

Fire 2.0/2.1 Chip Errata



Date	Comments
23 Jul 2007	External Release 1.0

Products Rights Notice:

Copyright © 1991-2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95054, U.S.A. All Rights Reserved

You understand that these materials were not prepared for public release and you assume all risks in using these materials. These risks include, but are not limited to errors, inaccuracies, incompleteness and the possibility that these materials infringe or misappropriate the intellectual property right of others. You agree to assume all such risks.

THESE MATERIALS ARE PROVIDED BY THE COPYRIGHT HOLDERS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS (INCLUDING ANY OF OWNER'S PARTNERS, VENDORS AND LICENSORS) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THESE MATERIALS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Sun, Sun Microsystems, the Sun logo, Solaris, OpenSPARC T1, OpenSPARC T2 and UltraSPARC are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd. The Adobe logo is a registered trademark of Adobe Systems, Incorporated. Part of the products covered by these materials may be derived from the Berkeley BSD systems licensed by the University of California. Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product described in these materials. This distribution may include materials developed by third parties who have intellectual property rights therein. Products covered by and information contained in these materials may be controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists may be prohibited.



Usage Information

Identifying Fire Revisions

The JBus Device ID Register, which is at offset 0x00.0000, is used to identify Fire devices and determine their revision.

The device identifying fields of this register reflect values actually contained in the JTAG Device ID Register, located in the DFT TAP controller (Design For Test, Test Access Port). The JTAG Device ID Register is accessible via the IEEE 1149.1 JTAG access port in Fire.

Within the JBus Device ID Register there are 3 fields pertaining to Fire identification and revision:

- MID: Manufacturer ID
- MT: Module Type
- MR: Module Revision

Each of these fields is actually a subset of the fields contained in the JTAG Device ID Register. Mapping between these two regs is as follows.

Table 1 JBus and JTAG Device ID mapping (Fire 2.0 values shown)

JBus Device ID Register		JTAG Device ID Register ¹	
Register	Value	Register	Value
MID[15:10]	0x36	MID[11:1]	0x036
MT[9:4]	0x22	PartNum[27:12]	0x1122
MR[3:0]	0x03	Version[31:28]	0x03

1. JTAG Device ID Register[0] is reserved, and set to '1'

See the Fire DFT Programmer's Reference for more information on the JTAG Device ID register.



Chip Revision Summary

Table 2 enumerates the values of the fields in the Device ID registers and the revision history of parts in the Fire family.

Table 2 Fire Family Revision Identification

Part	Revision	Silicon Available	MR	MT	MID	JTAG ID REG	Type ¹	Identifiable Markings
Fire	2.0	05/17/05	0x03	0x22	0x36	0x3112206D	PB	Sun Part # 528-1122-02
							ROHS	Sun Part # 528-1140-02
							PB Free	Sun Part # 528-1141-02
Fire	2.1	09/10/06	0x04	0x22	0x36	0x4112206D	PB	Sun Part # 528-1122-03
							ROHS	Sun Part # 528-1140-03

1. Type:

PB - ROHS compliant (server); lead in solder balls & bumps. Not compatible with lead-free assembly

ROHS - ROHS compliant (server & desktop); lead in bumps

PB Free - ROHS compliant (server & desktop); no lead in solder balls or bumps

For future revs of Fire, the only field that will change will be the MR field. For any second sources of Fire, the MID field will change and a corresponding rev for that vendor.



On a running sun4u system, the `'prtdiag -v'` command can be used to show the MR value for any Fire parts. The reported revision corresponds to the MR field value.

Code Example 1 Using prtdiag to view revision information

```
# prtdiag -v
...
ASIC Revisions:
-----
Path                Device                Status                Revision
-----
/pci@1e,600000      pciex108e,80f0      okay                  3
/pci@1f,700000      pciex108e,80f0      okay                  3
```

Note – There may be slight variations in the output on different systems, such as printing “pcie” instead of “Fire” for `'prtdiag'` output.

Entering Bugs

A summary is given for each known bug or anomaly and the bug ID within Sun’s internal GlobeTrack, and if applicable, BugTraq database is given. If a work around is known, it is indicated as well. When new bugs are discovered, they must be recorded in the Fire Family Errata Table for all affected versions and a detail summary must be recorded.

