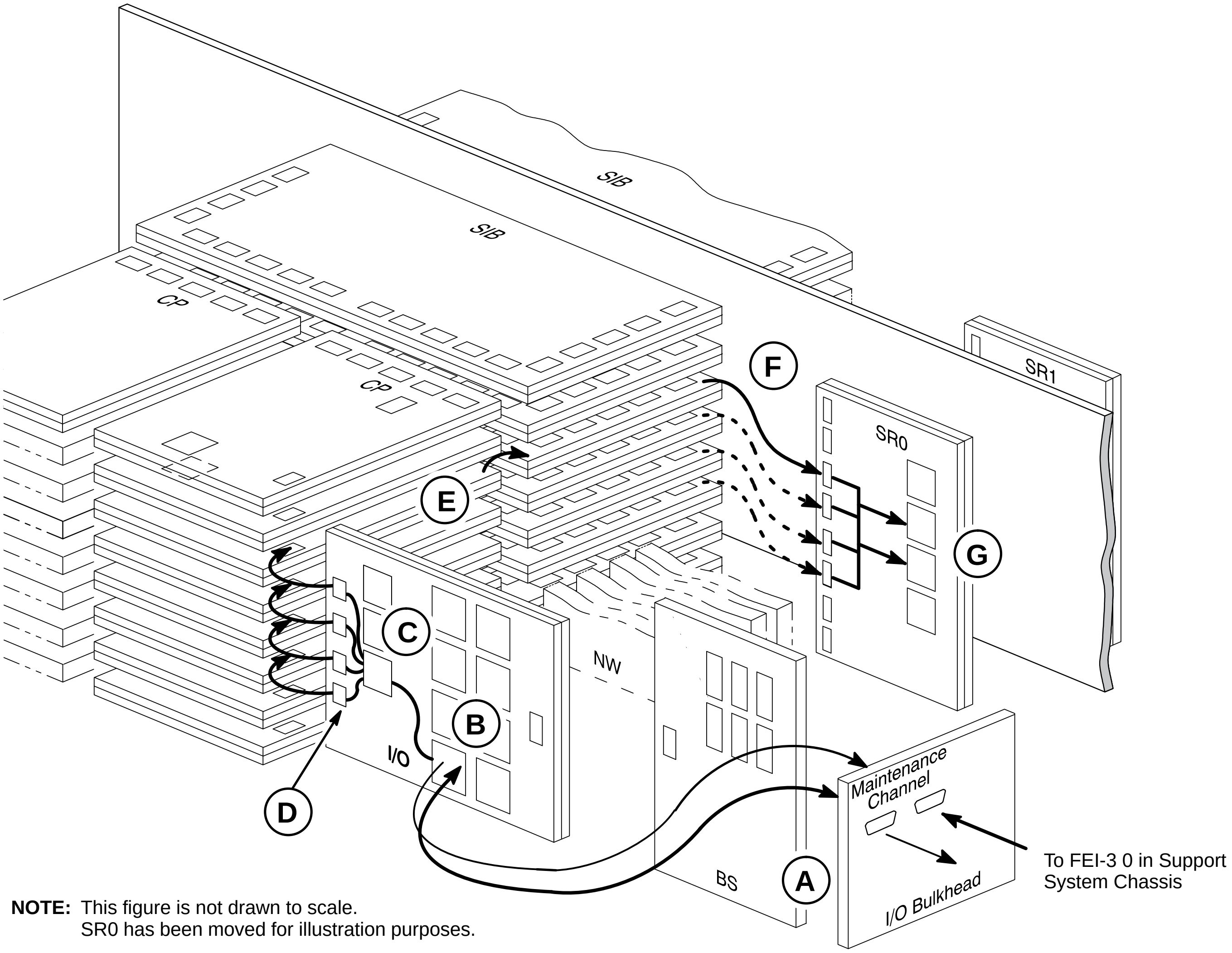


NOTE: This figure is not drawn to scale. SR0 has been moved for illustration purposes.

Initial Sanity Tree Configuration

The following process establishes an initial sanity tree. The letters in text correspond to the circled letters in Figure 12.

Figure 12. Initial Sanity Tree Path to Shared Module



To FEI-3 0 in Support System Chassis

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

A. The LOSPX maintenance channel leaving the support system chassis connects to the maintenance channel port on the I/O bulkhead.

B. The maintenance channel connects to a LOSP I/O channel connector on the IO module. The maintenance channel then enters a channel controller (DD0 option). 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 before it replies to the channel controller on the IO module (DD option). A disconnect signal (route code 35) or an attention signal from the channel controller takes 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 IO module has not received any sanity code at this time.

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

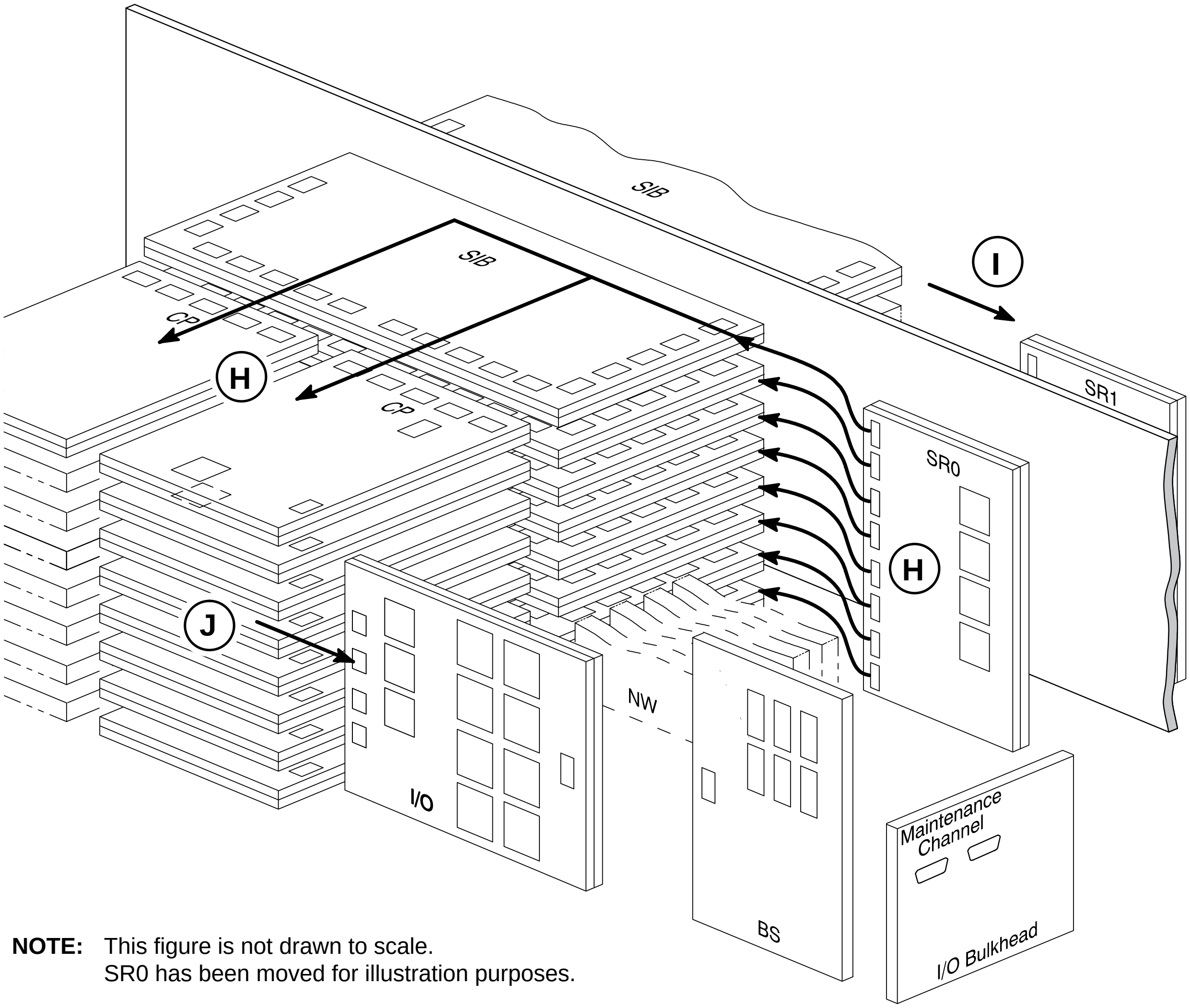
E. Any of the DC options and CP modules can be used; however, the System Configuration Environment (SCE) program will normally try DC0 first. If this path is not successful, the next DC option/CP module is used. Note that this sanity code is routed through the CP module and that the CP module does not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

