COTS Integration and Debugging Challenges - RBSP Lessons Learned

Subodh Harmalkar
Joseph Hennawy
Samuel Fix
Debbie Clancy

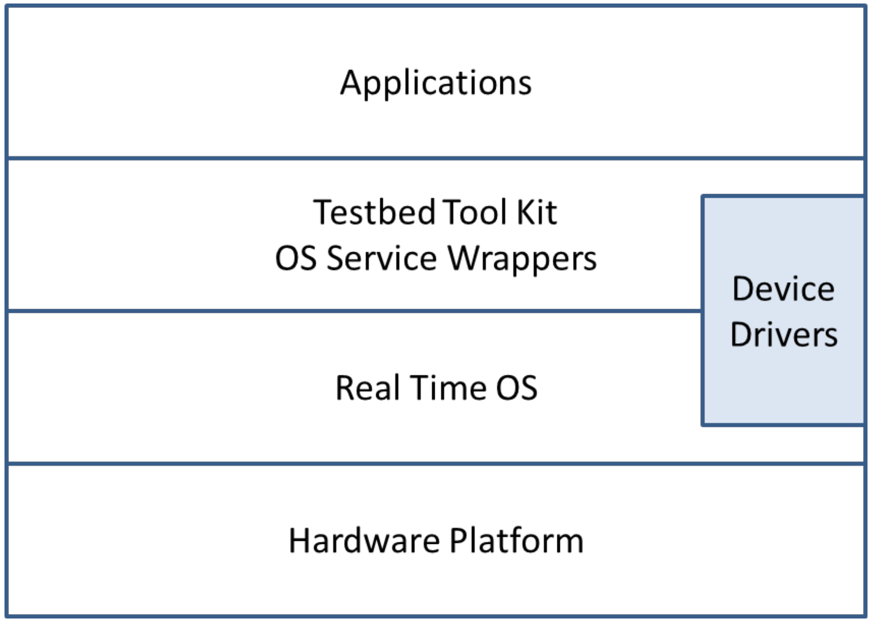
JOHNS HOPKINS UNIVERSITY
Applied Physics Laboratory

Agenda

- Background
- Testbed Architecture
- Intel PCI bus architecture
- Testbed modifications
- OS support and Internal investigations
- SBC Vendor and its BIOS support
- Lessons learned

Testbed Architecture

- cPCI Chassis
  - Single Board Computer (SBC)
    - Intel Pentium M based cPCI card
  - Other APL developed HW



Problem Description

- Intermittent hang, when testbed idle.
- Frequency variable: months to hours.
- Reset would always work !
- Features in Testbed SW releases were priority, so this issue was a low priority.
- Mission Simulation #3 (MSIM3) runs for 10 days. So, as MSIM3 came close, longevity became important.
- Issue became RBSP Project level risk.

Areas we looked into..

- Testbed Applications
- Testbed Device Drivers
- OS
- Underlying Hardware

- We analyzed, stressed *all* layers of testbed architecture, OS and board vendor’s hardware and support.

