SppRing and Distributed Memory Test V2.0

Document No. 700-033130-000

May 1995

CONVEX Press
Richardson, Texas
United States of America

SppRing and Distributed Memory Test V2.0

Document No. 700-033130-000

Printed in the United States of America

SppRing and Distributed Memory Test

1

Convex Document #700-033130-000
Revision 2.0
May, 1995.
Alex Chan

This document describes the operation of the *sppring* test program. *sppring* is a functional test for the SppRing and distributed memory portion of a Camelot system. *sppring* verifies the operation of the related hardware. Specifically, the test accomplishes the following:

- Verifies that the test station can access CCMC.
- Verifies that the test station can perform read and write operations to a node's memory without assistance from the T-chip.
- Verifies that the test station can download a driver to node memory.
- Verifies that the test station can access the SppRing controller through CCMC and SppRing.
- Verifies that the distributed memory circuit is functioning normally.
- Verifies memory coherency logic.
- Verifies memory non-coherent logic.
- Verifies BDT, special BDT, and EIR interrupt logic.
- Verifies messaging hardware.
- Verifies MAUI thread and TOC circuits.
- Stresses coherency circuits.

sppring does not exhaustively check the interconnect or test the gate array internal circuit, e.g. CCMC and BPR. Separate "interconnect" and CAST tests should be used for such purposes.

Hardware Requirements

sppring runs on a HP workstation connected to a Camelot system. It communicates with the utility board (MU) by passing message packets between the sparc workstation and the utility board through the DaRT bus. The utility board then performs the request. Table 1 shows the minimum set of Camelot

hardware that is required for the *sppring* program. As shown in Table 1, some of the subtests require the processor to be connected to a SppRing.

Table 1
Hardware Requirements

Subtests	Hardware Requirements
All	Camelot system with at least one node having one or more processors.
Class 7,8,9,10	2 or more nodes with nodes connected by SppRings.

Test Invocation

Test Invocation

sppring runs under the Camelot test user interface (TUI). To invoke *sppring*, user selects SppRing functional test in TUI and inputs the desired options (see TUI documentation).

You can also invoke *sppring* using the command line option. Use the sequence below for the invocation of *sppring*.

>sppring [*-option* [*-option* [...]]] RETURN

The list of invocation options for *sppring* are listed in Table 2:

Table 2
Invocation Options

Option	Description
-o complex_name	Specify complex that test is to run on.
-c value[,value]	Execute the specified class(es) of tests. Default is to execute all classes.
-s value[,value]	Execute the specified list of subtests. Default is to execute all subtests.
-pb [on,off]	Pause at the beginning of each subtest. Default is not to pause
-pe [on,off]	Pause at the end of each subtest. Default is not to pause.
-pp [on,off]	Pause when the subtest passes. Default is not to pause.
-pf [on,off]	Pause when the subtest fails. Default is not to pause.

Table 2
Invocation Options

Option	Description
-mt value	Set the number of allowable total failures to "value".
-ms value	Set the number of allowable subtest failures to "value'.
-lt value	Set number of loops run to "value".
-ls subtest# loop#	Set number of loops run ("loop#") for subtest#.
-U pattern1_0 pattern 1_1 pattern2_0 pattern2_1	Specify two 64 bit patterns for subtests 2120, 2230, 3120, and 3230.
-S seed1# seed2#	Specify two seed numbers for class 10 subtests.
-O [w,s]	Specify whether to run the subtests followed in "weak" or "strong" order mode. Default is "weakly ordered".
-P [on, off]	Specify whether to turn pre-fetch mode on or off (for SPP1200 only). Default is off.
-Z [hi, lo]	Specify stress size of 32 64-byte line (hi) or 4 64-byte line (lo). Default is lo.
-N #line	Specify number of lines to be used for subtests 10300 and 10310.

Class Descriptions

This section describes the class of subtests that are available during the execution phase of the *sppring* program. There are 10 classes of subtests, each of which verifies a different aspect of the node's functionality. In general, the test configuration requirements are the same for all subtests within a class.

Class 1 - Access Verification Subtests

Class 1 subtests verify that the sparc workstation can access memory, CCMC, SppRing controller, and SppRing through MU, MAUI, XBAR, CCMC, and BPR. It also verifies that the network cache and error reporting circuits of CCMC are functioning correctly.

Table 3
Class 1 Subtests Description

Subtest	Description
1000	CCMC Access Test
1010	Memory Access Test
1020	SppRing Controller Access Test

Table 3
Class 1 Subtests Description

Subtest	Description
1030	Network Cache Access Protection Test
1040	CCMC Error Reporting Test
1050	Variable Length Write/Read Test

Class 2 - Distributed Memory (Tag) Subtests

Class 2 subtests check the integrity of the distributed memory system. Subtests in class 2 are run on all the specified nodes simultaneously by stuffing corresponding tag code from primary loader PDC space to T-chip cache. The list of subtests in class 2 is listed in Table 4.

Table 4
Class 2 Subtests Description

Subtest	Description
2000	Tag Column Functionality Test
2010	Tag Uniqueness Test
2020	Tag MATS Test Pattern 1
2030	Tag MATS Test Pattern 2
2040	Tag MATS Test Pattern 3
2050	Tag MATS Test Pattern 4
2060	Tag MATS Test Pattern 5
2070	Tag MATS Test Pattern 6
2080	Tag MATS (User Specified Pattern)
2090	Tag NTA Test Pattern 1
2100	Tag NTA Test Pattern 2
2110	Tag NTA Test Pattern 3
2120	Tag NTA Test Pattern 4
2130	Tag NTA Test Pattern 5
2140	Tag NTA Test Pattern 6
2150	Tag NTA (User Specified Pattern)

Class 3 - Distributed Memory (Data) Subtests

Class 3 subtests check the integrity of the distributed memory system. Subtests in class 3 are run on all the specified nodes simultaneously by stuffing

corresponding memory code from primary loader PDC space to T-chip cache (except memory cycles test). The list of subtests in class 3 is listed in Table 5.

Table 5
Class 3 Subtests Description

Subtest	Description
3000	Memory Column Functionality Test
3010	Memory Uniqueness Test
3020	Memory MATS Test Pattern 1
3030	Memory MATS Test Pattern 2
3040	Memory MATS Test Pattern 3
3050	Memory MATS Test Pattern 4
3060	Memory MATS Test Pattern 5
3070	Memory MATS Test Pattern 6
3080	Memory MATS (User Specified Pattern)
3090	Memory NTA Test Pattern 1
3100	Memory NTA Test Pattern 2
3110	Memory NTA Test Pattern 3
3120	Memory NTA Test Pattern 4
3130	Memory NTA Test Pattern 5
3140	Memory NTA Test Pattern 6
3150	Memory NTA (User Specified Pattern)
3160	Memory Cycles Test

Table 6: Patterns used for MATS and NTA subtests

Pattern #	Patterns Used
1	0 & 0xffffffffffffffff
2	0x0000ffff0000ffff & 0x00ff00ff00ff00ff
3	0x0f0f0f0f0f0f0f0f & 0xf0f0f0f0f0f0f0f0
4	0x3333333333333333 & 0x5555555555555555
5	0x8000000080000000 & 0x2000000020000000
6	0x1000000010000000 & 0x0800000008000000

Class 4 - Non-Coherent Subtests

Class 4 subtests check the non-coherent read/write function of the system. The list of subtests in class 4 is listed in Table 7.