F. The sanity code enters the local SR 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 SR module is routed to the SM0 and SM1 options.

Refer to Figure 13 for the remaining processes of the initial sanity tree configuration process.

Figure 13. Initial Sanity Tree Path from Local Shared Module



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

H. The SR 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 IO modules on the other side of the system are to be included in the sanity tree, the IO module then sends sanity code to the remote SR module. The remote SR 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 IO 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 another; there is a sanity code arbitrator associated with each module that performs a first-come, first-serve arbitration.

You should note that on a large system, an SR module should serve as a central fan-out point for sanity code after the IO module. This centralized fan-out point can be used because both of the SR modules are connected. By using both SR modules, one distribution level should be removed from the tree structure.

Swapping Sanity Code Generators

In a system with more than one IO module, only one IO module can be the master sanity code generator. It is possible to designate a master sanity code generator on a different IO 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 IO module, the sanity code generator on a second IO module is started. Using the maintenance channel on the first IO 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 IO 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 IO module.

The sanity code generator on the first IO 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 logger 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 49 and 50 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 IO module to the SR 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 that follows a 37 route code can be captured, even though the data words may be part of a continuation sequence or a starting address. The capability to capture every data word that follows a 37 route code is provided to enable the desired data word to eventually be captured. The HB option signals the HC option when to send 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 from the HG option to both the HB and the HC options 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:

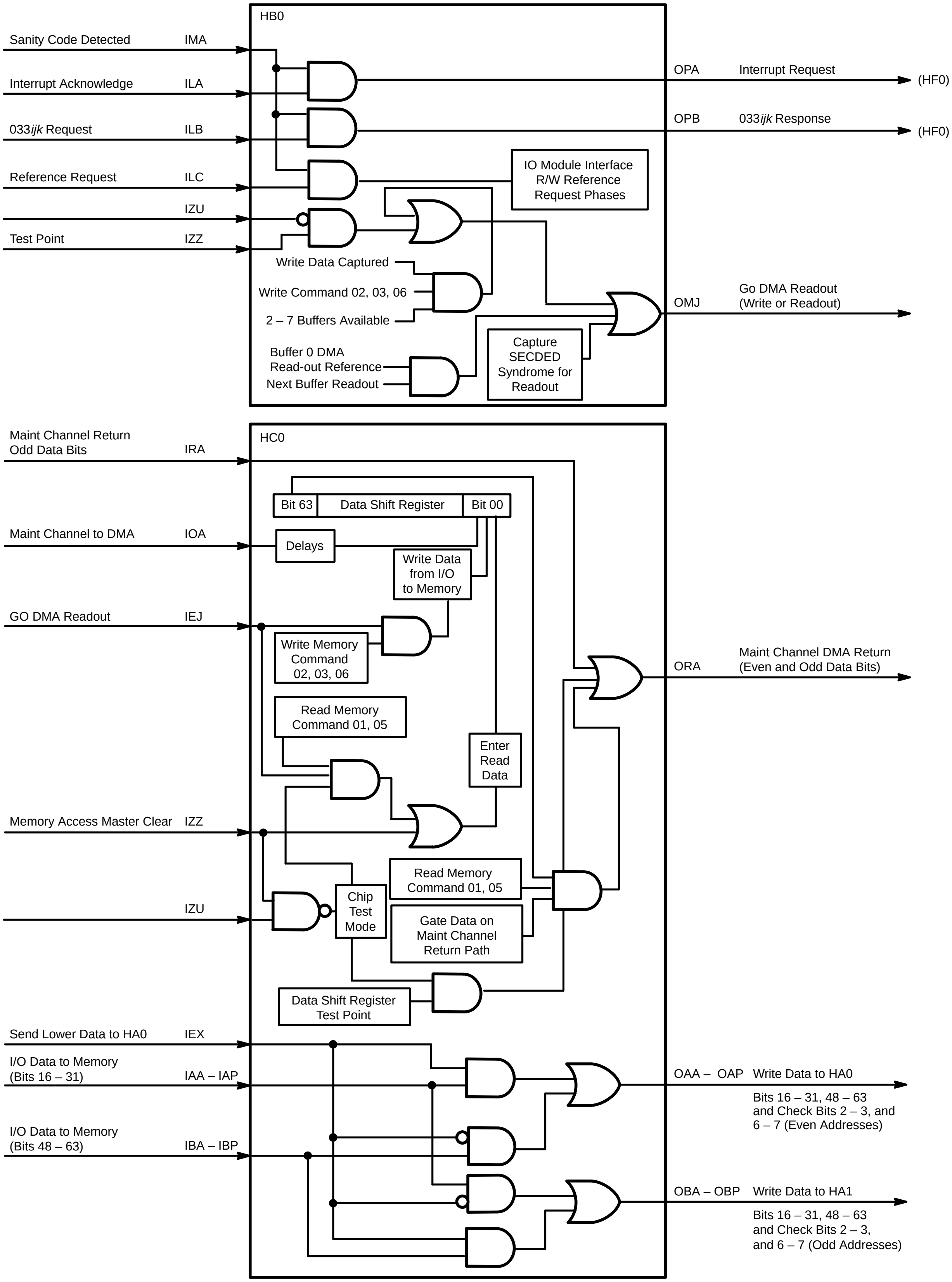
- Write Memory command. Data is transferred into main memory using the I/O memory port.

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

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

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

Figure 14. Data Paths through HB and HC Options


HB0
Sanity Code Detected
IMA
Interrupt Acknowledge
ILA
033*ijk* Request
ILB
Reference Request
ILC
IZU
Test Point
IZZ
OPA
Interrupt Request
(HF0)
OPB
033*ijk* Response
(HF0)
IO Module Interface R/W Reference Request Phases
Write Data Captured
Write Command 02, 03, 06
2 – 7 Buffers Available
Buffer 0 DMA Read-out 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
ORA
Maint Channel DMA Return (Even and Odd Data Bits)
Read Memory Command 01, 05
Enter Read Data
Memory Access Master Clear
IZZ
Read Memory Command 01, 05
IZU
Chip Test Mode
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
OAA – OAP
Write Data to HA0
Bits 16 – 31, 48 – 63 and Check Bits 2 – 3, and 6 – 7 (Even Addresses)
I/O Data to Memory (Bits 48 – 63)
IBA – IBP
OBA – OBP
Write Data to HA1
Bits 16 – 31, 48 – 63 and Check Bits 2 – 3, and 6 – 7 (Odd Addresses)

HG Option

The HG option provides the following maintenance functions for the CP module. Also refer to Figure 15 and Figure 16 for more information.

