Enterprise EXX00 Trouble Shooting and Diagnostics

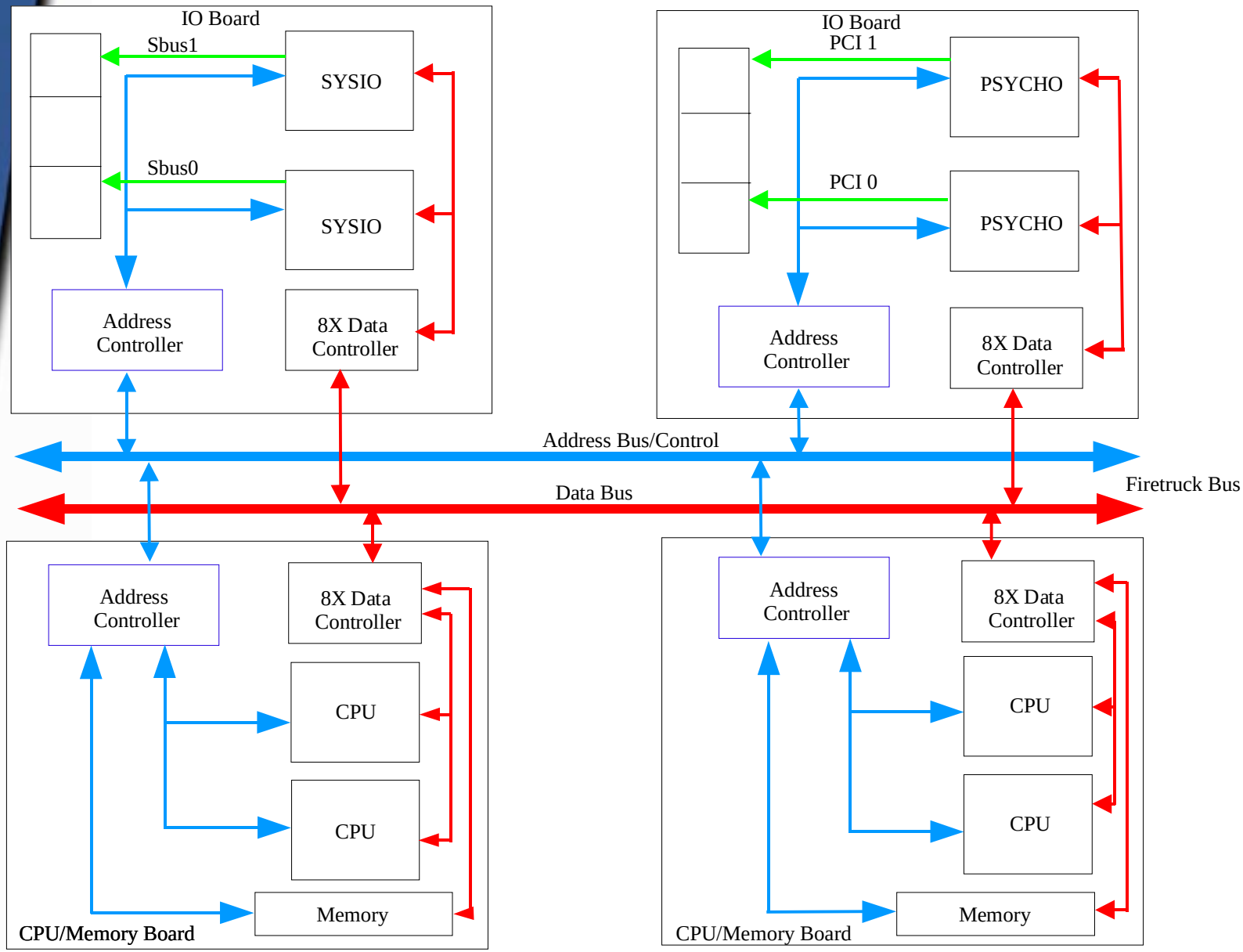
Jaffar Annab
Michel Volpi
SunfireClassic Diagnosability Team

SUN PROPRIETARY – INTERNAL USE ONLY

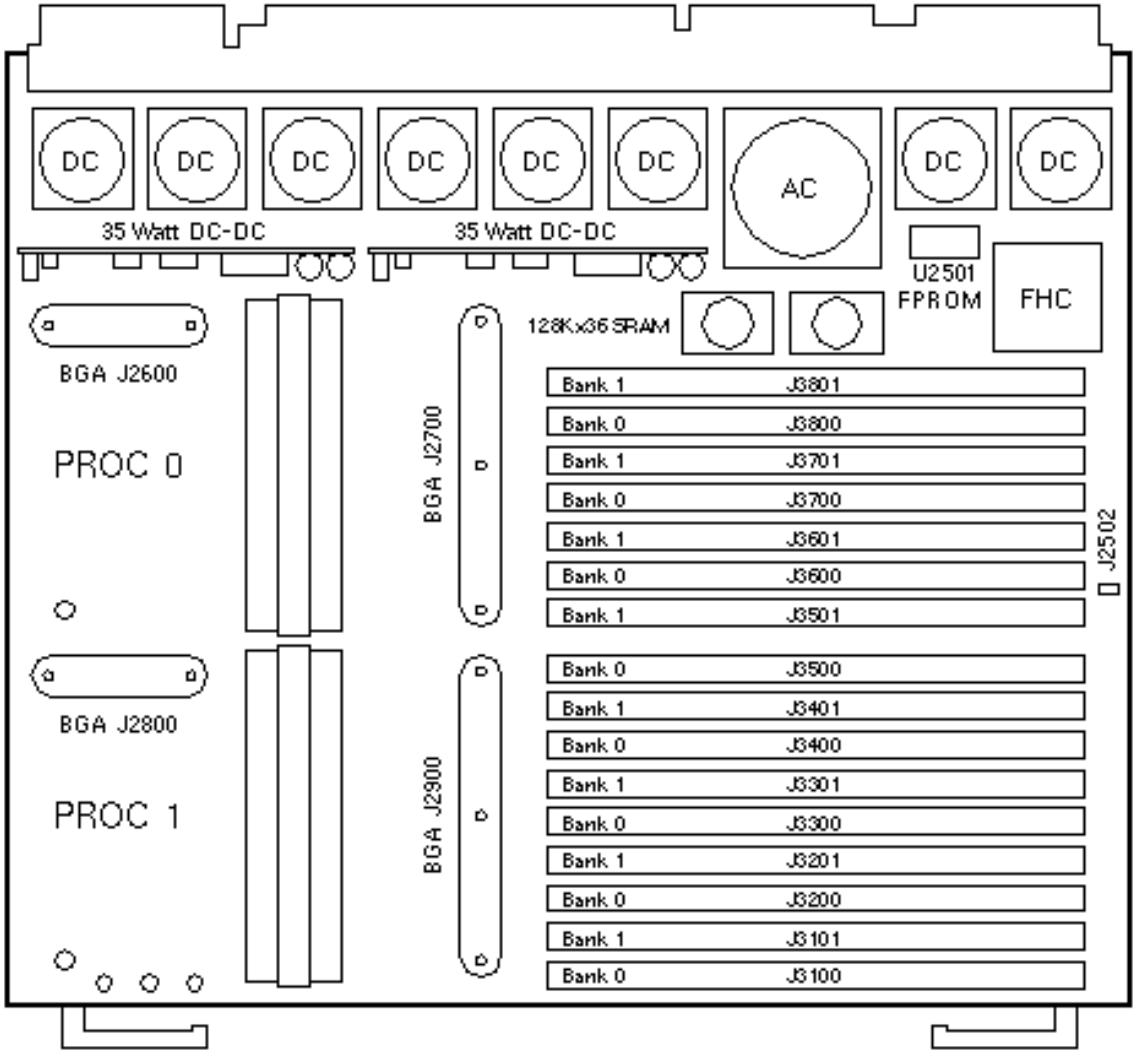
Enterprise EXX00 Diagnostics & Trouble Shooting