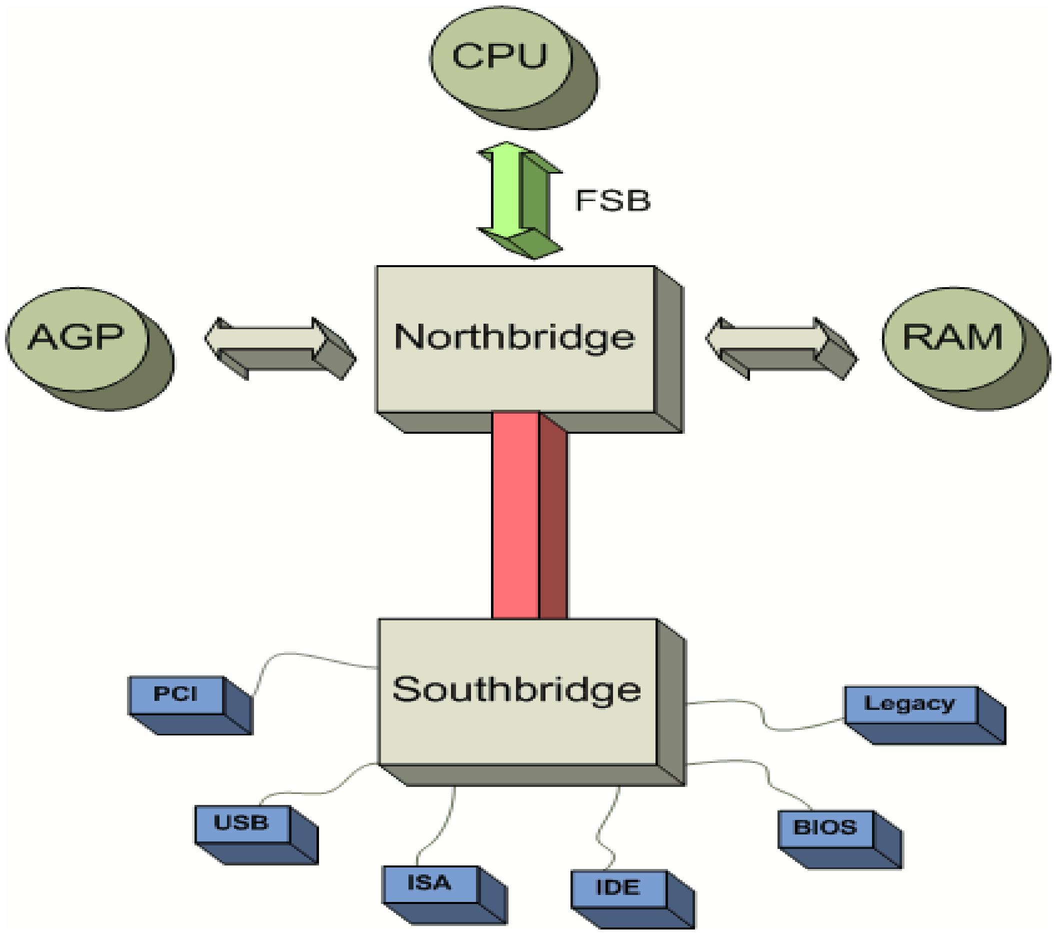
Testbed modifications

- Investigated PCI, device drivers, apps, testbed configurations
- Increased interrupt rates
- Stress tested testbed software interfaces
- Converted a 3U module to show pending interrupts on PCI bus when CPU hung.

Testbed SW stress test details

- Built Testbed stress test suite that stresses:
  - Testbed Commanding Interface (Ethernet)
  - Testbed Telemetry Interface (Ethernet)
  - UUT (Unit Under Test) Commanding Interface (PCI/Serial)
  - UUT Telemetry Interface (PCI/Serial)
- One chassis was running TBSW undisturbed (without any peripheral hardware)
  - Experimented with timer tick on this board
  - Never hung in 2-3 months

Intel PCI bus Architecture

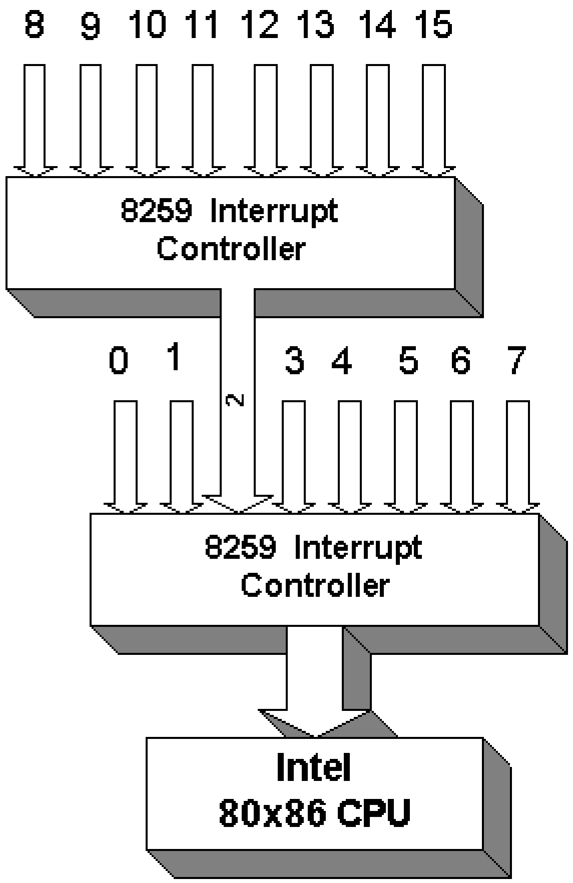


http://en.wikipedia.org/wiki/Northbridge_(computing)

Testbed modifications - SW

- Developed high priority ‘beeper’ and ‘blinker’ applications to detect lack of OS scheduling.
- Being Embedded engineers, disabled GUI. Wrote a simple script to print date on console.
- Developed an automatic email application (on another machine) that detected testbed hang and sent an email and text messages.
    - Useful for gathering data quickly
    - Network issues caused a lot of false alarms.
- Started discussions with APL’s internal Intel CPU experts and with RTOS customer support

80x86 Interrupts using PIC (Programmable Interrupt Controller)



Intel CPU (Pentium) gets interrupted through cascaded interrupt controllers, acknowledges that interrupt, completes the work and sends End Of Interrupt to Interrupt controller.