Table 7
Class 4 Subtests Description

Subtest	Description
4000	Local Non-Coherent Read/Write Test
4010	SppRing Non-Coherent Read/Write Test

Class 5 - Download Verification and EIR Check Subtests

Class 5 subtests check the download and CPA EIR interrupt generation function of the system. It also tests the MAUI TOC and thread count circuit. The list of subtests in class 5 is listed in Table 8.

Table 8
Class 5 Subtests Description

Subtest	Description
5000	Node Download Verification Test
5010	External Interrupt Verification Test
5020	MAUI CSRs Test
5030	MAUI Thread Count Circuit Test
5040	MAUI TOC Circuit Test
5050	MAUI TOC Interrupt Verification Test

Class 6 - Intranode Coherency Subtests

Class 6 subtests check the Intranode coherency function of the system. The list of subtests in class 6 is listed in Table 9.

Table 9
Class 6 Subtests Description

Subtest	Description
6000-6030	Intra-Coherency Test (Even)
6040-6070	Intra-Coherency Test (Odd)
6080-6110	Intra-Coherency Test (Even&Odd)
6120-6150	Intra-Coherency Parallel Test (Even)
6160-6190	Intra-Coherency Parallel Test (Odd)
6200-6230	Purge Data Cache Test

Class 7 - SppRing Internode Coherency Subtests

Class 7 subtests check the SppRing internode coherency circuit. The list of subtests in class 7 is listed in Table 10.

Table 10
Class 7 Subtests Description

Subtest	Description
7000-7030	BDT Test
7040-7070	Special BDT Test
7080-7110	Inter-Coherency Tag Test 1
7120-7150	Inter-Coherency Tag Test 2
7160-7190	Full Coherency Test (Even)
7200-7230	Full Coherency Test (Odd)
7240-7270	Full Coherency Parallel Test (Even)
7280-7310	Full Coherency Parallel Test (Odd)

Class 8 - Messaging Subtests

Class 8 subtests check the messaging hardware of the system. The list of subtest in class 8 is listed in Table 11.

Table 11
Class 8 Subtests Description

Subtest	Description
8000	CCMC Messaging Circuit Test (per slice)
8010	CCMC Messaging Circuit Test (all cpus)
8020	CCMC Messaging Queue Disable Test
8030	CCMC Messaging Queue Full Test

Class 9 - SCI Stress Subtests

Class 9 subtests perform sequences of load/store/flush to an internode address to stress SCI node chip. The list of subtest in class 9 is listed in Table 12.

Table 12
Class 9 Subtests Description

Subtest	Description
9000,9120,9240,9360,9480	SCI MATS Test Pattern 1
9010,9130,9250,9370,9490	SCI MATS Test Pattern 2
9020,9140,9260,9380,9500	SCI MATS Test Pattern 3
9030,9150,9270,9390,9510	SCI MATS Test Pattern 4
9040,9160,9280,9400,9520	SCI MATS Test Pattern 5
9050,9170,9290,9410,9530	SCI MATS Test Pattern 6
9060,9180,9300,9420,9540	SCI NTA Test Pattern 1
9070,9190,9310,9430,9550	SCI NTA Test Pattern 2
9080,9200,9320,9440,9560	SCI NTA Test Pattern 3
9090,9210,9330,9450,9570	SCI NTA Test Pattern 4
9100,9220,9340,9460,9580	SCI NTA Test Pattern 5
9110,9230,9350,9470,9590	SCI NTA Test Pattern 6

Class 10 - Coherency Stress Subtest

Class 10 subtest perform a random load/store/flush from each processor. Subtest in class 10 is listed in Table 13.

Table 13
Class 10 Subtest Description

Subtest	Description
10000	Coherency Stress Test
10010	Messaging Stress Test
10020	Messaging/Coherency Stress Test
10030	CCMC Stress Test
10040	CCMC/Messaging Stress Test
10050	Load/Store One Line Stress Test
10060	Mix Ops One Line Stress Test
10070	Load/Store Two Line Stress Test
10080	Mix Ops Two Line Stress Test
10090	Lock Stress Test
10100	Load One Line Stress Test
10110	Load Two Line Stress Test
10150	APA Mix Ops One Line Stress Test
10160	PMON Test #1
10170	PMON Test #2
10190	Lookaside
10200	Load Muti Line Stress Test
10210	Load/Store Multi Line Stress Test
10220	Mix Ops Multi Line Stress Test
10300	Load N Line Stress Test
10310	Load/Store N Line Stress Test
10500	OD_INV Stress Test
10510	Incr/decr Stress Test

Class 11 - SIMM Quick Sanity Subtests

Class 11 subtests perform a quick sanity check on tag and data address lines. This is mainly used for initial board bringup by manufacturing. It does not require any tchip to be present. The list of subtests in class 11 is listed in Table 14.

Table 14

Class 11 Subtests Description

Subtest	Description
11000	TS Address Bits InitThe list of subtests in class 11 is listed in Table 13
11010	TS Address Bits Uniqueness Quick Check

Class 12 - Tag SIMM Quick Check Subtests

Class 12 subtests perform a quick check on first 1KBytes of tag simms. This is mainly used for initial board bringup by manufacturing. It does not require any tchip to be present. The list of subtests in class 11 is listed in Table 15.

Table 15

Class 12 Subtests Description

Subtest	Description
12000	TS Tag Column Functionality Test
12010	TS Tag Uniqueness Test
12020	TS Tag MATS Test Pattern 1
12030	TS Tag MATS Test Pattern 2
12040	TS Tag MATS Test Pattern 3
12050	TS Tag MATS Test Pattern 4
12060	TS Tag MATS Test Pattern 5
12070	TS Tag MATS Test Pattern 6
12080	TS Tag NTA Test Pattern 1
12090	TS Tag NTA Test Pattern 2
12100	TS Tag NTA Test Pattern 3
12110	TS Tag NTA Test Pattern 4
12120	TS Tag NTA Test Pattern 5
12130	TS Tag NTA Test Pattern 6
12140	TS Tag Init

Class 13 - Data SIMM Quick Check Subtests

Class 13 subtests perform quick check on the first 1KBytes of data simms. This is mainly used for initial board bringup by manufacturing. It does not require any tchip to be present. The list of subtests in class 13 is listed in Table 16.

Subtest 1010 - Memory Access Test

This subtest verifies that the sparc workstation can perform read and write operations to memory through MU, MAUI, XBAR, CCMC, and BPR.

Subtest 1020 - SppRing Controller Access Test

This subtest verifies that the sparc workstation can access SppRing controller CSRs through SppRing.

Subtest 1030 - Network Cache Access Protection Test

Subtest 1030 verifies the network cache access protection circuit. It attempts to access network cache as part of the memory and expects an error to occur.

Subtest 1040 - CCMC Error Reporting Test

Subtest 1040 verifies the CCMC error reporting circuit. It sets various CCMC error bits and checks if the corresponding error occurs.

Subtest 1050 - Variable Length Write/Read Test

Subtest 1050 writes and reads variable length packets to memory. It checks out variable length write/read to memory from MU.

Class 2 Subtests

Subtest 2000 - Tag Column Functionality Test

Subtest 2000 tests tag column functionality. It performs a walking 1s and 0s test on the first word in each bank of tag memory (walking from least significant bit to most significant bit). All subtests in this class are downloaded to T-chip cache before execution.

Subtest 2010 - Tag Uniqueness Test

Subtest 2010 tests tag location uniqueness. It writes an incrementing value to each location in tag memory, then verifies that all locations contain the expected value.