Enterprise System Architecture
Fatal Error Overview
Diagnostic Methods
Case Studies

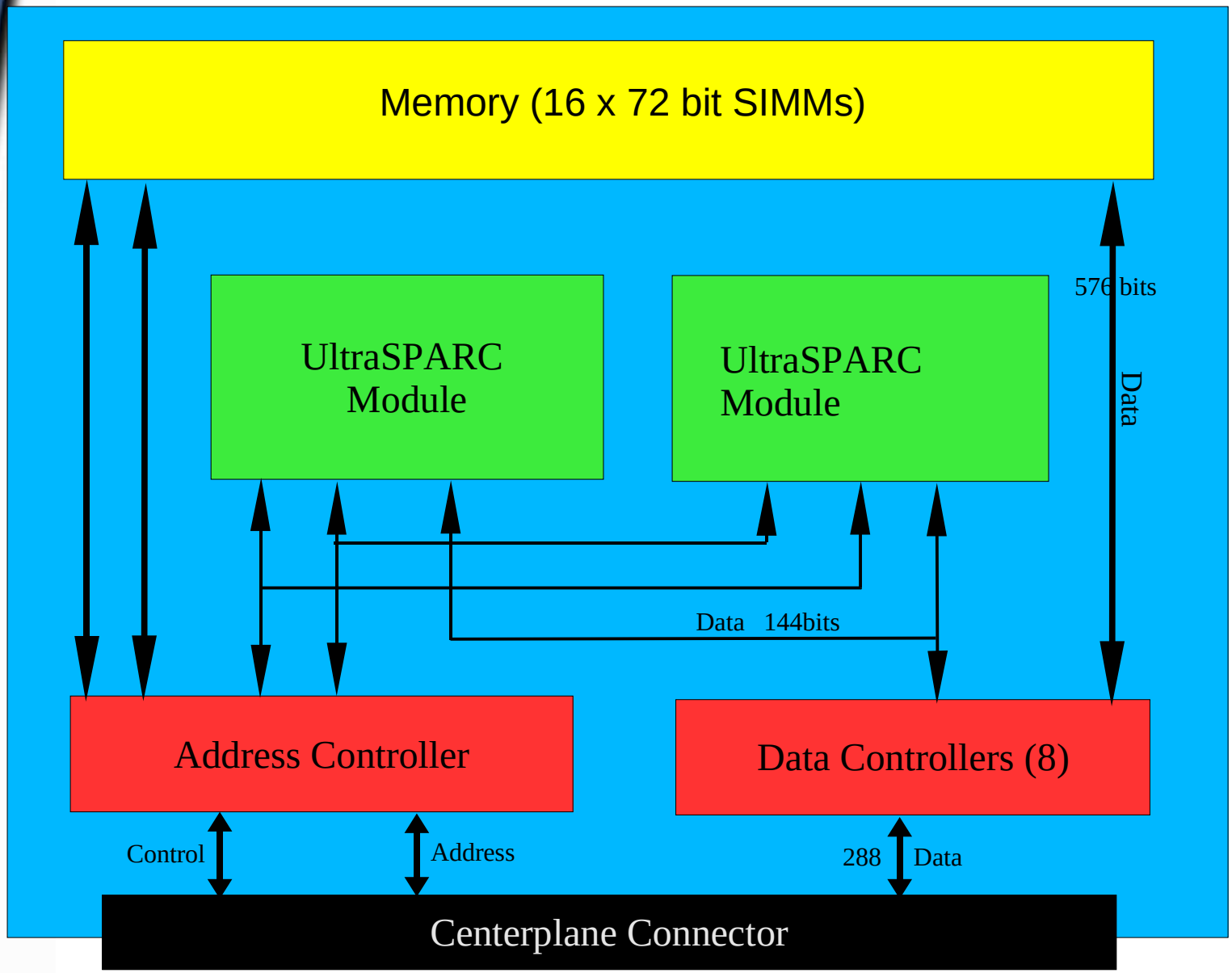
SunFire Logical Organization



CPU/Memory Board



CPU/Memory Board Block Diagram



CPU/Memory Board

Each CPU/Memory Board supports up to 2 UltraSPARC CPU modules and 16 SIMM slots for memory.

Each CPU/Memory board include an Address controller (AC), 8 bit-sliced Data Controllers (DC), 2 CPU slots (UPA A & B).