Errata Numbering

Errata x.x-n = x.x is chip revision, n is revision based sequential number



Errata Table

Table 3 Fire Family Errata Table

Errata	Globe Track	Bugster	FIRE		Description
			2.0	2.1	
			✓	✓	PCI Express loopback as a master is not supported
2.0-1	P2044		✓	✓	LPU incrementing its REPLAY_NUM incorrectly due to NAK before Recovery
2.0-2	P2138		✓	✓	TLP/DLLP Receiver error not set in LPU Link Layer Interrupt and Status Register for any receiver errors detected in first symbol time
2.0-3	P2138		✓	✓	LPU doesn't NAK TLP with any receiver errors detected starting from STP to last byte of TLP header
2.0-4	P2168		✓	✓	TLP is dropped as "Bad TLP" when first symbol time of the TLP is packed with the last symbol time of DLLP with EDB
2.0-5	P2238 P2244	6217422	✓	✓	LPU does not report error for any illegal PAD symbols
2.0-6	P2232		✓	✓	Unexpected ingress CplID causes ingress data buffer (IDB) corruption
2.0-7	P2247		✓	✓	Simultaneous PB and SW soft reset improperly logged as fatal reset
2.0-8	P2255		✓		When in Drain State caused by EHB/EDB parity error, Fire may either mis-map PIO read return, or send duplicate PIO read return
2.0-9	P2258		✓		Fire doesn't reply with completions and update FC credits to ingress poisoned Cfg or IO write requests, violating PCI-E 2.7.2.2
2.0-10	P2265		✓		Fire loses PIO completion hit by ingress header buffer (IHB) parity error
2.0-11	P2287		✓		IHB parity error can cause egress data corruption
2.0-12	P2288	6283529	✓	✓	Fire's L0s Power Management Locks up Transmits Side of PCIE Link
2.0-13	P2290		✓		Disabling TLU FC receiver overflow check yields unexpected data corruption
2.0-14	P2291	6298418	✓	✓	Fire's outgoing Message Requests do not fill in proper RequesterID value
2.0-16	P2270	6278984	✓	✓	MemoryBist diagnostic features do not operate at speed
2.0-17	P2295	6297420	✓	✓	Top level logicBist cannot be run in system
2.0-18	P2313	6362138	✓		Fire has potential for deadlock on long responses to PIO Read
2.0-19	P2314	6364442	✓		Broken PCIe device can cause data corruption from Fire on receipt of an 'overflow' DMAW packet
2.0-20	P2350	6463216	✓	✓	Fire 2.0/2.1 will not interoperate with PCIe2.0 devices that are directly connected to it.

Table Key:



-
- ✓ = Errata applies to this revision
 - n/a = Errata was not present in these versions
 - Fixed = Bug fixed or fix carried forward from previous version
 - Feature = Irregular feature that will not be changed in silicon

Errata 2.0-1:

LPU incrementing its REPLAY_NUM incorrectly due to NAK before Recovery

Bug ID(s):

P2044 (Globetrack)

Symptom:

Due to the incorrect incrementing of REPLAY_NUM, Fire will initiate entering Recovery earlier than expected.

Description:

LPU increments its REPLAY_NUM when a NAK is received, instead of incrementing its REPLAY_NUM until the related replay is executed as specified by PCI-E. This problem is self recoverable.

Workaround:

None.



Errata 2.0-2:

TLP/DLLP Receiver error not set in LPU Link Layer Interrupt and Status Register for any receiver errors detected in first symbol time

Bug ID(s):

P2138 (Globetrack)

Symptom:

TLP/DLLP Receiver error (bit[18]/bit[21]) is not set in LPU Link Layer Interrupt and Status Register when a receiver error is detected in first symbol time.