- Sanity code recognizers and the home port arbiter for first sanity code detection.
- Loop forwarders 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 are forwarded to the IO module. Logic monitor functions are also forwarded from the HG option to the IO module.
- Maintenance channel return messages are forwarded from this option.
- Error logger channel functions for the CP module are performed on this option.

Figure 15. Maintenance Function on the HG Option

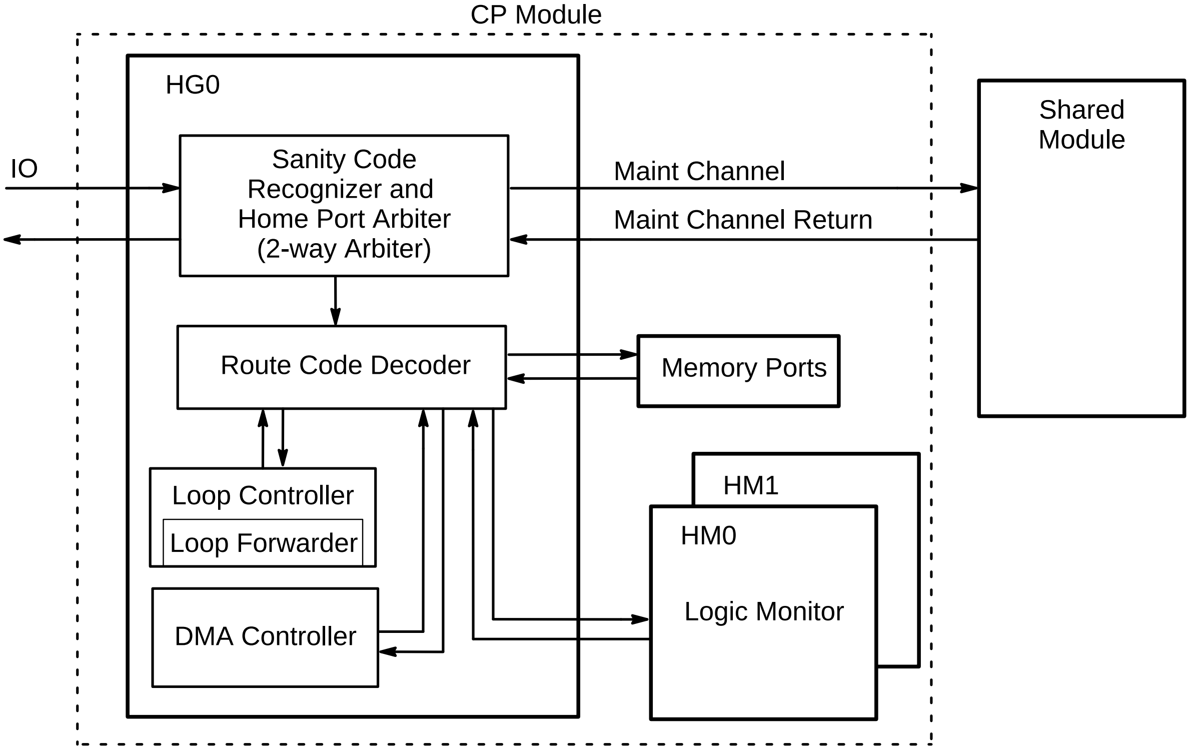
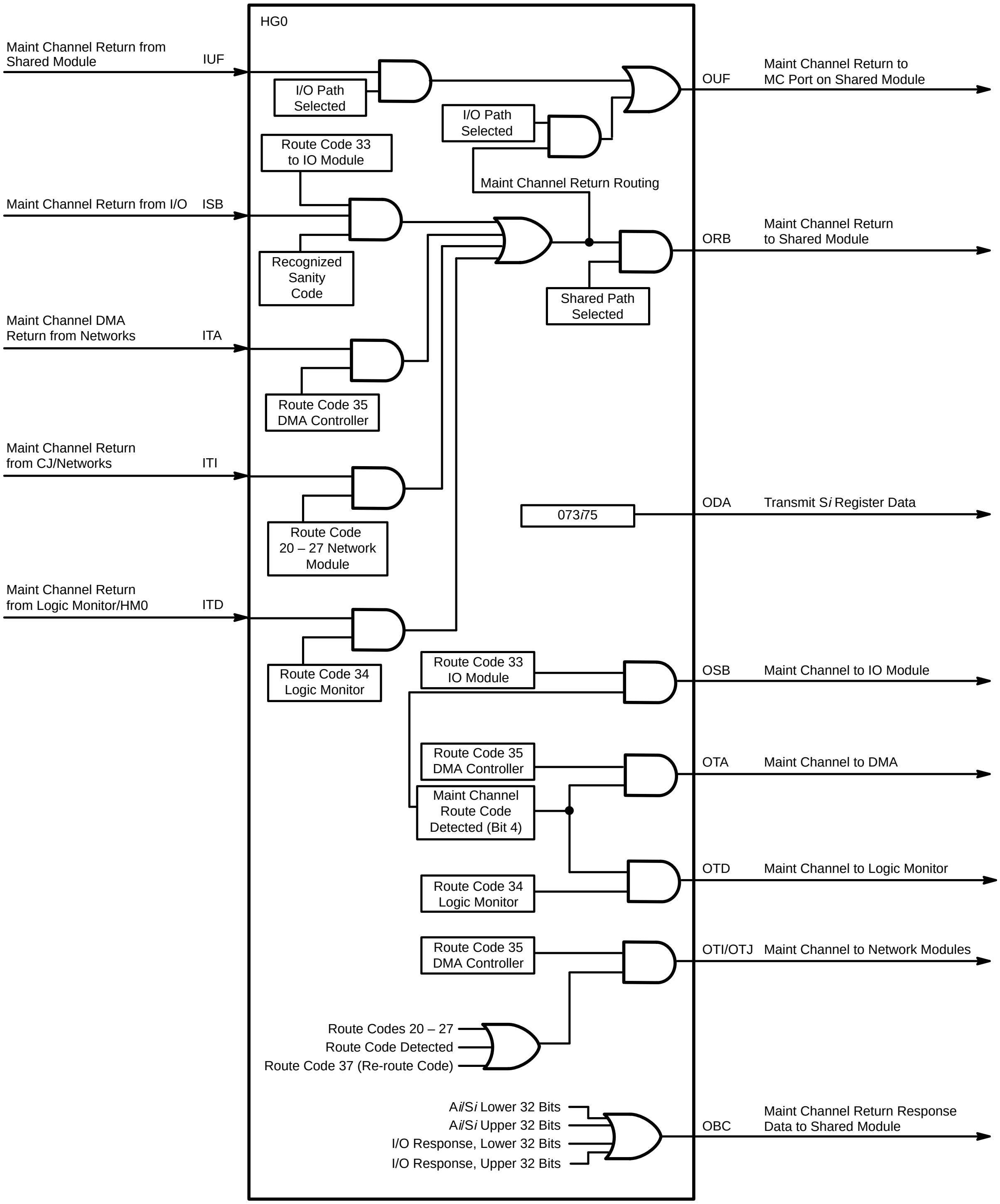