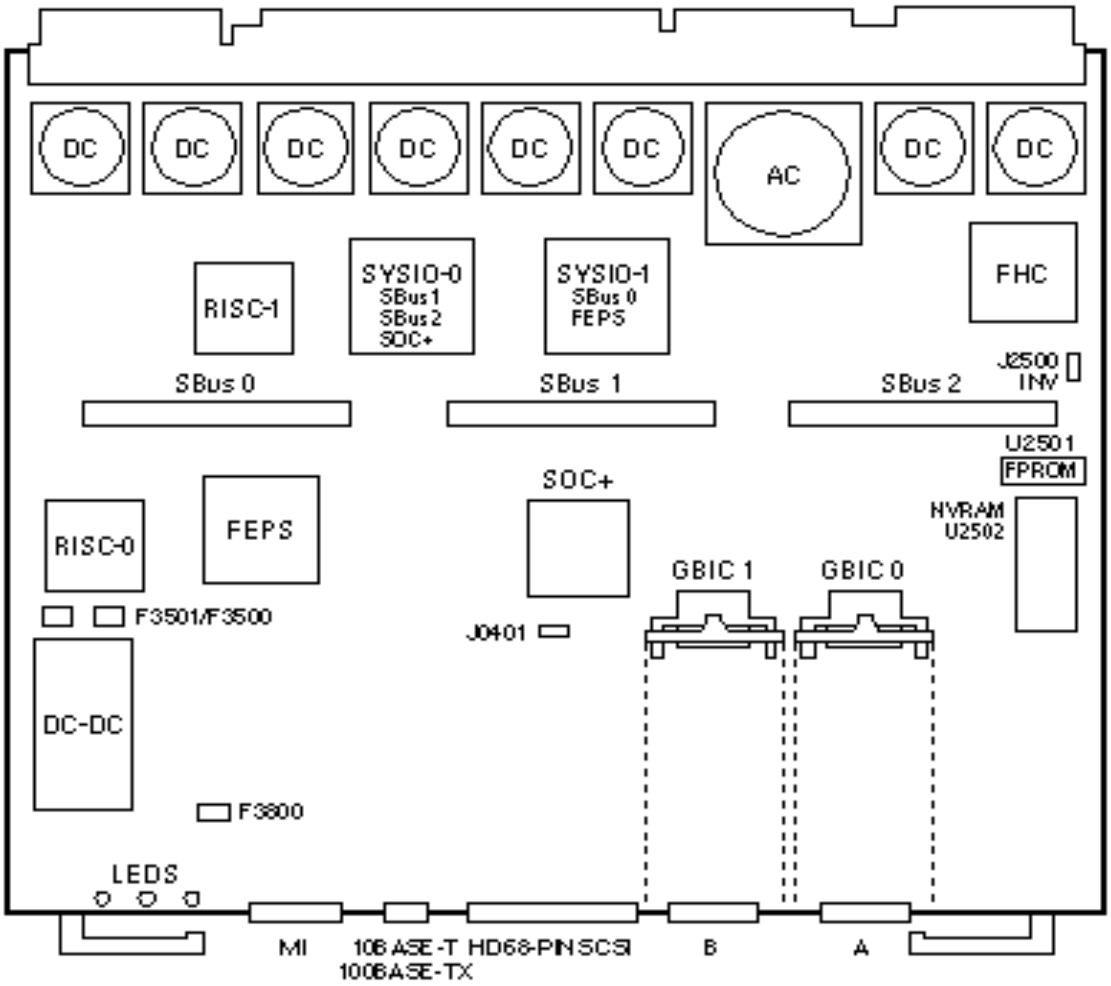
Gigaplane controller (FHC) and on-board devices that include a flash prom and SRAM.

DC has separate connections to the 2 CPUs and to memory.

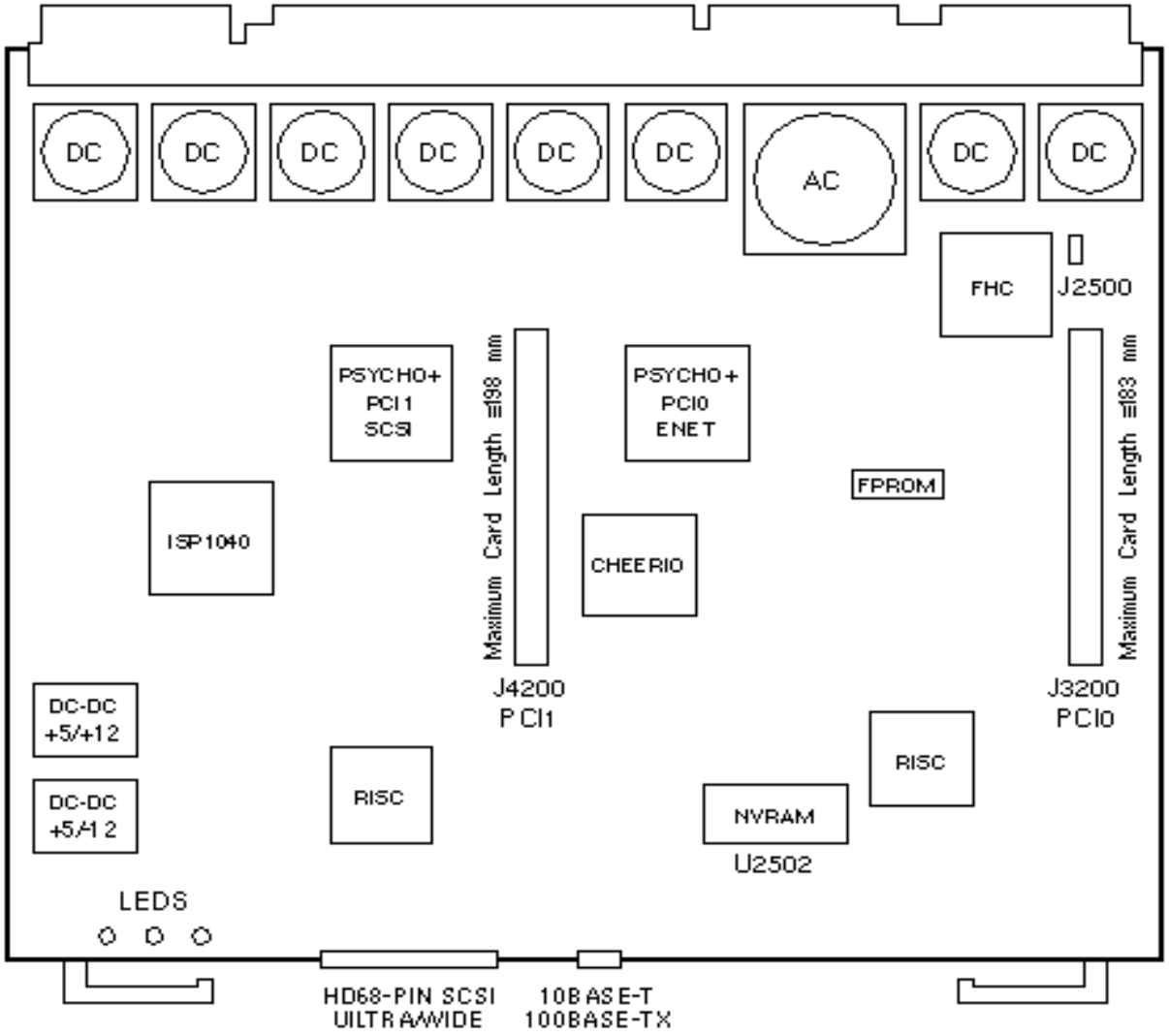
Memory is very wide (576 bits). Entire cache line can be transferred in a single clock cycle.