Subtests 2020-2080 - Tag MATS Test

Subtest 2020-2080 performs a MATS+ memory pattern test over the tag memory. The MATS+ algorithm consists of several passes through tag memory, with intermixed reads and writes of two patterns. In this implementation of the algorithm, various 64 bit data patterns (2 per subtest) are chosen such that all data and ECC bits will be tested within the 64 bit word (over the course of all pattern subtests), and all detectable intra-longword dependencies will be detected. Figure 1 illustrates MATS+ algorithm.

Table 16

Class 13 Subtests Description

Subtest	Description
13000	TS Memory Column Functionality Test
13010	TS Memory Uniqueness Test
13020	TS Memory MATS Test Pattern 1
13030	TS Memory MATS Test Pattern 2
13040	TS Memory MATS Test Pattern 3
13050	TS Memory MATS Test Pattern 4
13060	TS Memory MATS Test Pattern 5
13070	TS Memory MATS Test Pattern 6
13080	TS Memory NTA Test Pattern 1
13090	TS Memory NTA Test Pattern 2
13100	TS Memory NTA Test Pattern 3
13110	TS Memory NTA Test Pattern 4
13120	TS Memory NTA Test Pattern 5
13130	TS Memory NTA Test Pattern 6

Class 14 - HP PA-RISC Subtest

Class 14 subtest performs checks on the HP PA_RISC processors. The list of subtest in class 14 is listed in Table 17.

Table 17

Class 14 Subtest Description

Subtest	Description
14000	HP Dcache Verification Test

Subtest Descriptions

This section describes the subtests that are available in each class of tests.

Class 1 Subtests

Subtest 1000 - CCMC Access Test

This subtest verifies that the sparc workstation can access CCMC CSRs.

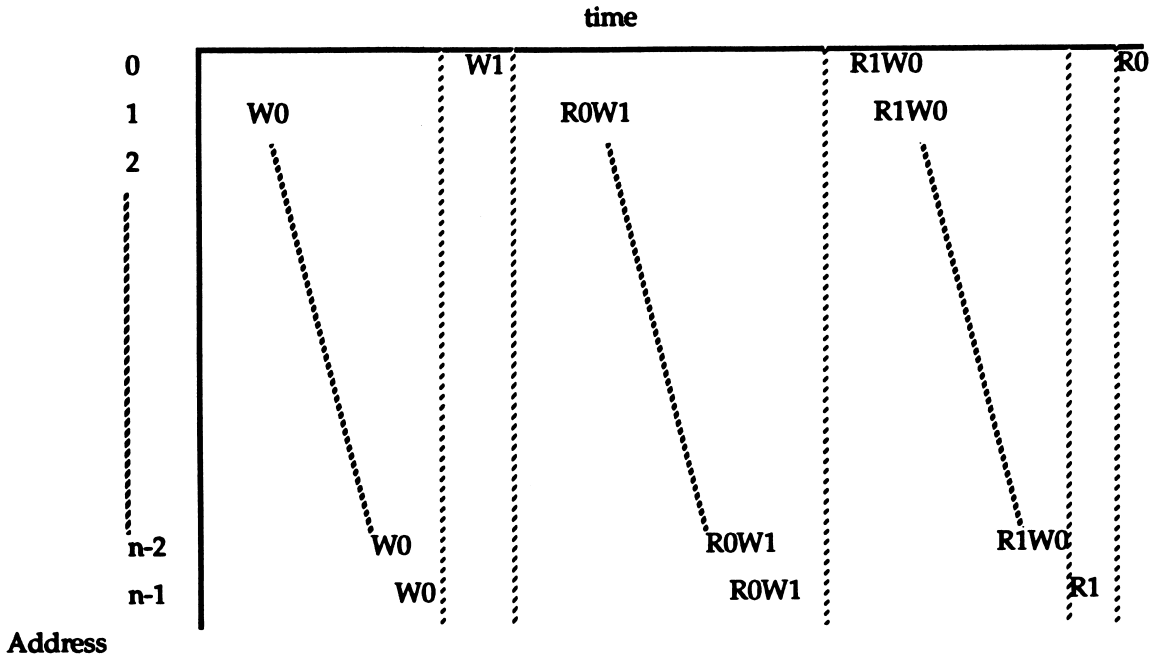


Figure 1 : MATS+ Algorithm

In the above figure, the test addresses are listed vertically, and time progresses from left to right. W means write and R means read and verify. 0 is the first of the two patterns, 1 is the second. Thus W0 means write the first pattern, while R0W1 indicates the location will first be read, then verified to be the first pattern, and finally written with the second pattern.

For each subtest, a pair of special patterns are used for MATS testing.

Subtests 2090-2150 - Tag NTA Test

Subtest 2090-2150 performs a NTA memory pattern test over the tag memory. The Nair, Thattle, and Abraham algorithm consists of several passes through tag memory, with intermixed reads and writes of two patterns. Figure 2 illustrates the algorithm.