Figure 16. Maintenance Channel Paths from HG0 Option


HG0
Maint Channel Return from Shared Module
IUF
I/O Path Selected
I/O Path Selected
Route Code 33 to IO Module
Maint Channel Return Routing
Maint Channel Return from I/O
ISB
Recognized Sanity Code
Shared Path Selected
Maint Channel DMA Return from Networks
ITA
Route Code 35 DMA Controller
Maint Channel Return from CJ/Networks
ITI
Route Code 20 – 27 Network Module
073*i*75
Maint Channel Return from Logic Monitor/HM0
ITD
Route Code 34 Logic Monitor
Route Code 33 IO Module
Route Code 35 DMA Controller
Maint Channel Route Code Detected (Bit 4)
Route Code 34 Logic Monitor
Route Code 35 DMA Controller
Route Codes 20 – 27
Route Code Detected
Route Code 37 (Re-route Code)
A*i*/S*i* Lower 32 Bits
A*i*/S*i* Upper 32 Bits
I/O Response, Lower 32 Bits
I/O Response, Upper 32 Bits
OUF
Maint Channel Return to MC Port on Shared Module
ORB
Maint Channel Return to Shared Module
ODA
Transmit S*i* Register Data
OSB
Maint Channel to IO Module
OTA
Maint Channel to DMA
OTD
Maint Channel to Logic Monitor
OTI/OTJ
Maint Channel to Network Modules
OBC
Maint Channel Return Response Data to Shared Module

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 options may receive sanity codes from IO 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. IO modules have priority over remote shared modules, and SM0 inputs have priority over SM1 inputs.

Sanity code is returned to any path that supplies 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 it is 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 74 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 74 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 option. Using a configuration mode, an alternate path between SM1 options 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 Shared Module Position 0 and Position 1

Two inputs per SM option are strapped (hardwired) to the SIB to determine whether the shared module is in position 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 21, on page 51. 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 options ensure that both SM options on the same module each recognize the module position of the other, which 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.