No special relationship between memory and CPUs on the same board. It is still necessary to go out to the centerplane to communicate.

I/O Board - SBUS



I/O Board - PCI


DC
DC
DC
DC
DC
DC
AC
DC
DC
FHC
J2500
PSYCHO+
PCI 1
SCSI
PSYCHO+
PCI0
ENET
Maximum Card Length =198 mm
Maximum Card Length =183 mm
FPROM
ISP1040
CHEERIO
J4200
PCI1
J3200
PCI0
DC-DC
+5/+12
DC-DC
+5/-12
RISC
RISC
NVRAM
U2502
LEDS
HD68-PIN SCSI
ULTRA/WIDE
10BASE-T
100BASE-TX

EXX00 I/O Boards

The I/O Boards come in 2 flavors: SBUS and PCI
SBUS boards come in two flavors: standard and graphics
SBUS I/O boards include two SYSIO ASICS that provide two Sbuses:

One for one plug-in SBUS card and 2 fibre interfaces

One for 2 plug-in SBUS cards and the on-board devices (10/100 MB TPE and Fast/Wide SCSI

PCI I/O boards include 2 PSYCHO+ ASICs that provide 2 PCI buses

One for PCI slot 0 and on-board 10/100 MB TPE

One for PCI slot 1 and on-board Ultra/Wide SCSI

The I/O FFB (Fast Frame Buffer) board includes one SYSIO that provides a single SBUS for 2 SBUS cards as well as those described for the I/O SBUS board

EXX00 Gigaplane (Centerplane)

Packet Switched Bus
Out of Order Completion
- Transactions don't tie up the bus
- 112 outstanding transaction can be active

Control/address wires can be used the at same time as data wires.
83/100 MHz speed, 83MHz only on E6500 and E5000 with 501-2978 clock board.
256-bit Data width.

EXX00 Fatal Error Overview

• Fatal Errors are detected on the Sunfire hardware by the Address Controller (AC) or the Data Controller (DC).

• Only errors causing an illegal hardware state will generate a reset, memory and SBUS cards fal into this category.

• A FATAL RESET cause the system to drop out of the running O/S or OBP, and perform a POST which handles the detection and mapping out of the failed component(s).

• **UPA Address Parity Error**

The UPA Address bus carries a parity bit. The UPA master port drives odd parity and parity errors can be detected by either the UPA slave or the AC.

<u>UPA Address Parity Error detected by UPA Port</u>

If the slave UPA port detects the parity error on the UPA Address Bus, it indicates this to the AC by issuing the P_FERR P_REPLY. The AC records this by setting the FERR bit in the AC-ESR.

<u>UPA Address Parity Error detected by AC</u>

If the AC detects the parity error on the UPA Address Bus, it sets the UPA_PERR bit in the AC-ESR.

• **FireTruck Address Parity Error**

An AC that detects a FireTruck Address Bus Parity sets the FTA_PERR bit in the AC-ESR

EXX00 Fatal Error Overview (Cont.)

- **Firetruck Control Parity Error**

  The Control signals on the FireTruck are also covered by parity signals. If an AC detects a FireTruck Control parity error, it set the FTC_PERR bit in the AC-ESR

. DTAG Parity Error

If the AC detects a parity error in the DTAGs(Duplicate Tags), it sets the DT_PERRA or DT_PERRB bit in the AC-ESR.

- **E-cache Tag Parity Error**

  This is detected by the CPU which then drives the P_FERR reply to signal this to the AC. The AC sets the FERR bit in the AC-ESR.

- **UPA Master Port Timeout**

  The AC has a timer associated with each UPA port. The timer for a port expires when there are outstanding requests from the port and no requests have been serviced for the timeout period. On detecting a master port timeout the MTIMEOUT bit in the AC-ESR is set.

- **Internal Error**

  Internal errors include coherence errors, queue overflows and errors indicated by the Data Controller

EXX00 Fatal Reset (Cont.)

- A Fatal Reset/Fatal Error condition is almost always caused by an un-recoverable hardware fault.
- When the system detects a FATAL RESET error all operations are immediately aborted.
- No corefile will be produced and no indication will be found in the /var/adm messages until after the next reboot where the message "system booting after fatal error FATAL" will be seen.
- When a Fatal Reset is detected, the system will reset and enter POST at maximum diagnostic level. The most important data which caused the Fatal Reset is displayed only to the **system console**.
- The Fatal Reset information can also been displayed using the "**prtdiag -v**" command.
- POST may disable the offending component so you may find some board deconfigured.

More on Fatal Errors

Fatal errors fall into the following major categories:
Address Controller Errors
CPU module Errors
I/O Board SYSIO errors

Here is a menu of the different keys that you can hit while the system is running Extended POST:

Key	Action
a	Toggle Pause CPU flag
c	Toggle Trace Test Case flag
d	Toggle DlSpeed Clear flag
l	Toggle Loop on Subtest flag
L	Toggle Loop on POST flag
e	Toggle Loop on Error flag
p	Toggle Print all Errors flag
v	Toggle Verbose Print Mode flag
s	**Toggle Stop Flag**
t	Toggle Timestamp flag
n	Skip to Next Subtest
N	Skip to Next Test
h or ?	Display this command summary

NOTE: After using any of these menu itemsI always do a reset-all and restart clean

Fatal Error Types

Address Controller Errors

Since the AC and the FHC are on the board the logical replacement here would be the called out board.