Description:

If a TLP/DLLP receiver error is detected, the following bits should be set:

- 1) Receiver error (bit[0] or bit[32]) in TLU Correctable Error Status Clear Register
- 2) TLP/DLLP Receiver error (bit[18]/bit[21]) in LPU Link Layer Interrupt and Status Register, and thus bit[31]
- 3) Link Interrupt Event (bit[11] or bit[43]) in TLU Other Event Status Clear Register, resulting from #2

However, when the receiver error happens on the first symbol time, the #2 and #3 are not set. Fortunately, #1 is set. Therefore, we recommend to rely on #1 register for receiver errors only.

Workaround:

Mask out TLP/DLLP Receiver error (bit[18]/bit[21]) in LPU Link Layer Interrupt and Status Register.



Errata 2.0-3:

LPU doesn't NAK TLP with any receiver errors detected starting from STP to last byte of TLP header

Bug ID(s):

P2138 (Globetrack)

Symptom:

No NAK sent as expected from Fire after a TLP receiver error, which is detected starting from STP to last byte of TLP header.

Description:

A NAK should be sent out when a TLP receiver error is detected. In this case, no NAK is sent out from Fire right after receiving the TLP with receiver error. However, a NAK will be sent out when Fire receives next TLP because the second TLP's sequence number must be out of order. The worse case, the remote device doesn't send more TLPs after the TLP with receiver error. Eventually, the replay timer will time out in the remote device and the remote device will replay again. Therefore, this bug is self recoverable.

Workaround:

None.



Errata 2.0-4:

A good TLP is dropped as "Bad TLP" when first symbol time of the TLP is packed with the last symbol time of DLLP with EDB

Bug ID(s):

P2168 (Globetrack)

Symptom:

An unexpected NAK is sent by Fire when Fire drops a "good TLP" as "Bad TLP" when first symbol time of the TLP is packed with the last symbol time of DLLP with EDB.

Description:

The described unexpected NAK will cause the remote device to replay. This is self correcting.

Workaround:

None.



Errata 2.0-5:

LPU does not report error for any illegal PAD symbols

Bug ID(s):

P2238, P2244 (Globetrack), 6217422 (Bugster)

Symptom:

Illegal PAD symbols are not checked by LPU, i.e, not reported as receiver error.

Description:

It was discovered that a certain graphics card sources illegal PAD symbols instead of IDLE symbols for all lanes, whenever it was connected to a link with a width less than x16. Fire was reporting this violation as a receive error (could not be disabled).

RFE P2238 removed this check but a side effect was the logging of all illegal PAD symbols has been eliminated from Fire 2.0. However, upon receipt of any illegal PAD symbol that occurs within a packet (not outside as the non-compliant graphics card does) Fire 2.0 will NAK the packet.

Workaround:

None.



Errata 2.0-6:

Unexpected ingress CplD causes ingress data buffer (IDB) corruption

Bug ID(s):

P2232 (Globetrack) with the HW ECO fix.

Symptom:

When IDB is full, an unexpected ingress CplD will over-write IDB and cause IDB corruption.

Description:

There were two potential problems in the original bug report, which are IDB overflow (corruption) due to 1) over-sized (more than 16 DWs) CplD or 2) unexpected CplD. With the HW ECO fix, the design filters out any ingress CplD with payload more than 16 DWs. Therefore, the first original problem is gone and only the unexpected CplD will cause the problem when IDB is full. However, the potential problem can be avoided if the ingress posted data initial credit is reduced by 4 from the default value. The reduced credit of 4 serves the purpose as a place holder for the unexpected CplD.

FYI, performance measurement has been made with the Niagara/Fire system and proved that there is no performance impact with the reduced by 4 from the default value on the ingress posted data initial credit.

Workaround:

The value of Posted Data Credit in TLU Ingress Credits Initial Register must be programmed by SW to avoid this potential ingress data buffer (IDB) overflow. The default value is 0xC0, but the allowed max. value is 0xBC.



Errata 2.0-7:

Simultaneous PB and SW soft reset improperly logged as fatal reset.

Bug ID(s):

P2247 (Globetrack)

Symptom:

A FATAL Reset is logged, when Software initiated a Soft Reset.

Description:

Jbus Resets can be initiated by a number of means within Fire. This bug occurs when Software has initiated a soft reset by writing the Reset Generation Register, and there is a soft reset generated via the assertion of PB_RESET_L signal.