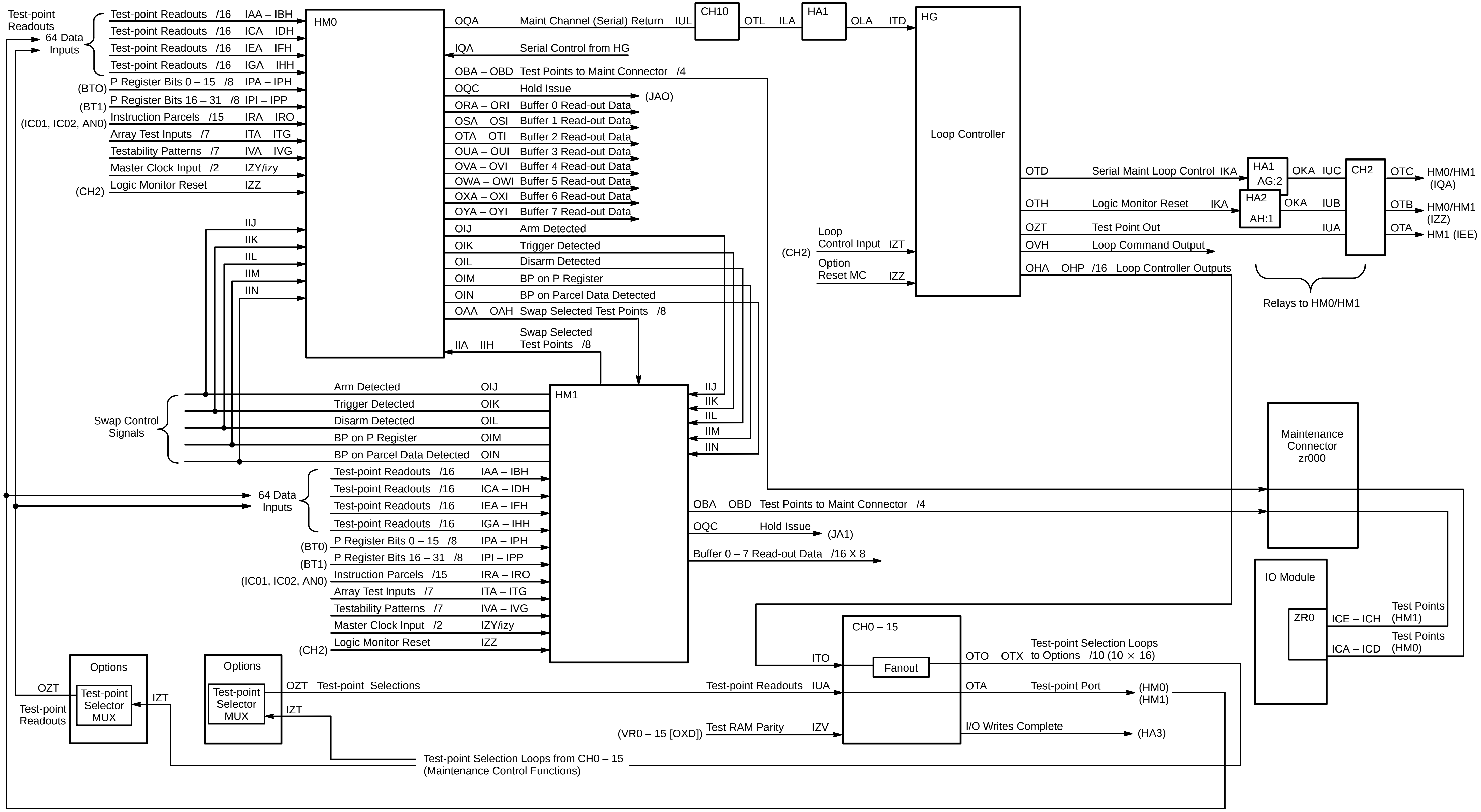

Test-point Readouts
64 Data Inputs
Test-point Readouts /16 IAA – IBH
Test-point Readouts /16 ICA – IDH
Test-point Readouts /16 IEA – IFH
Test-point Readouts /16 IGA – IHH
(BTO) P Register Bits 0 – 15 /8 IPA – IPH
(BT1) P Register Bits 16 – 31 /8 IPI – IPP
(IC01, IC02, AN0) Instruction Parcels /15 IRA – 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 Test Points to Maint Connector /4
OQC Hold Issue
(JAO)
ORA – ORI Buffer 0 Read-out Data
OSA – OSI Buffer 1 Read-out Data
OTA – OTI Buffer 2 Read-out Data
OUA – OUI Buffer 3 Read-out Data
OVA – OVI Buffer 4 Read-out Data
OWA – OWI Buffer 5 Read-out Data
OXA – OXI Buffer 6 Read-out Data
OYA – OYI Buffer 7 Read-out 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 Test Points /8
Swap Selected Test Points /8
IIA – IIH
CH10
OTL ILA
HA1
OLA ITD
HG
Loop Controller
OTD Serial Maint Loop Control IKA
HA1 AG:2
OKA IUC
CH2
OTC
HM0/HM1 (IQA)
OTH Logic Monitor Reset IKA
HA2 AH:1
OKA IUB
OTB
HM0/HM1 (IZZ)
OZT Test Point Out IUA
OTA
HM1 (IEE)
OVH Loop Command Output
OHA – OHP /16 Loop Controller Outputs
Relays to HM0/HM1
(CH2) Loop Control Input IZT
Option Reset MC IZZ
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
Test-point Readouts /16 IAA – IBH
Test-point Readouts /16 ICA – IDH
Test-point Readouts /16 IEA – IFH
Test-point 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 Test Points to Maint Connector /4
OQC Hold Issue
(JA1)
Buffer 0 – 7 Read-out Data /16 X 8
Maintenance Connector zr000
IO Module
ZR0
ICE – ICH
Test Points (HM1)
ICA – ICD
Test Points (HM0)
CH0 – 15
ITO
Fanout
OTO – OTX
Test-point Selection Loops to Options /10 (10 × 16)
Test-point Readouts IUA
OTA Test-point Port
(HM0) (HM1)
(VR0 – 15 [OXD]) Test RAM Parity IZV
I/O Writes Complete
(HA3)
Options
Test-point Selector MUX
OZT
Test-point Readouts
IZT
Options
Test-point Selector MUX
OZT Test-point Selections
IZT
Test-point Selection Loops from CH0 – 15 (Maintenance Control Functions)