Fire Hose Controller	FHCERR	FHC, AC
General hardware	HWERR	AC
Board SRAM parity	RAMPERR:	AC
FireTruck Address or Control Parity	FTA_PERR, FTC_PERR	If detected by only one: backplane pins, board connector pins, AC If detected by all ACs: backplane pins, board connector pins, backplane terminator, any AC If detected by some, but not all ACs: backplane terminators Many times this is an artifact of the real cause
FireTruck Share, or Arbitration	FT_SHRERR, FT_ARBERR, FT_DIDERR	backplane pin, connector pin, AC
Ultra Port	UPA_ERR	CPU module connector, FFB, CPU, AC, Sysio
Board SRAM Dtag	DT_PERRA DT_PERRB	Dtag A, AC Dtag B, AC As the DTAG is on the CPU/MEM board - replace the board on either of these errors
FireTruck to UPA overflow	FTUPAOV	backplane, board connector, AC, any AC
UPA to FireTruck overflow	FOUPAOV	AC, any AC
Queue OverFlow Errors	PREQOV LPQOV FPQOV CIQOV	CPU, AC AC AC, any AC backplane, board connector, AC, any AC
Transaction Timeout	MTIMEOUT SLEEPERR	any board (target of operation), AC software, AC
Fire Truck Read and Write Transactions	READVV READSVICT READVICT WBSVICT TAGVICT READTAG WBSTAG	AC, cpu AC, cpu AC, cpu AC, cpu AC, Dtag, cpu AC, Dtag, cpu AC, Dtag, cpu

Fatal Error Types (Cont.)

CPU Module Errors:

Module parity, ECache Tag parity, Ecache Data parity	IPREP FERR	cpu module, AC
System ECC error:		any board (source of data)

Sysio errors:

PIO access parity, DVMA access parity		SBus card connector, SBus card, Sysio
ECC:		any board (source of data)

EXX00 Troubleshooting and Diagnostics

Part 2

Diagnostic Tools and Techniques

Fatal Error Diagnostic Tools and Techniques

The most important attributes in good trouble shooting are

1. Determine what the problem(s) you are trying to solve really is
2. What facts do we know
3. What facts do we need to know
4. What are the possible causes
5. Do the facts we have allow the possible cause to standout
6. Not jumping to cause

Fatal Error Diagnostic Tools and Techniques (Cont.)

Background

Large Enterprise Servers can prove difficult to diagnose if some of the following conditions exist:

1. Multiple problems
2. Limited down time
3. Imprecise diagnostics output
4. Inability to pin-point failure

While such failures are rare, they do inflict considerable pain on the customer concerned and cost to Sun.

Fatal Error Diagnostic Tools and Techniques (Cont.)

Triggers for use

A number of triggers are suggested

1. One attempt at fix has failed to rectify the problem reported by the customer
2. Multiple problems are suspected on the system and we do not know for sure what they are
3. The system has a history of previous failure
4. We can not describe in 1 sentence what the problem is with the system
5. We are not 100% sure it is a hardware problem or we could have a class issue where a genuine product defect means that component replacement like-for-like will not resolve the issue at hand.

Fatal Error Diagnostic Tools and Techniques (Cont.)

System Console

Attach a console which can keep a persistent log of the console output. This is vital for catching failure messages while the customer has the system in production or during stress testing and also for capturing the extended POST log.

Sun Workstation, PC or a laptop attached to serial port A on the system

Use tip (on Sun), or hyperterm and be sure to save the output to a file that can be reviewed

ControlTower by Aurora Technologies is an excellent 3rd party product that provides console management tools for Sun systems

Fatal Error Diagnostic Tools and Techniques (Cont.)

What do we know?

State in order, with dates and time if possible, all events that we know of since the first failure occurred. Also look at least 1 month prior to the first failure.

Do we have crash dumps, panic messages, POST logs? What do they tell us and have we really asked the right person yet as to what they mean? Keep in mind that crash dumps and panic messages are not present for FATAL resets.

All future changes (hardware or software) must be recorded and all involved should be made aware of any change in advance.

What is the behavior of other customer systems at the same site over the past year. The intent here is to expose site based problems such as the environment and power.

What is the system doing when it fails?

Fatal Error Diagnostic Tools and Techniques (Cont.)

Hardware Baselining

Undertake a hardware audit which will require a limited amount of downtime.

1. What parts are in the system and where did they all come from?
2. Are all the parts qualified for that system?
3. Which parts have been installed/changed prior to or after the first failure?
4. Who has done the recent work and why did they do it?
5. What were the last parts to be added/changes on the system?
6. Check that all Field Replaceable Units part numbers in the system are compatible with the system type by the Sun System Handbook (URL here).
7. Ensure all parts are in good visual order
8. Components such a load boards, sufficient power supplies are in place.

Aim for minimal change/disturbance of hardware.
No actions should be taken such as cleaning/re-torquing CPU's at this stage.

The configuration which the system left the factory can be checked on
http://sfepdpkg2.ebay:8000/request/printcis For EXX00 systems
http://bops.west/Starfire/ManufacturingTools/ReprintFruList.html For E10k systems

An amber light on a board does not always mean it is faulty and is the root cause of the problem in hand. Disk boards in EXX00 systems always have the amber light on!

Fatal Error Diagnostic Tools and Techniques (Cont.)

Extended Post

Execute 5 loops of extended POST. Most faulty hardware will be identified using Extended POST. This step must not be avoided and should be done as soon as a potential multi-problem system is identified. This may take some time to run(the bigger the system, the longer it takes), but is an essential baseline.

The diag switch must be set to the diag position and power cycle the system.

At the OBP, set the **configuration-policy** to be board. Make sure that all the post logs are reviewed, not just that post reports a clean system. Post may report failures or warnings, but still configure the component.

If POST detects a suspect component, disable the board at the OBP and rerun POST. This would be positioned as the first down time slot if this has not been done before with *all* the parts which are in the system at this time.

Use **disabled-board-list** OBP variable to disable a board in the system without physically removing it.

setenv disabled-board-list 2 4 a

will disable boards 2 4 and 10 in the system

Fatal Error Diagnostic Tools and Techniques (Cont.)

Explorer

Take a current explorer and review its contents. Files which should be reviewed in detail include

- sysconfig/prtdiag-v.out - Inconsistencies
- sysconfig/eeprom.out - OBP setting are suitable for hardware debugging
- messages/* - Interesting messages starting with error, warning or panic for example
- etc/init.d/syssetup - check crash dump configuration
- etc/system - Look for unreasonable settings

A number of good tools exist for analysis of explorer. The best known examples is Sentinal(http://spider.aus/sentinal).
Take the explorer itself and/or the output of such a tool and look for obvious anomalies such as over temperature boards, firmware revisions which are very out of date, parts which have open FIN's against them.

In addition, http://otis.uk has a number of tools that can be used in FATAL Error analysis

Explorer gives an idea of the overall state of the system, and is useful in pushing back on the customer where their administration of the system in question is not what it might and/or should have been.

Fatal Error Diagnostic Tools and Techniques (Cont.)

Stress testing

If the root cause of the problem has not been found by this stage, then consideration must be given to reducing the configuration of the system and stressing the system in such as way as to reproduce the problem.