The bug only occurs if the PB_RESET_L signal is deasserted within a 1 clock cycle window around when the end of the SW initiated reset would have happened. In this case, the reset length is extended (proper behavior), but instead of the reset getting logged in the Reset Source Register as a PB_RST, it gets incorrectly logged as a FATAL reset.

Workaround:

A potential work around is for software to track when it requests a soft reset, and to use tracking to determine if a FATAL logged reset is potentially a false one. If PB_RESET_L assertions do not occur, this bug doesn't happen.



Errata 2.0-8:

When in Drain State caused by EHB/EDB parity error, Fire may either mis-map PIO read return, or send duplicate PIO read return.

Bug ID(s):

P2255 (Globetrack)

Symptom:

Bad PIOR return issued from Fire as either duplicate PIOR completion or data corruption to processor state during Drain State when cause is Egress Header or Data Buffer Parity Error, both which are fatal HW conditions.

Description:

It is a 3-cycle open window that an ingress CplD packet can be written into Ingress Header/Data Buffer (IHB/IDB) after Fire gets into Drain State caused by EHB/EDB parity error. The CplD written into IHB/IDB after Drain State is conflict with Drain State, which may result as either duplicate PIO read return or mis-mapped PIO read return (potential data corruption) if the ingress pipeline stalls.

FYI, the source of this is an internal regfile parity error the FIT rate is known to be 70, which equates a MTBF of every 1631 years, or one failure every 1631 machines. However, this is the regfile failure rate. The actual bug failure rate is much smaller as the '3-cycle' window must be also be hit along with the PIO reads reissue and pipeline stall.

Workaround:

None.



Errata 2.0-9:

Fire doesn't reply with completions and update FC credits to ingress poisoned Cfg or IO write requests, violating PCI-E 2.7.2.2.

Bug ID(s):

P2258 (Globetrack)

Symptom:

Time out errors will be reported from remote device and FC credits are lost when the remote device sends poisoned Cfg or IO write requests to Fire.

Description:

When a poisoned Configuration or IO Space write request (CfgWr or IOWr) is received, Fire drops it without collecting FC credits and without supplying a completion.

Based on PCI-E protocols, a completion needs to be sent for any non-posted request; FC credits need be collected and updated to the remote device. Fire violates these two protocols when a poisoned CfgWr or IOWr request is received. However, Fire does detect the two associated errors in the request and sets the error bits in TLU UE Error Status Clear Register (Poisoned Packet (PP bit) and Unsupported Request (UR bit)).

Note that Fire supports neither of these transaction types (CfgWr or IOWr) on ingress side (receiving from PCI-E port).

Workaround:

None.



Errata 2.0-10:

Fire loses PIO completion hit by ingress header buffer (IHB) parity error

Bug ID(s):

P2265 (Globetrack)

Symptom:

PIO read timeout error may be reported by the source processor due to the lost PIO completion hit by IHB parity error in Fire.

Description:

When an IHB parity error is detected by Fire on a PIO read completion header, it should be only one notification of the error, which is an interrupt from Fire for the IHB parity error if the interrupt is enabled. However, due to the bug, there may be two notifications of this error:

- 1) Interrupt from Fire for IHB Parity Error
- 2) PIO Read timeout error at the source processor.

The #2 above is caused by the lost PIO read completion hit by the IHB parity error in Fire.

The IHB parity error causes Fire to enter the Drain State and the logic bug is that Fire wipes the PIO read completion hit by IHB parity error thus causing an eventual timeout at the source processor.

At this point it could be a race between the IHB parity error interrupt and the PIO read timeout, though most likely we'd see the interrupt first.

Also note, it is believed that the IHB parity error in itself is fatal since it causes hardware to enter the Drain State.

Workaround:

None.

Errata 2.0-11:

IHB parity error can cause egress data corruption

Bug ID(s):

P2287 (Globetrack)

Symptom:

Corrupted data transmitted from Fire to PCIe port during Drain State when cause is Ingress Header Buffer (IHB) parity error, which is fatal HW condition

Description:

In order to stream the pipeline of writing headers/data to EHB/EDB, a header is pushed to the pipeline to EHB one cycle sooner than the last write to EDB. However, at entering Drain state, both header and data pipelines are stopped at same time. Therefore, if Drain State happens at the single cycle, on which the header is pushed to the pipeline to EHB, but not the last data write to EDB, data corruption occurs.