Figure 17. CP Module Logic Monitor Block Diagram

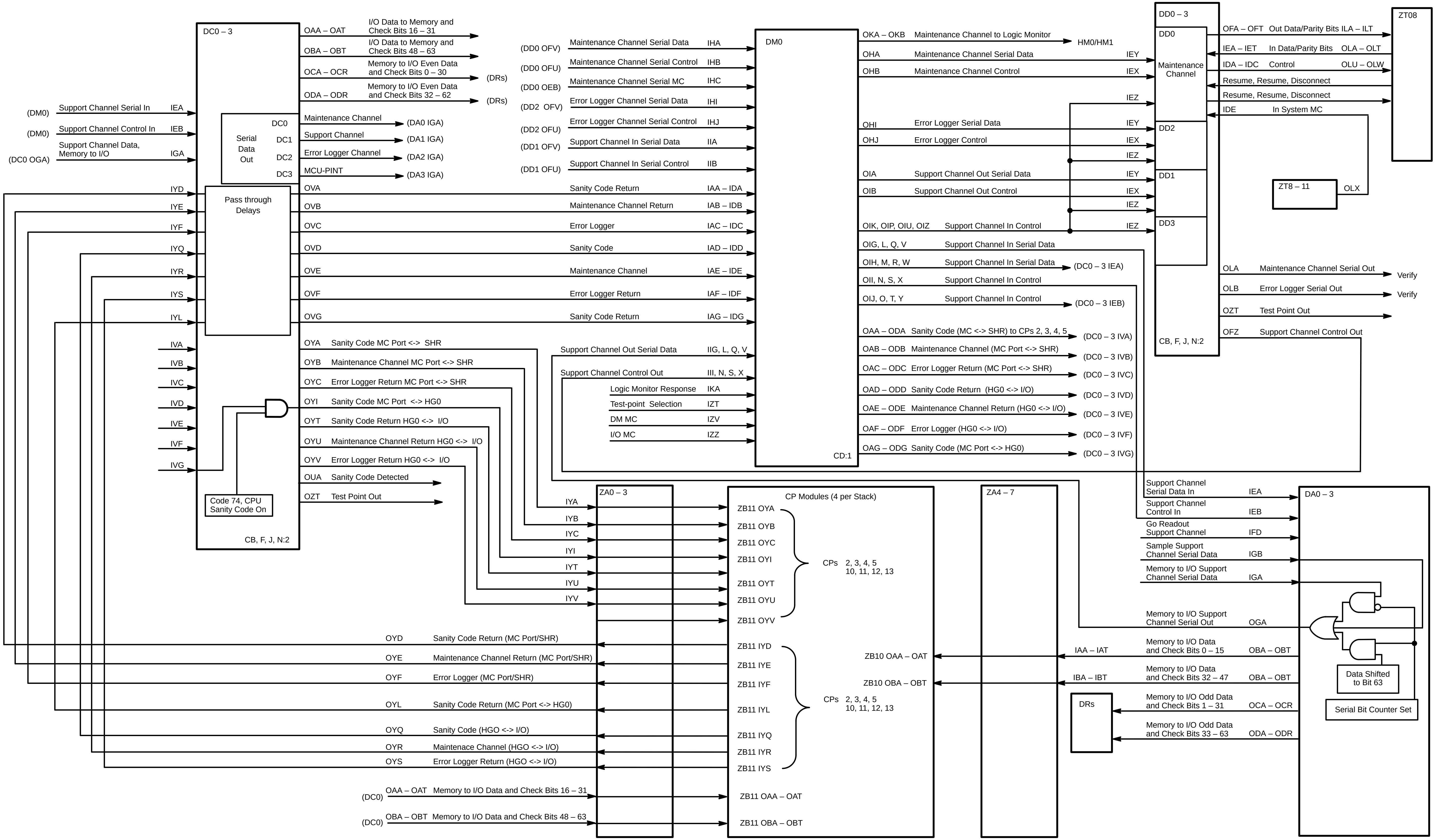

DC0 – 3
OAA – OAT
I/O Data to Memory and Check Bits 16 – 31
OBA – OBT
I/O Data to Memory and Check Bits 48 – 63
OCA – OCR
Memory to I/O Even Data and Check Bits 0 – 30
(DRs)
ODA – ODR
Memory to I/O Even Data and Check Bits 32 – 62
(DRs)
(DM0)
Support Channel Serial In
IEA
(DM0)
Support Channel Control In
IEB
(DC0 OGA)
Support Channel Data, Memory to I/O
IGA
Serial Data Out
DC0
DC1
DC2
DC3
Maintenance Channel
(DA0 IGA)
Support Channel
(DA1 IGA)
Error Logger Channel
(DA2 IGA)
MCU-PINT
(DA3 IGA)
IYD
IYE
IYF
IYQ
IYR
IYS
IYL
Pass through Delays
OVA
OVB
OVC
OVD
OVE
OVF
OVG
IVA
IVB
IVC
IVD
IVE
IVF
IVG
OYA
Sanity Code MC Port <-> SHR
OYB
Maintenance Channel MC Port <-> SHR
OYC
Error Logger Return MC Port <-> SHR
OYI
Sanity Code MC Port <-> HG0
OYT
Sanity Code Return HG0 <-> I/O
OYU
Maintenance Channel Return HG0 <-> I/O
OYV
Error Logger Return HG0 <-> I/O
OUA
Sanity Code Detected
OZT
Test Point Out
Code 74, CPU Sanity Code On
CB, F, J, N:2
(DD0 OFV)
Maintenance Channel Serial Data
IHA
(DD0 OFU)
Maintenance Channel Serial Control
IHB
(DD0 OEB)
Maintenance Channel Serial MC
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
Maintenance Channel Return
IAB – IDB
Error Logger
IAC – IDC
Sanity Code
IAD – IDD
Maintenance 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
Test-point Selection
IZT
DM MC
IZV
I/O MC
IZZ
DM0
CD:1
OKA – OKB
Maintenance Channel to Logic Monitor
HM0/HM1
OHA
Maintenance Channel Serial Data
IEY
OHB
Maintenance 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 – 3 IVA)
OAB – ODB
Maintenance Channel (MC Port <-> SHR)
(DC0 – 3 IVB)
OAC – ODC
Error Logger Return (MC Port <-> SHR)
(DC0 – 3 IVC)
OAD – ODD
Sanity Code Return (HG0 <-> I/O)
(DC0 – 3 IVD)
OAE – ODE
Maintenance Channel Return (HG0 <-> I/O)
(DC0 – 3 IVE)
OAF – ODF
Error Logger (HG0 <-> I/O)
(DC0 – 3 IVF)
OAG – ODG
Sanity Code (MC Port <-> HG0)
(DC0 – 3 IVG)
DD0 – 3
DD0
Maintenance Channel
DD2
DD1
DD3
CB, F, J, N:2
OFA – OFT
Out Data/Parity Bits ILA – ILT
IEA – IET
In Data/Parity Bits
OLA – OLT
IDA – IDC
Control
OLU – OLW
Resume, Resume, Disconnect
Resume, Resume, Disconnect
IDE
In System MC
ZT08
ZT8 – 11
OLX
OLA
Maintenance Channel Serial Out
Verify
OLB
Error Logger Serial Out
Verify
OZT
Test Point Out
OFZ
Support Channel Control Out
ZA0 – 3
IYA
IYB
IYC
IYI
IYT
IYU
IYV
CP Modules (4 per Stack)
ZB11 OYA
ZB11 OYB
ZB11 OYC
ZB11 OYI
ZB11 OYT
ZB11 OYU
ZB11 OYV
CPs 2, 3, 4, 5 10, 11, 12, 13
ZB11 IYD
ZB11 IYE
ZB11 IYF
ZB11 IYL
ZB11 IYQ
ZB11 IYR
ZB11 IYS
CPs 2, 3, 4, 5 10, 11, 12, 13
ZB10 OAA – OAT
ZB10 OBA – OBT
ZB11 OAA – OAT
ZB11 OBA – OBT
OYD
Sanity Code Return (MC Port/SHR)
OYE
Maintenance Channel Return (MC Port/SHR)
OYF
Error Logger (MC Port/SHR)
OYL
Sanity Code Return (MC Port <-> HG0)
OYQ
Sanity Code (HGO <-> I/O)
OYR
Maintenace 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
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
Memory to I/O Support Channel Serial Data
IGA
Memory to I/O Support Channel Serial Out
OGA
Memory to I/O Data and Check Bits 0 – 15
OBA – OBT
Memory to I/O Data and Check Bits 32 – 47
OBA – OBT
Memory to I/O Odd Data and Check Bits 1 – 31
OCA – OCR
Memory to I/O Odd Data and Check Bits 33 – 63
ODA – ODR
DA0 – 3
Data Shifted to Bit 63
Serial Bit Counter Set

Figure 18. Maintenance Channel and Sanity Code Paths on IO Module

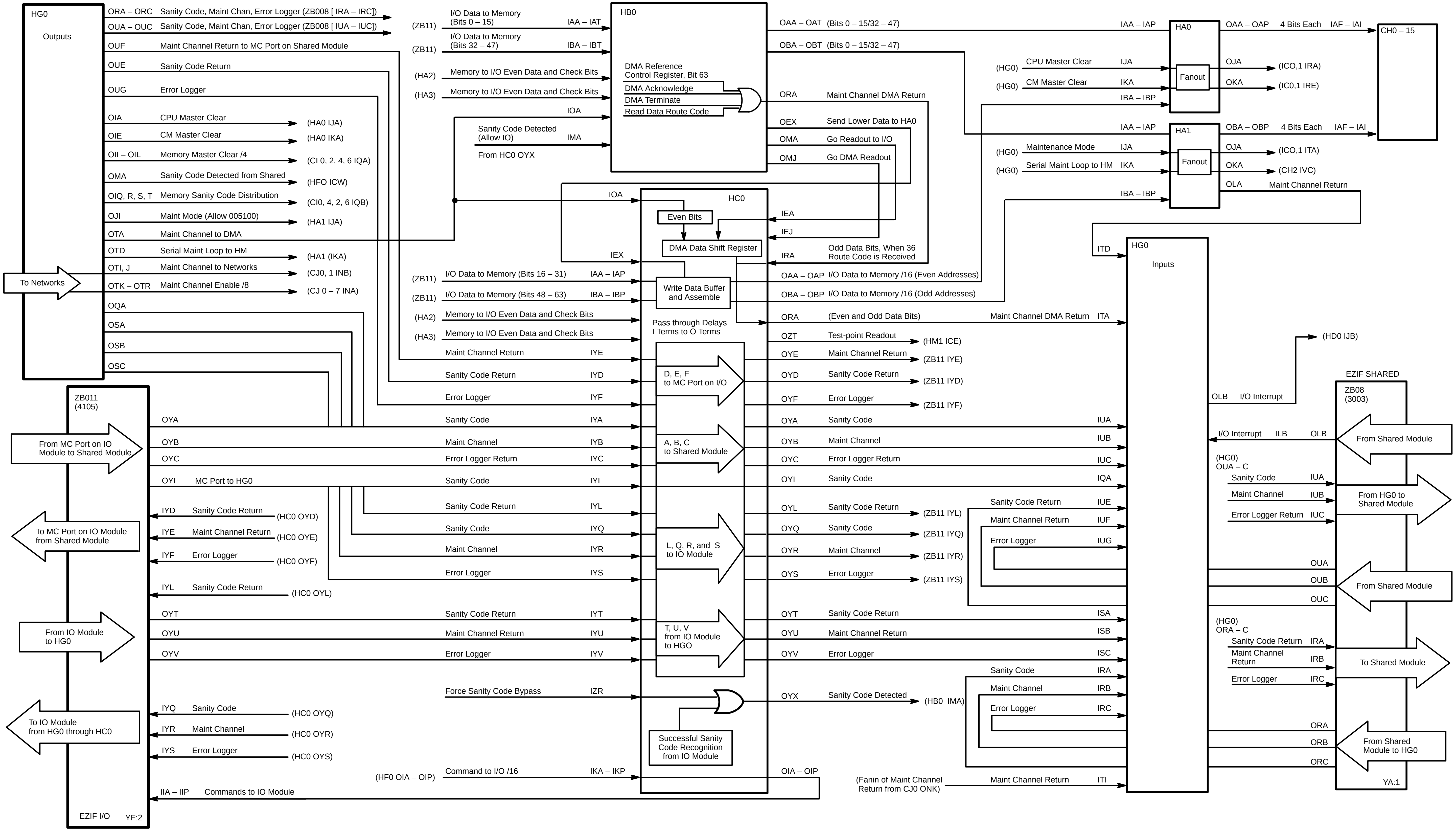
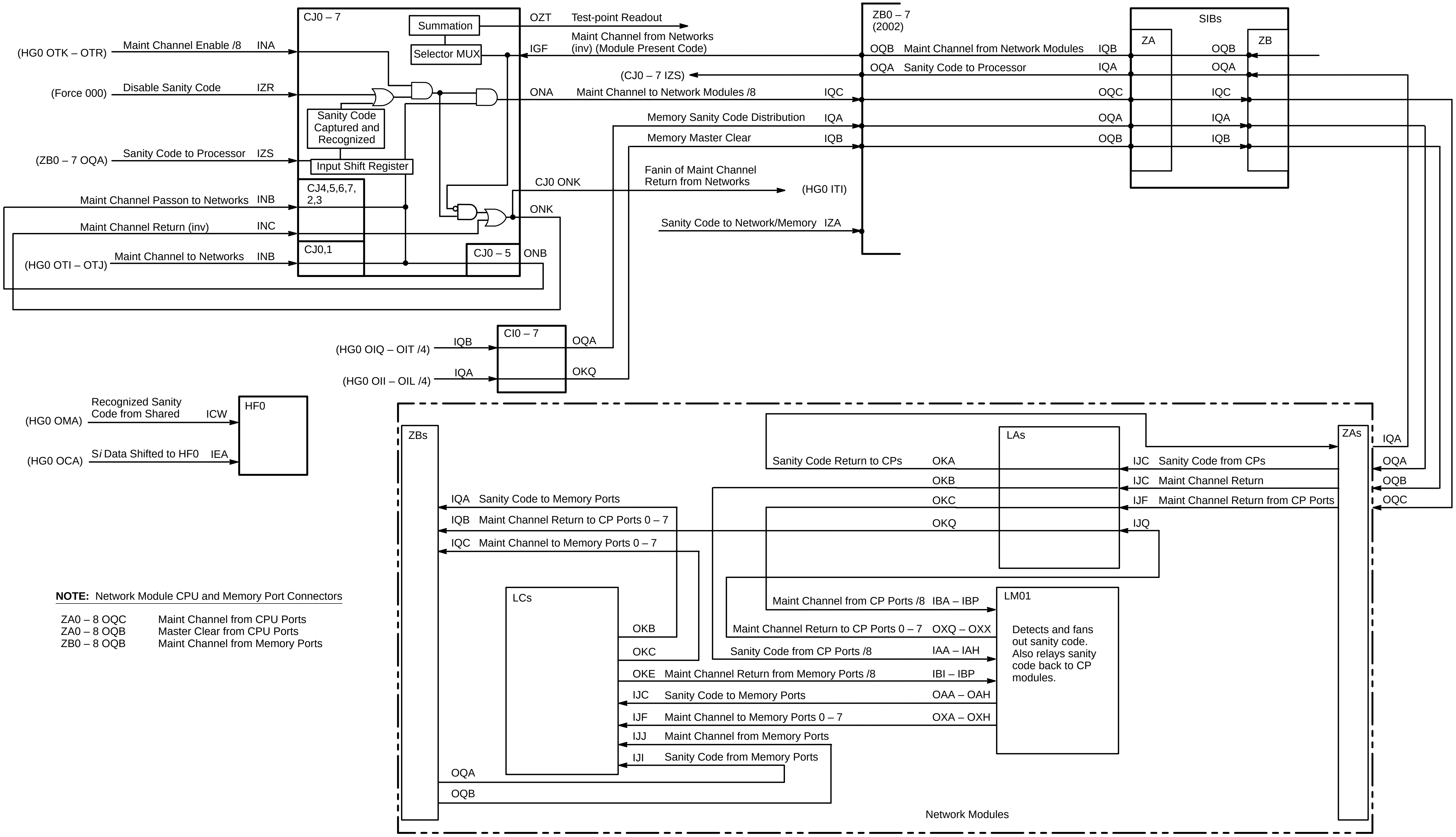