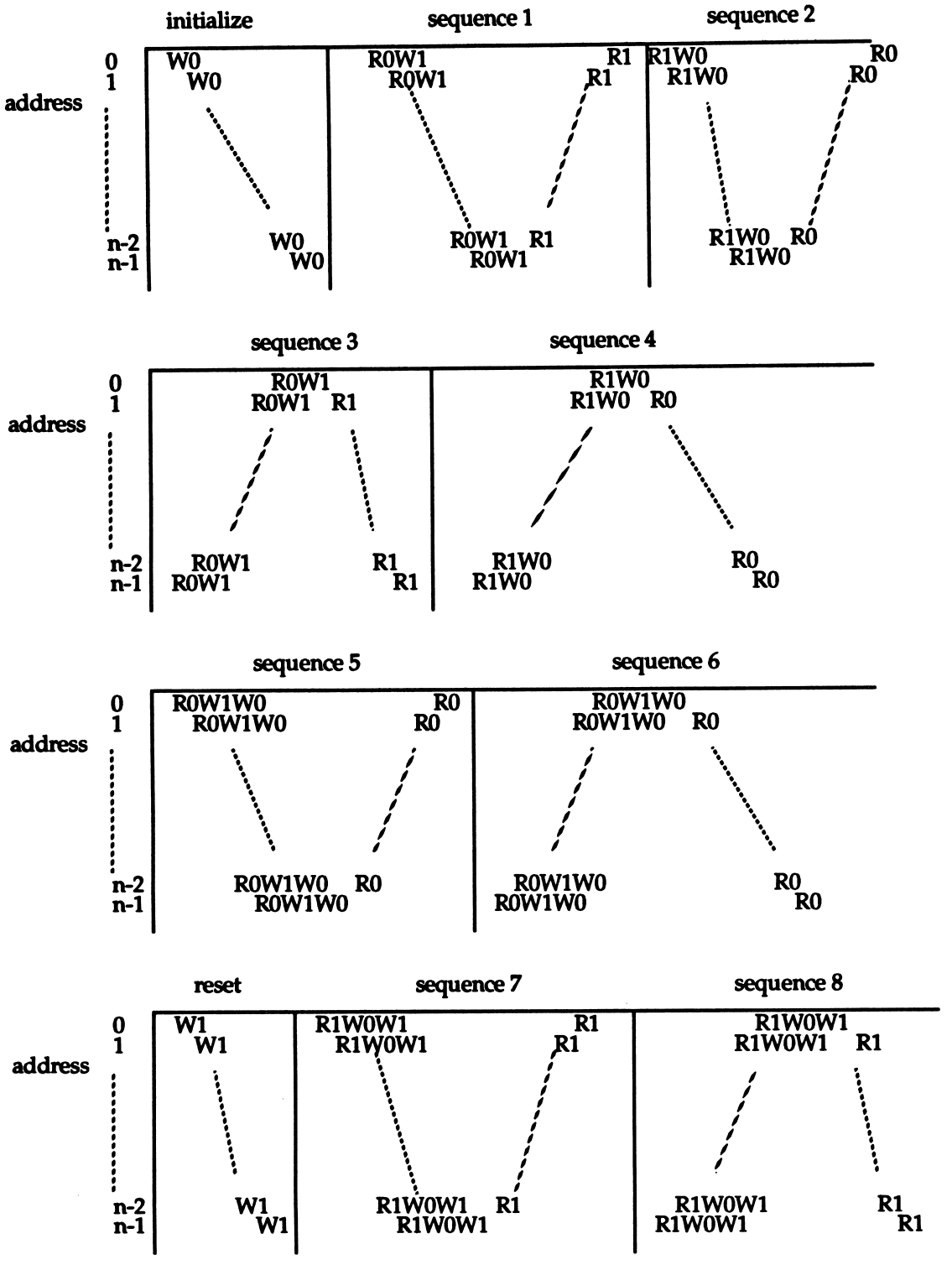


Figure 2 : Nair, Thatte, and Abraham's Algorithm

For each subtest, a pair of special patterns are used for NTA testing.

Class 3 Subtests

Subtest 3000 - Memory Column Functionality Test

Subtest 3000 tests memory column functionality. It performs a walking 1s and 0s test on the first word in each bank of memory (walking from least significant bit to most significant bit). All subtests in this class are downloaded to T-chip cache before execution.

Subtest 3010 - Memory Uniqueness Test

Subtest 3010 tests memory location uniqueness. It writes an incrementing value to each location in memory, then verifies that all locations contain the expected value.

Subtests 3020-3080 - Memory MATS Test

Subtests 3020-3080 perform a MATS+ memory pattern test over the memory. The MATS+ algorithm consists of several passes through memory, with intermixed reads and writes of two patterns. In this implementation of the algorithm, various 64 bit data patterns (2 per subtest) are chosen such that all data and ECC bits will be tested within the 64 bit word (over the course of all pattern subtests), and all detectable intra-longword dependencies will be detected. Figure 1 illustrates MATS+ algorithm.For each subtest, a pair of special patterns are used for MATS testing.

Subtests 3090 -3150- Memory NTA Test

Subtests 3090-3150 perform a NTA memory pattern test over the memory. The Nair, Thattle, and Abraham algorithm consists of several passes through memory, with intermixed reads and writes of two patterns. Figure 2 illustrates the algorithm.For each subtest, a pair of special patterns are used for NTA testing.

Subtest 3160 - Memory Cycle Test

Subtest 3160 performs load and store of byte/half word/full word to verify that processor can perform these basic operations correctly. It also verifies flush and purge operations. Fetch-and-inc and fetch-and-dec are also verified.

Class 4 Subtests

Subtest 4000 - Local Non-Coherent Read/Write Test

Subtest 4000 tests non-coherent read/write to memory from different processors in the same node. It performs non-coherent read and write operations to memory from MU.

Subtest 4010 - SppRing Non-Coherent Read/Write Test

Subtest 4010 tests SppRing non-coherent read/write commands. It performs non-coherent read and write operations over the SppRing to another node's memory.

Class 5 Subtests

Subtest 5000 - Node Download Verification Test

The PA-RISC driver program is downloaded by *sppring*. The HP test station communicates to the driver program to verify that it is successfully downloaded.

Subtest 5010 - External Interrupt Verification Test

Subtest 5010 tests the CPA external interrupt circuit. It writes to EIR to generate an external interrupt and checks if the correct interrupt is generated.

Subtest 5020 - MAUI CSRs Test

Subtest 5020 tests the following MAUI CSRs : MAUI_PVT_HI, MAUI_PVT_LO, MAUI_INT_MASK, MAUI_TOCMR_HI, MAUI_TOCMR_LO. It writes a walking 1, walking 0, and six different patterns to CSRs to verify that the CSR is written and read with correct result.

Subtest 5030 - MAUI Thread Count Circuit Test

Subtest 5030 tests the MAUI thread count circuit. It increments MAUI_TC by reading MAUI_TC_FINC from 0 to 0xffff and to 0. For each increment, it checks to verify that MAUI_PVT_HI and MAUI_PVT_LO are incrementing when MAUI_TC sign bit is not set, and they are not incrementing when MAUI_TC sign bit is set. It then decrements MAUI_TC by reading MAUI_TC_FDEC. MAUI_TC is checked for each FDEC read. MAUI_TC_FCLR is also verified that it resets MAUI_TC.

Subtest 5040 - MAUI TOC Circuit Test

Subtest 5040 tests the MAUI TOC circuit. It syncs up all MAUIs in different nodes. MAUI_TOC_HI and MAUI_TOC_LO are checked to ensure that they sync up correctly.

Subtest 5050 - MAUI TOC Interrupt Verification Test

Subtest 5050 verifies the MAUI TOC interrupt circuit. It sets MAUI_TOCMR to a known value and verifies that an interrupt is generated when MAUI_TOC matches MAUI_TOCMR.