Since the Drain State caused by IHB parity error doesn't stall the egress PCIe transmit port (all the other causes do stall it), the corrupted data in EDB is transmitted to PCIe port if the Drain State is caused by IHB parity error.

Also note, we believe the IHB parity error in itself is fatal since it causes hardware to enter the Drain State.

Workaround:

None.



Errata 2.0-12:

Fire's L0s Power Management Locks up Transmits Side of PCIE Link

Bug ID(s):

P2288 (Globetrack), 6283529 (Bugster)

Symptom:

System hangs since Fire can NO longer transmit anything

Description:

When enabling both Fire and the down stream PCIE device to use L0s power management, Fire stops sending data out of its transmit side. Fire gets stuck in L0s_Tx_f state and never comes back out.

When enabling Fire by itself to do L0s no problems are seen. Fire goes in and out of the TX power management states fine.

When enabling the downstream device by itself to do L0s no problems are seen. Fire goes in and out of the RX power management states fine.

Workaround:

Do not enable both Fire and downstream device's L0s power management at same time.

Errata 2.0-13:

Disabling TLU FC receiver overflow check yields unexpected data corruption

Bug ID(s):

P2290 (Globetrack)

Symptom:

Corrupted data is written to Memory when TLU FC receiver overflow check is disabled and PCIe Receiver FC overflow is detected.

Description:

When TLU FC receiver overflow check is disabled and PCIe Receiver FC overflow is detected, TLU correctly writes the header to IHB, but stops writing data to IDB. Therefore, wrong data is pulled out of IDB in processing the header from IHB - data corruption happens.

Workaround:

Do not disable FC Receiver overflow check (set bit 38) in TLU Diagnostic Register. By default, the check is on.



Errata 2.0-14:

Fire's outgoing Message Requests do not fill in proper RequesterID value

Bug ID(s):

P2291 (Globetrack), 6298418 (Bugster)

Symptom:

RequesterID field of Set Slot Power Limit Message is always 0x0 even if the RequesterID field in Fire programmed to a non-zero value.

Description:

Fire's RequesterID field in the DMC PCI Express Configuration Register (offset 0x653100/0x753100, REQ_ID field, bits 15:0) does not propagate to all message types it needs to. Set Slot Power Limit Message all the outgoing power management Message Requests are effected.

Workaround:

Leave the REQ_ID field at its reset value of 0x0.

This is what OBP does in platforms today (Aug., 2005).

Errata 2.0-15:

LogicBist does not operate at speed in Fire 2.0

Bug ID(s):

P2269, P2294 (Globetrack), 6267974 (Bugster)

Symptom:

PEC logicBist can run at only 200MHz and top level logicBist can run at 100MHz. DMC logicBist, in Fire 2.0, does run at it's native frequency of 200MHz.

Description:

Fire's top level logicBist can be run at half frequency (100MHz). The frequency division can be accomplished by setting the clockRatio bits of the JTAG instruction, so a 200MHz clock can still be applied if necessary. On the asic tester, at speed coverage is attained for the top level using Transition Delay Fault vectors (f0 pattern).

Fire's PEC logicBist can be run at only 200MHz. By setting the instruction to use pll bypass mode the jbus clock is fed in to the logicBist clock circuitry, so a 200MHz clock is applied. On the asic tester, complete stuck at coverage is attained using the 200MHz logicBist vector, and at speed coverage is attained using functional tests.

Workaround:

Both PEC and top level logicBist can be run at slow frequencies to attain stuck at fault coverage.



Errata 2.0-16:

MemoryBist diagnostic modes do not operate at speed in Fire 2.0

Bug ID(s):

P2270 (Globetrack), 6278984 (Bugster)

Symptom:

Fire's top level memoryBist controllers cannot support diagnostic modes at 200MHz.