HG0
Outputs
ORA – ORC Sanity Code, Maint Chan, Error Logger (ZB008 [IRA – IRC])
OUA – OUC Sanity Code, Maint Chan, Error Logger (ZB008 [IUA – IUC])
OUF Maint Channel Return to MC Port on Shared Module
OUE Sanity Code Return
OUG Error Logger
OIA CPU Master Clear (HA0 IJA)
OIE CM Master Clear (HA0 IKA)
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)
OTK – OTR Maint Channel Enable /8 (CJ 0 – 7 INA)
To Networks
OQA
OSA
OSB
OSC
ZB011
(4105)
OYA
OYB
OYC
OYI MC Port to HG0
From MC Port on IO Module to Shared Module
IYD Sanity Code Return (HC0 OYD)
IYE Maint Channel Return (HC0 OYE)
IYF Error Logger (HC0 OYF)
IYL Sanity Code Return (HC0 OYL)
To MC Port on IO Module from Shared Module
OYT
OYU
OYV
From IO Module to HG0
IYQ Sanity Code (HC0 OYQ)
IYR Maint Channel (HC0 OYR)
IYS Error Logger (HC0 OYS)
To IO Module from HG0 through HC0
IIA – IIP Commands to IO Module
EZIF I/O
YF:2
(ZB11) I/O Data to Memory (Bits 0 – 15) IAA – IAT
(ZB11) I/O Data to Memory (Bits 32 – 47) IBA – IBT
(HA2) Memory to I/O Even Data and Check Bits
(HA3) Memory to I/O Even Data and Check Bits
IOA
Sanity Code Detected (Allow IO) 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)
OBA – OBT (Bits 0 – 15/32 – 47)
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 /16 (Even Addresses)
OBA – OBP I/O Data to Memory /16 (Odd Addresses)
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, C 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
Maint Channel IYR
Error Logger IYS
L, Q, R, and S to IO 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 IO Module to HGO
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 IO Module
(HF0 OIA – OIP) Command to I/O /16 IKA – IKP
OIA – OIP
(Fanin of Maint Channel Return from CJ0 ONK) Maint Channel Return ITI
IAA – IAP
HA0
OAA – OAP 4 Bits Each IAF – IAI
CH0 – 15
(HG0) CPU Master Clear IJA
(HG0) CM Master Clear IKA
Fanout
OJA (ICO,1 IRA)
OKA (IC0,1 IRE)
IBA – IBP
IAA – IAP
HA1
OBA – OBP 4 Bits Each IAF – IAI
(HG0) Maintenance Mode IJA
(HG0) Serial Maint Loop to HM IKA
Fanout
OJA (ICO,1 ITA)
OKA (CH2 IVC)
OLA Maint Channel Return
IBA – IBP
ITD
HG0
Inputs
Sanity Code Return IUE
Maint Channel Return IUF
Error Logger IUG
Sanity Code IRA
Maint Channel IRB
Error Logger IRC
(HD0 IJB)
OLB I/O Interrupt
EZIF SHARED
ZB08
(3003)
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
OUA
OUB
OUC
From Shared Module
(HG0)
ORA – C
Sanity Code Return IRA
Maint Channel Return IRB
Error Logger IRC
To Shared Module
ORA
ORB
ORC
From Shared Module to HG0
YA:1

Figure 19. Maintenance Channel and Sanity Code Paths on CP Module



NOTE: 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 20. Maintenance Channel and Sanity Code Paths, CP-to-Network Modules