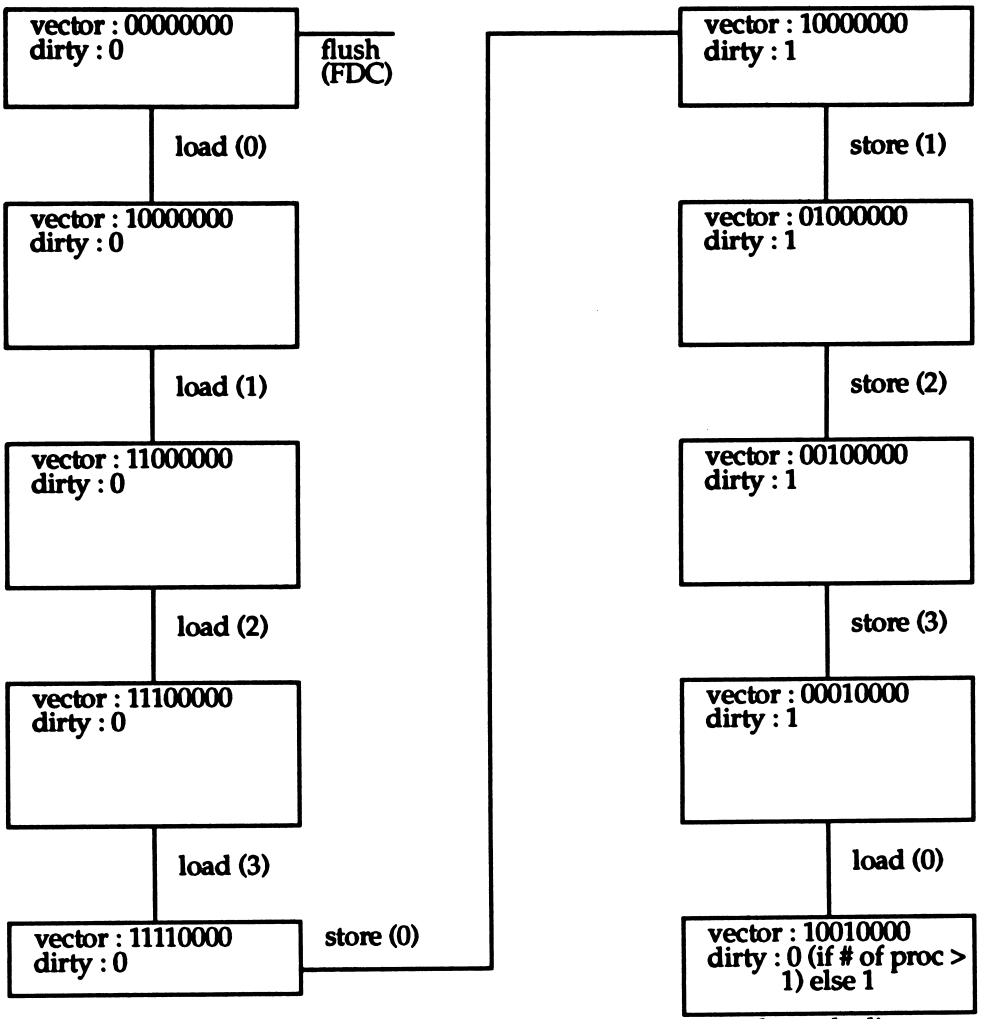
Class 6 Subtests

Subtests 6000-6110 - Intranode Coherency Tag Test

Subtests 6000-6110 verify the tag info for intranode coherency logic for both even and odd cache lines of each slice. It performs a combination of load and store operations among processors in a single node. Vector, dirty, and special BDT bits are checked. Figure 3 shows the test and the expected values of vector and dirty bits.

This subtest starts by flushing the processor cache. Tag vector and dirty bits are checked to ensure that it equals to zero. A load operation followed by a store operation are executed by each processor. The vector and dirty bits are compared against the expected values at each stage.

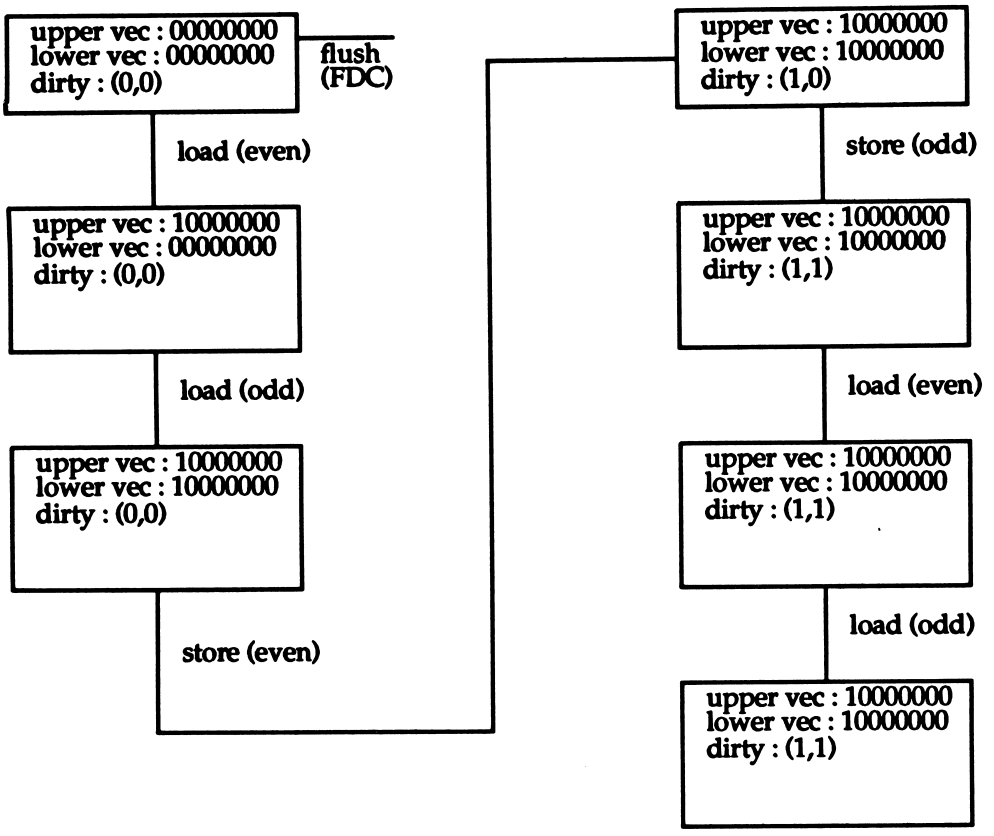
Note : All load and store operations are accessing the same memory location.



Key : vector = 8-bit vector specifying list of processors sharing the cache line.
(displays in binary format with bit 0 corresponds to processor 0 etc.)
dirty = 1 if dirty, 0 if clean
load (P), store (P) = performs load/store operation on processor #P.

Figure 3 : Intranode Coherency Tag Test (Subtests 6000-6070)

Note : All load and store operations are accessing the same cache line.



Key : upper and lower vec = 8-bit vector specifying list of processors sharing the cache line. (displays in binary format with bit 0 corresponds to processor 0 etc.)
dirty (upper,lower) = 1 if dirty, 0 if clean for upper and lower cache line.
load (P), store (P) = performs load/store operation on P cache line.

Figure 4 : Intranode Coherency Tag Test (Subtests 6080-6110)

Subtests 6120-6190 - Intranode Coherency Parallel Tag Test

Subtests 6120-6190 test the intranode coherency logic in parallel for both even and odd cache lines of each slice. All processors in a node read from a 32-byte memory line in parallel and the vector and dirty bits are checked and verified. All processors then write to the 32-byte memory line and the vector and dirty bits are checked.

Subtest 6200-6230 - Purge Data Cache Test

Subtests 6200-6230 verify the purge (pdc) command on each slice. The first processor of each node reads an address and writes the same address followed

by a PDC command. A read is done to the same address to confirm that the written data is purged. Tag data is checked after the purge operation.

Class 7 Subtests

Subtest 7000-7030 - BDT Test

Subtests 7000-7030 verify the CPA BDT circuit of each slice. It sets up a BDT to point to the first node from other nodes. The first node does a write to the setup BDT memory and the rest of the nodes read from the setup BDT memory to verify that it is reading correct data.