It is essential that any stress tests are only run with the system in question out of production and with data filesystems unmounted.

Additional message reporting

Prior to starting stress tests, add the following entries in `/etc/system` to enable more verbose reporting of errors by the kernel and then reboot.

- *Not all settings are valid on all versions of Solaris, so some warnings may be given out at boot time.*

- *These must be removed before the system goes back into production as they can have an adverse effect on system performance.*

Fatal Error Diagnostic Tools and Techniques (Cont.)

Additional message reporting

Prior to starting stress tests, add the following entries in /etc/system to enable more verbose reporting of errors by the kernel and then reboot.

Variable	***Purpose***
set snooping=1	Enable the deadman kernel - useful in debugging hangs
set obpdebug=1	Enable kernel symbolic addresses to be used in OBP
set forthdebug=1	Enable kernel debugging from OBP
set ce_debug=1	Enable correctable error logging
set ce_show_data=1	Show the ECC data/cache line
set ce_verbose=1	Be verbose about ECC errors
set report_ce_console=1	Report correctable ECC errors to the system console
set report_ce_log=1	Report correctable ECC errors to the system log files
set ue_debug=1	Enable uncorrectable error logging
set sd:sd_error_level=1	Set the scsi disk error reporting to 1
set max_ce_err=1	Max number of ECC correctable errors before starting report

Fatal Error Diagnostic Tools and Techniques (Cont.)

Strategy - Divide and Conquer

The divide and conquer strategy allows the field engineer to home in on the faulty components by eliminating good components as suspects.

This approach is valid for systems with a single faulty component, particular if the reporting of the failure appears to move between components. However, it is often the only way to proceed when a system has multiple faults (often combined hardware and software). Finding the root cause of one problem does not mean that all root causes have been found.

It is essential to remember that the component that reports a failure is not necessarily the component which is the root cause of the failure.

The Field Replaceable Unit (FRU) must be considered to be the system board level. Individual system boards must be off-lined at the OBP by setting the disable-board-list OBP property. This avoids the potential of introducing additional problems through contact with the system. DO NOT REMOVE BOARDS.

Fatal Error Diagnostic Tools and Techniques (Cont.)

Strategy - Divide and Conquer(cont.)

The "divide and conquer" strategy requires a log to be made of all changes. No hardware should be physically removed/changed in the system until a firm failure is narrowed down to a single board. When a problem board is identified, it should be replaced with a "warm" board which has been run-in on a lab system and then tested with extended POST and suitable period of stress testing.

Set the configuration-policy for POST to be board.

```
setenv configuration-policy board

setenv disabled-board-list 45ef (disable boards in slot 4, 5, 14 and 15)
```

To reset a list to null, enter the following at the ok prompt:
ok set-default disabled-board-list

These variables take effect on the next reset or power-on.

Fatal Error Diagnostic Tools and Techniques (Cont.)

Tidyup

Remember to reset all values to their previous (unless we know them to be wrong and the customer is both aware and in agreement).

Class issues

Once a individual system has been diagnosed and is running without fault, it is important to identify other systems which may be exposed to the same root cause with that customer an other customers.

The action will depend on the problem type.

If a product defect (as opposed to a faulty component) is possible, an escalation to CTE should be raised to investigate this further. The triggers for this may include

1. Multiple systems showing the same problem across a range of customers or individual customer sites.
2. New systems having a high failure rate

Fatal Error Diagnostic Tools and Techniques (Cont.)

Stress Testing Tools

Caveat

If you have not used these tools in a safe environment such as a lab, then don't use them on a customer site.

Diagnostics Clearing House

http://dch.sfbay You need to register before you can get access to this site.

General

Tests for both I/O and CPU memory

SYSTEST

This is a general load generator available from http://otis.uk. It should not be used on customer's production systems as it tends to fill the root filesystem up.

SUNVTS

SunVTS provides hardware fault detection, fault isolation and system exercise capabilities

http://diagnosis.eng/sunvts/sunvts_intro.shtml

/bin/compress

Put a large file in /tmp and run compress and uncompress on it in a loop. Large is at least as big as physical memory.

Fatal Error Diagnostic Tools and Techniques (Cont.)

Stress Testing Tools(cont.)

CPU/Memory

SETI
SETI@home is a scientific experiment that uses Internet-connected computers in the Search for Extraterrestrial Intelligence (SETI).

It is very good at keeping CPU's busy.

http://setiathome.ssl.berkeley.edu/

CPU and Memory

These are cache-specific tests which are multi-threaded, with each thread binding to different CPUs and then running the specific tests. The same is true for the dcachtest1.c and dcachtest2.c.

http://dch.sfbay

Checkout VLINPAK it is also a good tool for stress testing the CPU.

Fatal Error Diagnostic Tools and Techniques (Cont.)

Stress Testing Tools(cont.)

SSO

Tests all of the CPU modules in the system simultaneously. The diagnostics are executed in an order where the most diagnostics which are the most likely to fail are executed first and the most often. The actual run times of the individual diagnostics are scaled based on CPU frequency and external cache size in order to keep the actual run times relatively constant.

The most likely failure mode of these diagnostics is a parity error. When this occurs it will cause the system to panic. The CPU which reports the parity error should then be blacklisted or removed before the system is rebooted and the test restarted. All failures will be found sequentially, with each failure requiring a system reboot.

SSO runs a sequence of utilities including Linpack.

****WARNING: Do not use SSO on 501-4995 processors as good modules will give eache parity errors. Running SSO requires examining the processors first to determine the part number. ****

****WARNING: Do not change the processor speed as suggested in the SSO documentation ****

Available from http://dch.sfbay

Fatal Error Diagnostic Tools and Techniques (Cont.)

Stress Testing Tools(cont.)

/bin/savecore -L

The goal here is to scan all of memory every few minutes

1. Delete the swap devices with swap -d
2. Set up the dump device as a partition different from the swap partition which is just larger than physical memory. This might be a challenge on some systems.

```
/usr/sbin/dumpadm -d /dev/dsk/cnnnnnnn
```

3. Set the system to dump the whole of physical memory used, not just the kernel.

```
/usr/sbin/dumpadm -c all
```

4. Run this script

```
#!/bin/ksh
while true
do
   date
   /bin/savecore -L
   /bin/rm /var/crash/`uname -n`/*
   sleep 300 # system does other stuff
done
```

An alternative is wc -c /dev/mem

Fatal Error Diagnostic Tools and Techniques (Cont.)

Stress Testing Tools(cont.)