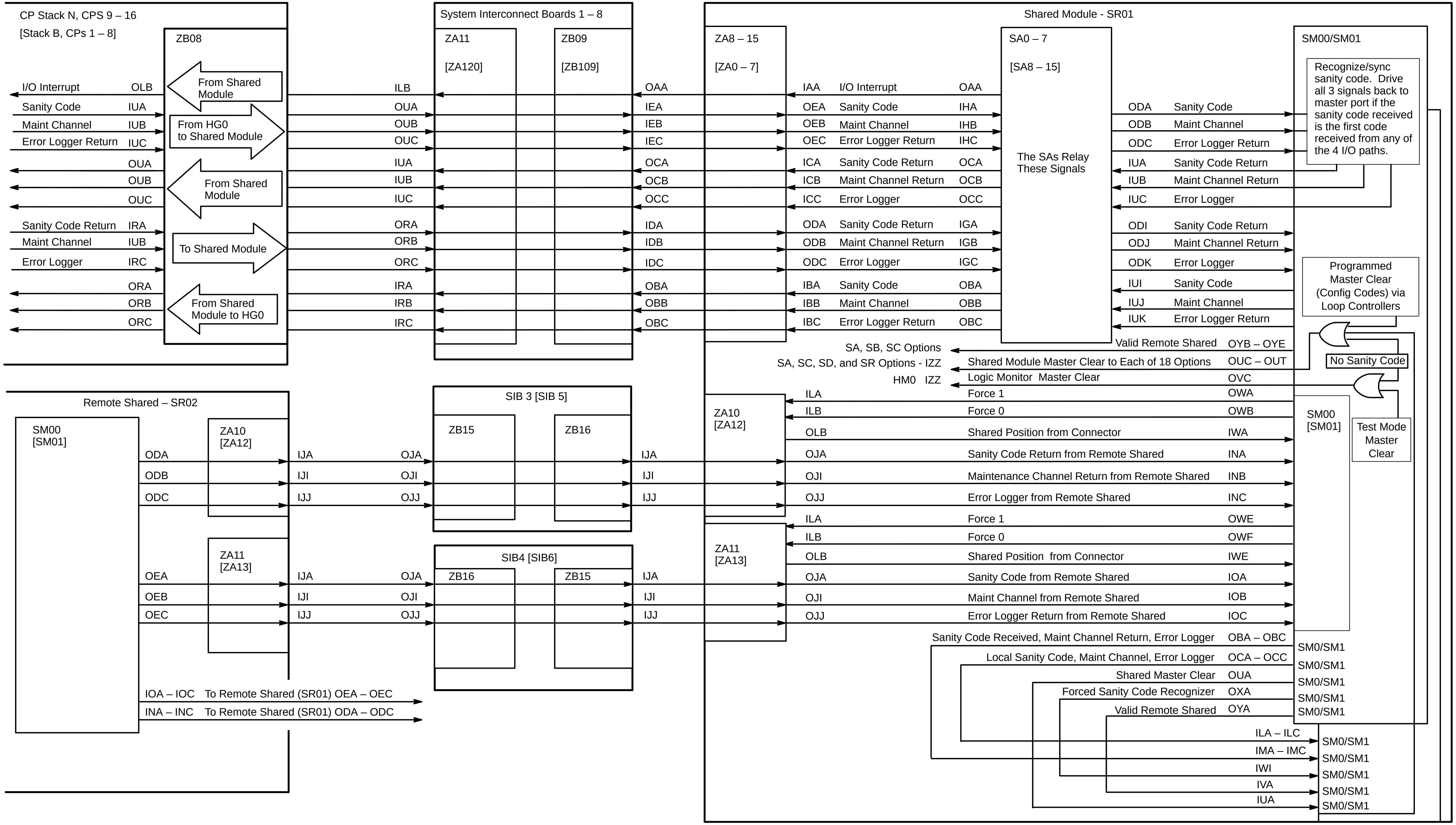

CP Stack N, CPS 9 – 16
[Stack B, CPs 1 – 8]
ZB08
I/O Interrupt OLB
Sanity Code IUA
Maint Channel IUB
Error Logger Return IUC
OUA
OUB
OUC
Sanity Code Return IRA
Maint Channel IUB
Error Logger IRC
ORA
ORB
ORC
From Shared Module
From HG0 to Shared Module
From Shared Module
To Shared Module
From Shared Module to HG0
ILB
OUA
OUB
OUC
IUA
IUB
IUC
ORA
ORB
ORC
IRA
IRB
IRC
System Interconnect Boards 1 – 8
ZA11
[ZA120]
ZB09
[ZB109]
OAA
IEA
IEB
IEC
OCA
OCB
OCC
IDA
IDB
IDC
OBA
OBB
OBC
Shared Module - SR01
ZA8 – 15
[ZA0 – 7]
IAA I/O Interrupt OAA
OEA Sanity Code IHA
OEB Maint Channel IHB
OEC Error Logger Return IHC
ICA Sanity Code Return OCA
ICB Maint Channel Return OCB
ICC Error Logger OCC
ODA Sanity Code Return IGA
ODB Maint Channel Return IGB
ODC Error Logger IGC
IBA Sanity Code OBA
IBB Maint Channel OBB
IBC Error Logger Return OBC
SA0 – 7
[SA8 – 15]
The SAs Relay These Signals
ODA Sanity Code
ODB Maint Channel
ODC Error Logger Return
IUA Sanity Code Return
IUB Maint Channel Return
IUC Error Logger
ODI Sanity Code Return
ODJ Maint Channel Return
ODK Error Logger
IUI Sanity Code
IUJ Maint Channel
IUK Error Logger Return
SM00/SM01
Recognize/sync sanity code. Drive all 3 signals back to master port if the sanity code received is the first code received from any of the 4 I/O paths.
Programmed Master Clear (Config Codes) via Loop Controllers
Valid Remote Shared OYB – OYE
No Sanity Code
SA, SB, SC Options
SA, SC, SD, and SR Options - IZZ
HM0 IZZ
Shared Module Master Clear to Each of 18 Options OUC – OUT
Logic Monitor Master Clear OVC
Force 1 OWA
Force 0 OWB
Shared Position from Connector IWA
Sanity Code Return from Remote Shared INA
Maintenance Channel Return from Remote Shared INB
Error Logger from Remote Shared INC
Force 1 OWE
Force 0 OWF
Shared Position from Connector IWE
Sanity Code from Remote Shared IOA
Maint Channel from Remote Shared IOB
Error Logger Return from Remote Shared IOC
ZA10
[ZA12]
ILA
ILB
OLB
OJA
OJI
OJJ
ZA11
[ZA13]
ILA
ILB
OLB
OJA
OJI
OJJ
SM00
[SM01]
Test Mode Master Clear
Sanity Code Received, Maint Channel Return, Error Logger OBA – OBC SM0/SM1
Local Sanity Code, Maint Channel, Error Logger OCA – OCC SM0/SM1
Shared Master Clear OUA SM0/SM1
Forced Sanity Code Recognizer OXA SM0/SM1
Valid Remote Shared OYA SM0/SM1
ILA – ILC SM0/SM1
IMA – IMC SM0/SM1
IWI SM0/SM1
IVA SM0/SM1
IUA SM0/SM1
Remote Shared – SR02
SM00
[SM01]
ODA
ODB
ODC
OEA
OEB
OEC
ZA10
[ZA12]
ZA11
[ZA13]
IJA OJA
IJI OJI
IJJ OJJ
IJA OJA
IJI OJI
IJJ OJJ
SIB 3 [SIB 5]
ZB15
ZB16
IJA
IJI
IJJ
SIB4 [SIB6]
ZB16
ZB15
IJA
IJI
IJJ
IOA – IOC To Remote Shared (SR01) OEA – OEC
INA – INC To Remote Shared (SR01) ODA – ODC

Figure 21. Maintenance Channel and Sanity Code Paths from CPU, through SIB, to the Shared Module