Subtest 7040-7070 - Special BDT Test

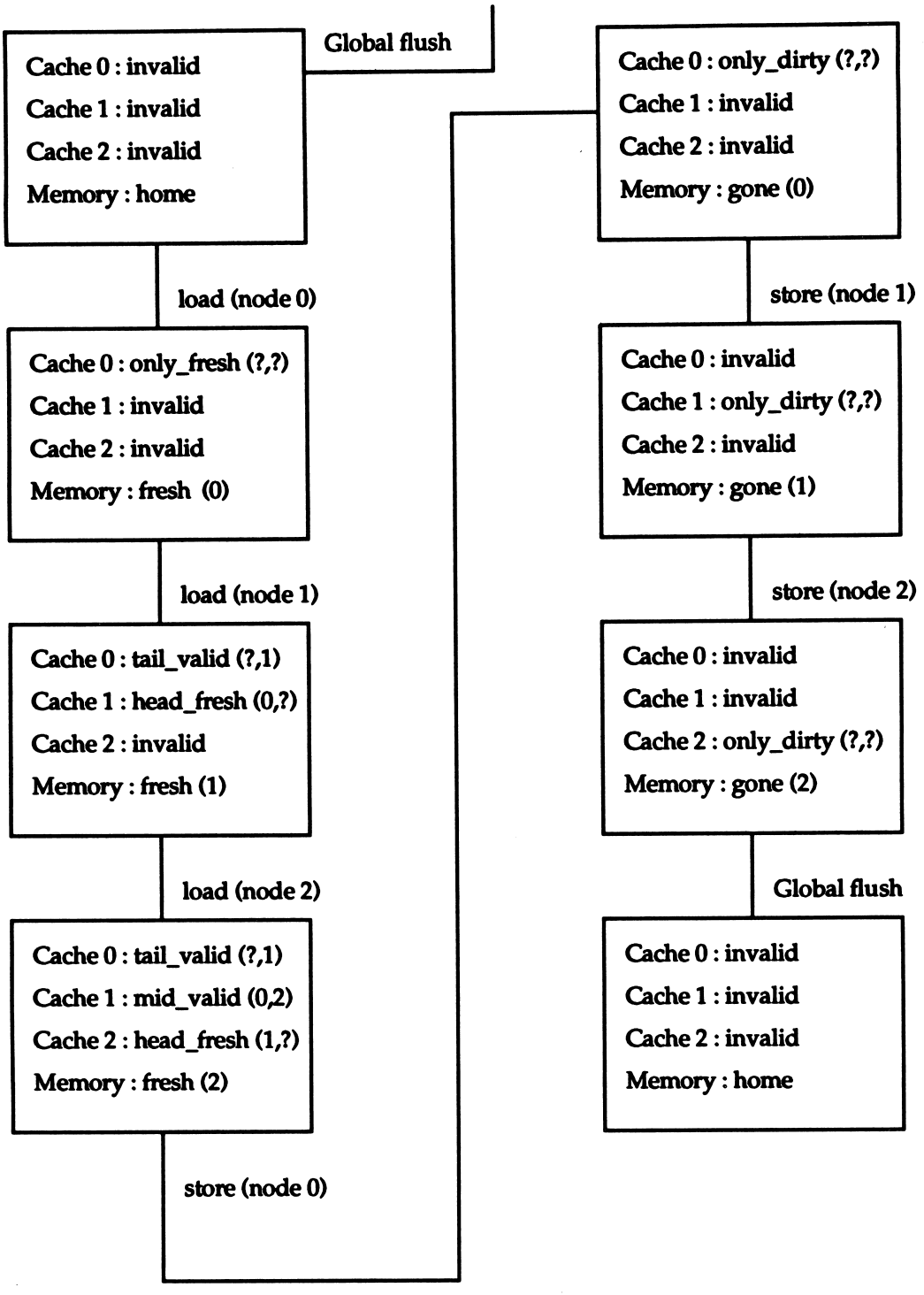
Subtests 7040-7070 verify the CPA special BDT circuit of each slice. It sets up the special BDTs to point to the first node. The first node does a write to the memory referenced by the special BDT memory and the rest of the nodes read from the special BDT memory to verify that it is reading correct data.

Subtests 7080-7150 - Internode Coherency Tag Test

Subtests 7080-7150 verify the tag info for internode coherency logic of each slice. It performs combination of load and store operations among nodes. It has two parts.

The first part (subtests 7080-7110) performs a series of load operations by all specified nodes followed by a series of store operations. The cache and memory states after each operation are compared with the expected result. Any discrepancy is reported. Figure 4 illustrates the test and describes the expected result.

The second part (subtests 7120-7150) performs a series of load operations followed by a store operation and a series of load operations. The cache and memory states after each operation are compared with the expected result. Any discrepancy is reported. Figure 5 illustrates the test and describes the expected result.



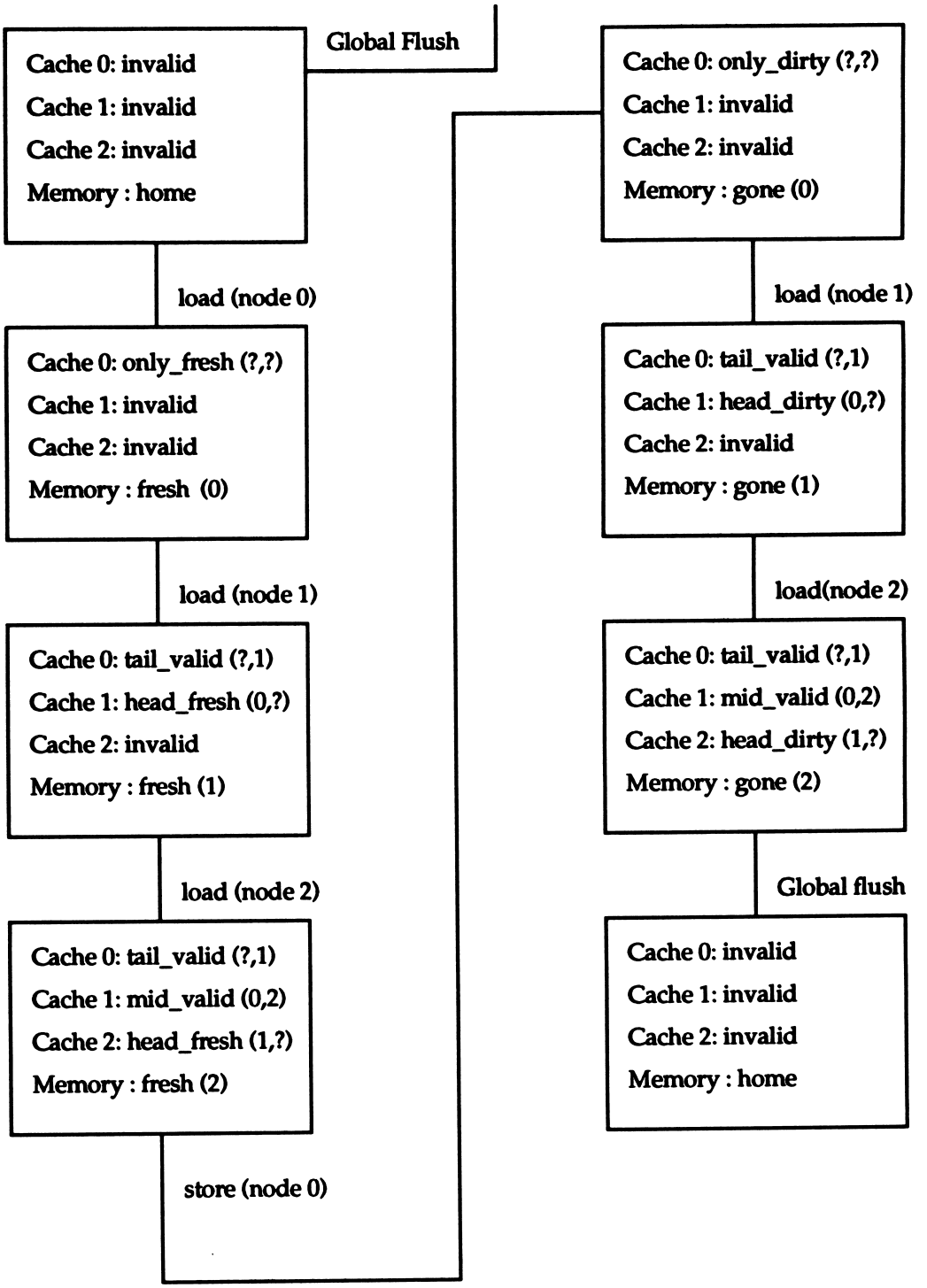
Note : load and store operations are performed to the same memory location.

Key : Cache N : state (x,y) = x is the forward pointer, and y is the backward pointer.
(Please refer IEEE P1596, SCI, for details.)