I/O Stress Tools

/bin/dd

The aim is to read across each controller in the system many times. As an example for a system with 3 I/O controllers. Pick any disk on these controller, but all controllers must be covered, not all disks.

```
#!/bin/ksh
while true
do
   dd if=/dev/dsk/c0t0d0s0 bs=8k of=/dev/null &
   dd if=/dev/dsk/c1t0d0s0 bs=8k of=/dev/null &
   dd if=/dev/dsk/c2t0d0s0 bs=8k of=/dev/null &
   wait;
done
```

Block size (bs) can be varied between 0 and 1 meg.

Fatal Error Diagnostic Tools and Techniques (Cont.)

Stress Testing Tools(cont.)

I/O Stress Tools

Diskomizer

Diskomizer is a program for testing and verifying disk subsystems. It uses multiple processes to do asynchronous writes and reads to devices that are specified and then verifies the data that is read back is the data that was written to that block. Every block of data has a unique header each time it is written and the body of the data changes every time the block is written.

Diskomizer will find broken devices and paths to devices software bugs and latent faults in hardware. It does not break these devices, it simply finds faults that are already there. It knows nothing about the under lying storage devices, and hence can be used just as well to generate load on NFS file systems as any other device.

Diskomizer can be run as an ordinary user as long as that user has permission to open the files and or devices that are being used.

http://cte-www.uk.sun.com/diskomizer/latest/diskomizer.html

Substantial system resources may be required to drive a large storage subsystem.

Diskomizer is generally used for destructive testing.

Fatal Error Diagnostic Tools and Techniques (Cont.)

Stress Testing Tools(cont.)

I/O Stress Tools

Stortools: dex32

The dex32 program generates I/O transactions to a device.The command-line options allow you to specify:

- Test information including the type of test, test-related flags, and the number of I/O transactions per pass
- The range across the media to test
- The number of passes to run
- The type of I/O transaction
- Whether to stop after a certain number of errors or an elapsed time frame
- The data pattern used during testing

You can use dex32 to exercise disk drives, tape drives, file systems, and memory. dex32 is best run in conjunction with Stortools to flush out any problems with FCALtype storage.

Run sequential test, write only, across 100 Mbytes on the specified device. The test performs eight I/O transactions per pass.

```
dex32 -S8 100m -w /dev/rdsk/c1t2d0s6
```

Fatal Error Diagnostic Tools and Techniques (Cont.)

Stress Testing Tools(cont.)

I/O Stress Tools

Network

It has been found useful that when running CPU intensive stress tests to also generate some periodic network traffic. This will interrupt the processor execution and has been known to expose certain types of CPU and network card problems.

Modify this script to suit your requirements

```
#!/bin/ksh
while true
do
   spray <host under test>
   sleep 1
done
```

Case Studies

Here is an example of a FATAL error that can lead you down the wrong track:

```
18-MAR-2001 06:09:05.17 Fatal Reset
18-MAR-2001 06:09:06.55 0,0>FATAL ERROR
18-MAR-2001 06:09:07.00 0,0>    At time of error: System software was running.
18-MAR-2001 06:09:07.00 0,0>    Diagnosis: Board 0, centerplane pin, connector pin, AC
18-MAR-2001 06:09:07.22 0,0>    Diagnosis: Board 2, UPA PORT Device, AC
18-MAR-2001 06:09:07.23 0,0>Log Date: Mar 18 11:15:33 GMT 2001
18-MAR-2001 06:09:07.23 0,0>
18-MAR-2001 06:09:07.23 0,0>RESET INFO for CPU/Memory  board in slot 0
18-MAR-2001 06:09:07.44 0,0>    AC ESR 00000400.00000000 FT_ARBERR
18-MAR-2001 06:09:07.44 0,0>    DC[0] 00
18-MAR-2001 06:09:07.44 0,0>    DC[1] 00
18-MAR-2001 06:09:07.44 0,0>    DC[2] 00
18-MAR-2001 06:09:07.45 0,0>    DC[3] 00
18-MAR-2001 06:09:07.45 0,0>    DC[4] 00
18-MAR-2001 06:09:07.45 0,0>    DC[5] 00
18-MAR-2001 06:09:07.45 0,0>    DC[6] 00
18-MAR-2001 06:09:07.45 0,0>    DC[7] 00
18-MAR-2001 06:09:07.66 0,0>    FHC  CSR 00050200 LOC_FATAL SYNC NOT_BRD_PRES
18-MAR-2001 06:09:07.66 0,0>    FHC RCSR 02000000  FATAL
```

.
.
.

Continued on the next slide....

Case Studies

```
18-MAR-2001 06:09:07.67 0,0>RESET INFO for CPU/Memory  board in slot 2
18-MAR-2001 06:09:07.67 0,0>    AC ESR 00000000.00600001 IPREP FERR UPA_A_ERR
18-MAR-2001 06:09:07.88 0,0>    DC[0] 00
18-MAR-2001 06:09:07.88 0,0>    DC[1] 00
18-MAR-2001 06:09:07.88 0,0>    DC[2] 00
18-MAR-2001 06:09:07.88 0,0>    DC[3] 00
18-MAR-2001 06:09:07.88 0,0>    DC[4] 00
18-MAR-2001 06:09:08.10 0,0>    DC[5] 00
18-MAR-2001 06:09:08.10 0,0>    DC[6] 00
18-MAR-2001 06:09:08.10 0,0>    DC[7] 00
18-MAR-2001 06:09:08.11 0,0>    FHC  CSR 00050030 LOC_FATAL SYNC BRD_LED_M BRD_LED_R
18-MAR-2001 06:09:08.11 0,0>  FHC RCSR 02000000  FATAL
```

In this situation POST flagged CPU/MEM board 0 and placed it in low power mode, but if you look at the actual error on board 2, you will see that an error occurred on UPA A which is CPU module 0.

As it turns out CPU 0 on board 2 suffered a ETAG parity error, board 0 was a victim in this case. The action is to replace CPU0 on board 2, always following the Best Practices Guide for CPU replacement.