Description:

Fire's top level memoryBist default test gets complete coverage at 200MHz, but the diagnostic features, such as stop-on-error, cannot be run at full speed. For diagnostic efforts, the clock should be set to half speed mode with the jtag instruction.

Workaround:

Diagnostics can still be run, but they must be run using half speed mode. Default mode testing can be completed at speed.

Errata 2.0-17:

Top level logicBist does not work in system in Fire 2.0

Bug ID(s):

P2295 (Globetrack), 6297420 (Bugster)

Symptom:

Fire's top level logicBist signature mismatches when run in system.

Description:

Fire's top level logicBist default test does not pass in system, the signature mismatches. This same test passes on the tester, the failure cannot be recreated in simulation or on the tester.

Workaround:

Complete coverage is attained on the tester before parts are shipped, so the system can be confident that they are receiving high quality parts. The PEC and DMC logicBist tests can be run in system, these blocks make up approximately 85% of Fire's gates, so a significant portion of Fire's gates can be re-tested in system if desired.



Errata 2.0-18:

Fire has potential for deadlock on long responses to PIO Read

Bug ID(s):

P2313 (Globetrack), 6362138 (Bugster)

Symptom:

System panic occurs due to timeout counter expiring in the IO fabric.

Description:

A system deadlock can occur due to an architectural limitation in Fire where stalled PIO traffic can cause DMA read returns to become blocked. The stalled PIO traffic ultimately times-out and causes a panic.

This was discovered during the debug of an Erie system panic that occurred with a certain third party ethernet chip undergoing stress testing.

Workaround:

None; Though the presence of this bug most likely means a driver is accessing a non-existent register in an IO device, or some other abnormal behavior causing a slow or late response to a PIO Read transaction. Identifying the long transaction and correcting it will eliminate the issue.

Errata 2.0-19:

Broken PCIe device can cause data corruption from Fire on receipt of an 'overflow' DMAW packet

Bug ID(s):

P2314 (Globetrack), 6364442 (Bugster)

Symptom:

Memory data is corrupted by a bad update from Fire. No error in Fire will be signaled.

Description:

The bug occurs when the remote device sends an illegal "receiver overflow" DMA Memory-Write request. The "overflow" relates to the Fire IDB (input data buffer), which is full and cannot accept any new request with data (thus the device has no credits and should not be transmitting the DMA Mem-Write). If the IDB is drained (contents sent to memory) at the same time as the illegal packet is being processed, and if the "overflow" condition no longer exists at the end of the packet, then Fire's TLU will NOT indicate the error and the IDB's contents are compromised. Memory can then be compromised when the invalid packets are sent from the IDB to memory.

This behavior was discovered in simulation and requires a device to violate the PCIe protocol at the same time a specific corner case occurs inside the Fire bridge ASIC and is thus deemed unlikely to occur in actual hardware.

Workaround:

Replace the broken or non-PCIe compliant target device.



Errata 2.0-20:

Bug in PCIe link core prevents Fire 2.0/2.1 from interoperating with directly connected PCIe 2.0 devices

Bug ID(s):

P2350 (Globetrack), 6463216 (Bugster)

Symptom:

PCIe 2.0 devices (when available) will not inter-operate with Fire 2.0/2.1 silicon if they advertise 5GHz speed.

Description:

The link core used in Fire-2.0 and Fire-2.1 contains an error whereby a TS (training sequence) MUST have a DRI (data rate identifier) equal to 'h02 to be considered valid. Any PCIe-2.0 device which supports 5GHz link speed will advertise a DRI of 'h06, which is interpreted as an invalid TS by Fire, and so it cannot complete the link training process.

This means that PCIe 2.0 devices that direct connect to Fire will not link train if they advertise 5GHz operation. PCIe 2.0 devices are expected on the market in mid to late 2007 and most are expected to advertise 5GHz ability.

Current platforms that provide a direct connection to Fire are: Chicago, Erie and Seattle.

In all other platforms, PCIe cards can only connect to Fire through the on-board PCIe switch and hence do not have this problem.

Workaround:

At a system level, plug any PCIe 2.0 card into one of the PCIe slots that connect to Fire via the on-board PCIe switch (usually several such slots exist per platform).



This page is intentionally blank.