Memory : state (z) = z is the forward pointer.

? = Don't care

Figure 5 : Internode Coherency Tag Test Part I



Note : load and store operations are performed to the same memory location.

Key : Cache N : state (x,y) = x is the forward pointer, and y is the backward pointer. (Please refer IEEE P1596, SCI, for details.)

Memory : state (z) = z is the forward pointer.

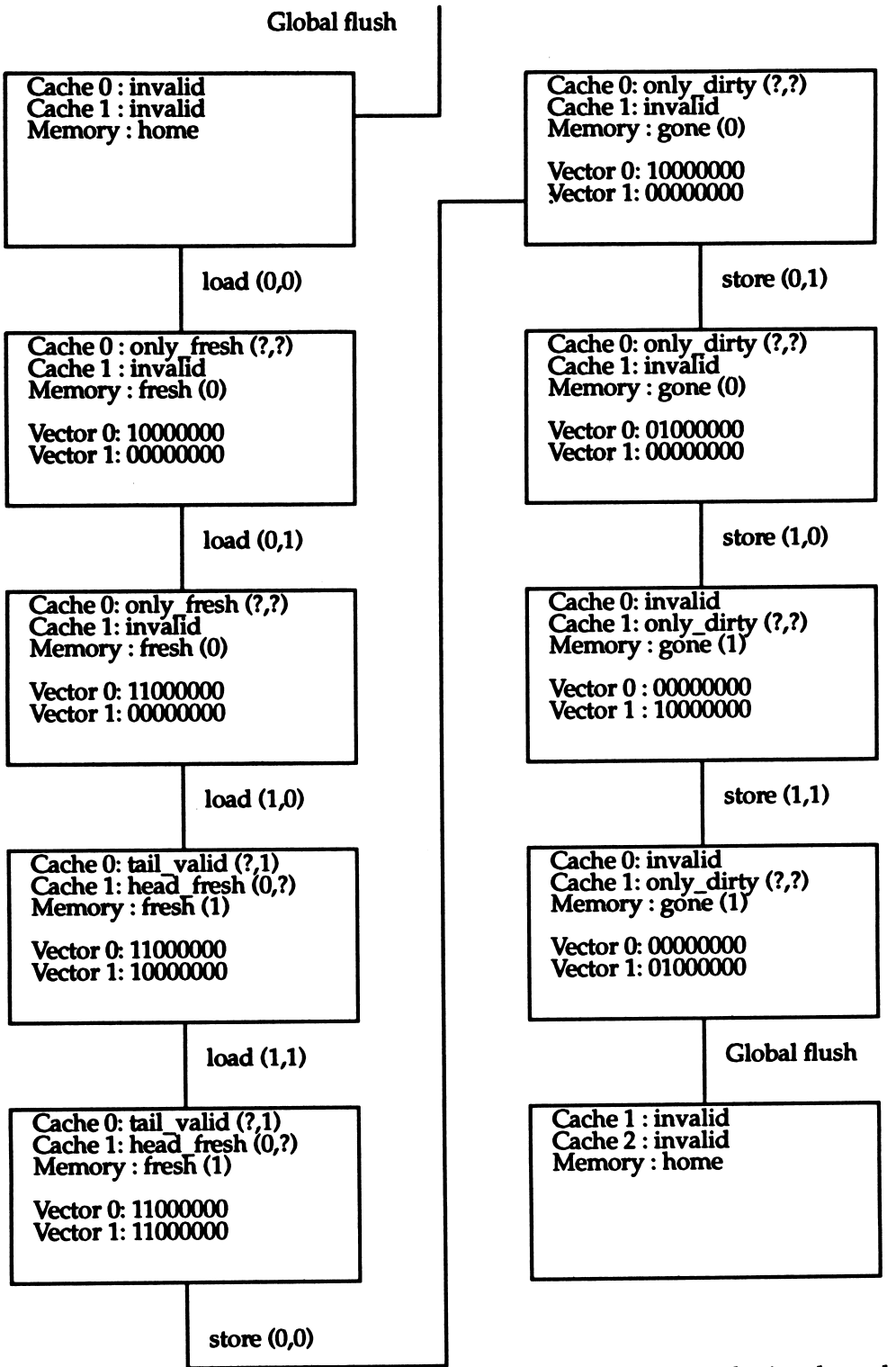
? = Don't care

Figure 6 : Internode Coherency Tag Test Part II

Subtest 7160-7230 - Full Coherency Test

Subtests 7160-7230 verify the tag info for the system coherency logic for both the even and odd cache lines of each slice. It performs combination of load and store operations among nodes and processors.

These subtests perform a series of load operations among nodes and processors within a node followed by a series of store operations. The cache and memory states, and vector bits are compared with the expected values. Any discrepancy is reported. Figure 6 illustrates the test and the expected result.



Key : vector N = 8-bit vector of node N, specifies list of processors sharing the cache line. (displays in binary format with bit 0 corresponds to processor 0 etc.)
load (N,P) and store (N,P) = load and store operation performed on node N, processor P. All load and store are performed to the same memory location.

? = Don't care

Figure 7 : Full Coherency Test

Subtests 7240-7310 - Full Coherency Parallel Test

Subtests 7240-7310 test the coherency logic in parallel for both even and odd cache lines of each slice. All processors in all specified nodes read from a 32-byte memory line in parallel and the vector and dirty bits, memory and cache states are checked and verified. All processors in all specified nodes then write to the 32-byte memory line and the tag info are checked.

Class 8 Subtests

Subtest 8000 - CCMC Messaging Circuit Test (per slice)

Subtest 8000 verifies the messaging circuit by writing messages to all available T-chips on the same slice of a neighboring node. It then confirms that the messages are received by the corresponding T-chip of the destination node.

Subtest 8010 - CCMC Messaging Circuit Test (all cpus)

Subtest 8010 verifies the messaging circuit by writing messages to all available T-chips of a neighboring node. It then confirms that the messages are received by the corresponding T-chip of the destination node.

Subtest 8020 - CCMC Queue Disable Test

Subtest 8020 verifies the messaging queue disable circuit by writing messages to a tchip that has its message queue disabled. It then verifies that a HPMC is received..

Subtest 8030 - CCMC Messaging Queue Full Test

Subtest 8030 verifies the messaging queue full circuit by writing messages to a tchip that has a full message queue. It then verifies that a HPMC is received..

Class 9 Subtests

Subtests 9000-9050,9120-9170,9240-9290,9360-9410,9480-9530 - SCI MATS Test

Subtests 9000-9050,9120-9170,9240-9290,9360-9410,9480-9530 perform a MATS+ memory pattern test over the internode memory for each sci ring and all sci rings together. The MATS+ algorithm consists of several passes through memory, with intermixed reads and writes of two patterns. In this implementation of the algorithm, various 64 bit data patterns (2 per subtest) are chosen such that all data and ECC bits will be tested within the 64 bit word (over the course of all pattern subtests), and all detectable intra-longword dependencies will be detected. Figure 1 illustrates MATS+ algorithm.For each subtest, a pair of special patterns are used for MATS testing.

Subtests 9060-9110,9180-9230,9300-9350,9420-9470,9540-9590 - SCI NTA Test

Subtests 9060-9110,9180-9230,9300-9350,9420-9470,9540-9590 perform a NTA memory pattern test over the internode memory for each sci ring and all sci rings together. The Nair, Thattle, and Abraham algorithm consists of several passes through memory, with intermixed reads and writes of two patterns. Figure 2 illustrates the algorithm.For each subtest, a pair of special patterns are used for NTA testing.