Case Studies

UPA_A_ERR (Etag Parity Error)

Fatal Reset
0,0> FATAL ERROR
0,0> At time of error: System software was running.
0,0> Diagnosis: Board 0, UPA PORT Device, AC
0,0> Log Date: May 22 12:02:03 GMT 2000
0,0> RESET INFO for CPU/Memory board in slot 0
0,0> AC ESR 00000000.00600001 IPREP FERR UPA_A_ERR
0,0> DC[0] 00
0,0> DC[1] 00
0,0> DC[2] 00
0,0> DC[3] 00
0,0> DC[4] 00
0,0> DC[5] 00
0,0> DC[6] 00
0,0> DC[7] 00
0,0> FHC CSR 00050200 LOC_FATAL SYNC NOT_BRD_PRES
0,0> FHC RCSR 02000000 FATAL

Comments:
Fatal Reset Errors that have the IPREP and FERR error bits set in the Address Controller Error Status Register indicate that an Ecache Tag parity Error may be the cause of the Fatal Reset. In the typical error message shown above, the failure can be diagnosed to be an Ecache Tag Parity Error on Board 0 CPU 0.

The Sun UltraSPARCSII Processor system CPUs will send a P_ERR P_REPLY to the AC for the following two conditions;

- Parity error on UPA address bus while AC is bus master & CPU is the slave.
- E-Cache tag parity error. This is not reported as a trap like an E-Cache data error because system coherence is lost for this condition and the system must be reset.

Case Studies

UPA_A_ERR (Etag Parity Error) Multiple Errors

Fatal Reset

7,0>FATAL ERROR
7,0> At time of error: System software was running.
7,0> Diagnosis: Board 9, UPA PORT Device, AC
7,0> Diagnosis: centerplane terminators
7,0>Log Date: Jan 19 13:54:45 GMT 2000
7,0>
7,0>RESET INFO for IO Type 4 board in slot 1
7,0> AC ESR 00002000.00000000 FTA_PERR
7,0> DC[0] 00
7,0> DC[1] 00
7,0> DC[2] 00
7,0> DC[3] 00
7,0> DC[4] 00
7,0> DC[5] 00
7,0> DC[6] 00
7,0> DC[7] 00
7,0> FHC CSR 00040000 LOC_FATAL
7,0> FHC RCSR 02000000 FATAL
7,0>RESET INFO for IO Type 4 board in slot 3
7,0> AC ESR 00002000.00000000 FTA_PERR
7,0> DC[0] 00
7,0> DC[1] 00
7,0> DC[2] 00
7,0> DC[3] 00
7,0> DC[4] 00
7,0> DC[5] 00
7,0> DC[6] 00
7,0> DC[7] 00
7,0> FHC CSR 00040000 LOC_FATAL
7,0> FHC RCSR 02000000 FATAL

Diagnosis # 2 - UPA_A_ERR (Etag Parity Error) Multiple Errors (continued)

7,0>RESET INFO for CPU/Memory board in slot 7
7,0> AC ESR 00002000.00000000 FTA_PERR

Case Studies

UPA_A_ERR (Etag Parity Error) Multiple Errors (continued)

```
7,0>RESET INFO for CPU/Memory board in slot 7
7,0> AC ESR 00002000.00000000 FTA_PERR
7,0> DC[0] 00
7,0> DC[1] 00
7,0> DC[2] 00
7,0> DC[3] 00
7,0> DC[4] 00
7,0> DC[5] 00
7,0> DC[6] 00
7,0> DC[7] 00
7,0> FHC CSR 00050200 LOC_FATAL SYNC NOT_BRD_PRES
7,0> FHC RCSR 02000000 FATAL
7,0>RESET INFO for CPU/Memory board in slot 9
7,0> AC ESR 00000000.00600002 IPREP FERR UPA_B_ERR
7,0> DC[0] 00
7,0> DC[1] 00
7,0> DC[2] 00
7,0> DC[3] 00
7,0> DC[4] 00
7,0> DC[5] 00
7,0> DC[6] 00
7,0> DC[7] 00
7,0> FHC CSR 00050030 LOC_FATAL SYNC BRD_LED_M BRD_LED_R
7,0> FHC RCSR 02000000 FATAL
```

Comments:

The error on CPU/Memory Board 9 is either a CPU (location 1 / B port) Etag parity error or an incoming system address parity error on the UPA. Engineering experience says that it is most likely a CPU Etag (90%) as an ISAP would likely be accompanied by the bit UPA_PERR.

The FTA_PERRs reported by the other slots are likely artifacts.

Case Studies

DT_PERR (DTAG Parity Error)

Fatal Reset
0,0>FATAL ERROR
0,0> At time of error: System software was running.
0,0> Diagnosis: Board 6, Dtag A (UPA Port 0), AC
0,0>Log Date: Dec 17 22:20:15 GMT 2000
0,0>
0,0>RESET INFO for CPU/Memory board in slot 6
0,0> AC ESR 00000020.00000000 DT_PERRA
0,0> DC[0] 00
0,0> DC[1] 00
0,0> DC[2] 00
0,0> DC[3] 00
0,0> DC[4] 00
0,0> DC[5] 00
0,0> DC[6] 00
0,0> DC[7] 00
0,0> FHC CSR 00050030 LOC_FATAL SYNC BRD_LED_M BRD_LED_R
0,0> FHC RCSR 02000000 FATAL
0,0> Config policy change

Comments:
The DT_PERRA in slot 6 indicates a Duplicate Tag SRAM (DTAG) parity error. These DTAG SRAMs reside on the CPU/Memory boards.

Case Studies

Here is another example:

Here the error was due to a faulty cpu on UPA port B on system board 0. This was verified by placing the cpu on a new board on UPA port A and seeing the same error however UPA_B_ERR became UPA_A_ERR

```
Fatal Reset
0,0>FATAL ERROR
0,0>    At time of error: System software was running.
0,0>    Diagnosis: Board 0, AC, CPU, Dtags
0,0>Log Date: Apr  9  8:56:22 GMT 2000
0,0>
0,0>RESET INFO for CPU/Memory  board in slot 0
0,0>    AC ESR 00000000.00001002 TAGVICT UPA_B_ERR
0,0>    DC[0] 00
0,0>    DC[1] 00
0,0>    DC[2] 00
0,0>    DC[3] 00
0,0>    DC[4] 00
0,0>    DC[5] 00
0,0>    DC[6] 00
0,0>    DC[7] 00
0,0>    FHC  CSR 00050200 LOC_FATAL SYNC NOT_BRD_PRES
0,0>    FHC RCSR 02000000  FATAL
0,1>
```

References and Sites for more information

More Case studies, analysis and troubleshooting tips can be found in:

http://infoserver.central/data/910/910-4188/pdf/

A great set of tools for analyzing FATAL errors and others can be found at

http://otis.uk

Navigator is another tool that can be used for explorer data analysis

http://web.central/Navigator/index.html

Diagnostic Clearing House has an extensive list of diagnostic tools for various Sun systems

http://dch.sfbay.sun.com