Class 10 Subtest

Subtest 10000 - Coherency Stress Test

Subtest 10000 stresses the coherency circuit by performing random load/store/flush from each tchip.

Subtest 10010 - Messaging Stress Test

Subtest 10010 stresses the messaging hardware by sending a large amount of messages to another node.

Subtest 10020 - Messaging/Coherency Stress Test

Subtest 10020 stresses the coherency circuit by performing random load/store/flush and sending messages to another node.

Subtest 10030 - CCMC Stress Test

Subtest 10030 stresses the coherency circuit by performing random coherent and non-coherent operations.

Subtest 10040 - CCMC Messaging Stress Test

Subtest 10040 stresses the coherency circuit by performing random coherency and non-coherent operations and sending messages to another node.

Subtest 10050 - Load/Store One Line Stress Test

Subtest 10050 stresses the coherency circuit by performing random load and store on a globally shared block for all nodes.

Subtest 10060 - Mix Ops One Line Stress Test

Subtest 10060 stresses the coherency circuit by performing random coherent /non-coherent operations on a globally shared block for all nodes and sending messages to another node.

Subtest 10070 - Load/Store Two Line Stress Test

Subtest 10070 stresses the coherency circuit by performing random load and store on two globally shared blocks mapping on the same network cache for all nodes.

Subtest 10080 - Mix Ops Two Line Stress Test

Subtest 10080 stresses the coherency circuit by performing random coherent /non-coherent operations on two globally shared blocks mapping on the same network cache for all nodes and sending messages to another node.

Subtest 10090 - Lock Stress Test

Subtest 10090 stresses the semaphore circuit by performing lock operations.

Subtest 10100 - Load One Line Stress Test

Subtest 10100 stresses the coherency circuit by performing random load on a globally shared block for all nodes.

Subtest 10110 - Load Two Line Stress Test

Subtest 10110 stresses the coherency circuit by performing random load on two globally shared blocks mapping on the same network cache for all nodes.

Subtest 10150 - APA Mix Ops One Line Stress Test

Subtest 10150 stresses the coherency circuit and APA runway bus by performing random coherent /non-coherent operations on a globally shared block for all nodes and reading/writing APA STATUS0 csr.

Subtest 10160 - PMON Test#1

Subtest 10160 tests the PMON circuit.

Subtest 10170 - PMON Test#2

Subtest 10170 tests the PMON circuit.

Subtest 10190 - Lookaside

Subtest 10190 tests the SPP1200 lookaside circuit.

Subtest 10200 - Load Multi- Line Stress Test

Subtest 10200 stresses the coherency circuit by performing random load on multiple globally shared blocks mapping on the same network cache for all nodes.

Subtest 10210 - Load/Store Multi- Line Stress Test

Subtest 10210 stresses the coherency circuit by performing random load and store on multiple globally shared blocks mapping on the same network cache for all nodes.

Subtest 10220 - Mix Ops Multi-Line Stress Test

Subtest 10220 stresses the coherency circuit by performing random coherent /non-coherent operations on mutliple globally shared blocks mapping on the same network cache for all nodes and sending messages to another node.

Subtest 10300 - Load N Line Stress Test

Subtest 10300 stresses the coherency circuit by performing random load on number of user specified globally shared blocks mapping on the same network cache for all nodes.

Subtest 10310 - Load/Store N Line Stress Test

Subtest 10310 stresses the coherency circuit by performing random load and store on number of user specified globally shared blocks mapping on the same network cache for all nodes.

Subtest 10500 - OD_INV Stress Test

Subtest 10500 tests the SCI OD_INV condition.

Subtest 10510 - Incr/decr Stress Test

Subtest 10510 performs random increment and decrement of a false sharing block to check for data integrity.

Class 11 Subtests

Class 11 subtests are mainly used by manufacturing for initial board bringup. It does not require any tchip to be present.

Subtest 11000 - TS Address Bits Init

Subtest 11000 initializes tags by writing to the WRINIT csr for subtest 11010.

Subtest 11010 - TS Address Bits Uniqueness Quick Check

Subtest 11010 performs a quick uniqueness check on address bits. It does an uniqueness check on a 1KByte block of each address bit starting from 0x10000 to 0x20000000.

Class 12 Subtests

Class 12 subtests are mainly used by manufacturing for initial board bringup. It does not require any tchip to be present.

Subtest 12000 - TS Tag Column Functionality Test

Subtest 12000 tests tag column functionality. It performs a walking 1s and 0s test on the first word in each bank of tag (walking from least significant bit to most significant bit).

Subtest 12010 - TS Tag Uniqueness Test

Subtest 12010 tests tag location uniqueness. It writes an incrementing value to each location in tag up to 1Kbytes, then verifies that the locations contain the expected value.

Subtests 12020-12070 - TS Tag MATS Test

Subtests 12020-12070 perform a MATS+ memory pattern test over the tag upto 1Kbytes. The MATS+ algorithm consists of several passes through tag, with intermixed reads and writes of two patterns. In this implementation of the algorithm, various 64 bit data patterns (2 per subtest) are chosen such that all data and ECC bits will be tested within the 64 bit word (over the course of all pattern subtests), and all detectable intra-longword dependencies will be detected. Figure 1 illustrates MATS+ algorithm.For each subtest, a pair of special patterns are used for MATS testing.

Subtests 12080 -12130- TS Tag NTA Test

Subtests 12080-12130 perform a NTA memory pattern test over the tag upto 1Kbytes. The Nair, Thattle, and Abraham algorithm consists of several passes through tag, with intermixed reads and writes of two patterns. Figure 2 illustrates the algorithm.For each subtest, a pair of special patterns are used for NTA testing.

Subtests 12140- TS Tag Init

Subtests 12140 initailizes tag for class 13 subtests.

Class 13 Subtests

Class 13 subtests are mainly used by manufacturing for initial board bringup. It does not require any tchip to be present.

Subtest 13000 - TS Memory Column Functionality Test

Subtest 13000 tests memory column functionality. It performs a walking 1s and 0s test on the first word in each bank of memory (walking from least significant bit to most significant bit).

Subtest 13010 - TS Memory Uniqueness Test

Subtest 13010 tests memory location uniqueness. It writes an incrementing value to each location in memory up to 1Kbytes, then verifies that the locations contain the expected value.

Subtests 13020-13070 - TS Memory MATS Test

Subtests 13020-13070 perform a MATS+ memory pattern test over the memory upto 1Kbytes. The MATS+ algorithm consists of several passes through memory, with intermixed reads and writes of two patterns. In this implementation of the algorithm, various 64 bit data patterns (2 per subtest) are chosen such that all

data and ECC bits will be tested within the 64 bit word (over the course of all pattern subtests), and all detectable intra-longword dependencies will be detected. Figure 1 illustrates MATS+ algorithm.For each subtest, a pair of special patterns are used for MATS testing.

Subtests 13080 -13130- TS Memory NTA Test

Subtests 13080-13130 perform a NTA memory pattern test over the memory upto 1Kbytes. The Nair, Thattle, and Abraham algorithm consists of several passes through memory, with intermixed reads and writes of two patterns. Figure 2 illustrates the algorithm.For each subtest, a pair of special patterns are used for NTA testing.

Class 14 Subtest

Class 14 subtest verifies the SPP1200 processor dcache.

Subtest 14000 - HP Dcache Verification Test

Subtest 14000 tests SPP1200 processor dcache circuit.