APL experts decided to investigate with JTAG. It showed that when testbed hung, the IF (Interrupt Enable Flag) was enabled.

CPU was not receiving any more interrupts. When we forced EOI to PIC, system woke up but hung again in few minutes.

Most likely, CPU had missed sending EOI.
This could be an ***OS*** issue…

OS – Tier 1 and Tier 3 Support

- Reviewed OS release notes, user forums etc. Did not see similar problems since OS release.
- Upgraded OS maintenance contract to Tier 3 (Gold/Bronze) to gain better and quicker support with engineering contacts.
- Tier 3 reviewed our suspected code snippets quickly. They found no issue in our system clock, interrupt handlers, device driver specific code.
- **Tier 3 requested to reduce system clock frequency. To every one's surprise, testbeds started hanging quickly !! No one could explain why.**
- Kernel level debugging tools could not capture the last "snapshot" of system state, since CPU was not responding. The only possibility was to analyze data in memory AFTER the hang. We needed access to OS internal data structures.
- We were not successful in getting access to OS source code due to legal issues, in timely manner.

OS Tier 3 response..

- Tier 3 folks discussed with kernel developers. Some engineer remembered “Ah.. We had a problem with Intel’s HLT instruction. Tell them to stop using HLT.”
- The referred issue would happen at startup. Our problem was happening during run time, but we tried that approach.
- HLT is used in IDLE task of OS
- “-h” at startup replaces HLT by while(1) loop.
- **Testbeds stopped hanging..**

- But we kept looking for root cause.

Clues emerged from data..

- By now, Integration and Test (I&T) testbeds were used constantly (with HLT instruction) and were NOT hanging.
- So, we investigated the differences in HW.
- I&T testbeds had newer boards than Flight SW testbeds.
- SBC vendor had not informed us about new revision updates. We had not investigated the differences when we received boards.
- Intel's Pentium M errata sheet had an Erratum X21 related to "HLT" instruction.

Plausible root cause

- **X21 Problem**: If an external snoop causes an eviction from the Instruction Fetch Unit (IFU) instruction cache, the processor may, on exit from HLT state, erroneously read stale data from the victim cache.
- **Implication**: This may lead to unpredictable behavior. Intel has only observed this condition in non-mobile configuration.
- **Workaround**: It is possible for the BIOS to contain a workaround.
- Intel speak for “Update your BIOS !!”

- [http://download.intel.com/support/processors/mobile/pm/sb/30220925.pdf]

Plausible root cause

- Plausible scenario:
  - CPU stops at HLT
  - CPU is interrupted by an interrupt that is asynchronous to timer interrupt i.e. Network / EIDE related DMA completion.
  - The snoop cycle causes IFU eviction when Instruction cache has been victimized.
  - IFU fetches invalid instruction or incorrect program counter.
  - On that or the next interrupt cycle, CPU does not issue EOI due to some interaction between OS and CPU states.

- Few options
  - (a) Avoid “HLT”
  - (b) Avoid DMA (not good for system timing)
  - (c) **Update BIOS**.

Updating BIOS

- SBC Vendor – "You have up-to-date BIOS from our BIOS vendor."
- BIOS Vendor – "Our BIOS is up-to-date."
- Errata name changed from X21 to Y36 from 2005 to 2009.
- Google "Intel Pentium errata Y36"
- Pushed "Toshiba fixed this in 2005. Your BIOS release date is two years earlier."
- Two days later, BIOS vendor gave an update to SBC Vendor.
- A configuration that hung every 6 hrs, ran well for 10+ days with new BIOS.

- Root cause:
  - Not following CPU Erratum and lack of CM (Configuration Mgmt) and Integration testing in COTS products.

Lessons Learned

- Stress test Flight (HW+SW) ***and*** Testbed (HW+SW) !!
- Understand the platform we build upon !!
- Keep detailed records “what (configuration), when, symptoms”. Increase priority of low frequency “hang” issues to program management as soon as possible.
- Build “continuous self-test” to exercise possible features during testbed idle time.
- Keep JTAG, Bus analyzer like hardware debug tools and skills up-to-date.
- Get access (not necessarily purchase) to source code for OS, BIOS and drivers.
- Investigate levels of integration of COTS products, support by vendors and track their release notes.

Lessons Learned

- An integrated solution with turn key guarantees, and adequate support is preferred for critical and potentially multi-user systems.
- Adequate risk level grading process needs to be devised and applied to each COTS SW / HW / FW component introduced into a system.
- The process should include the mechanism of managing COTS risk assignment throughout different mission phases.

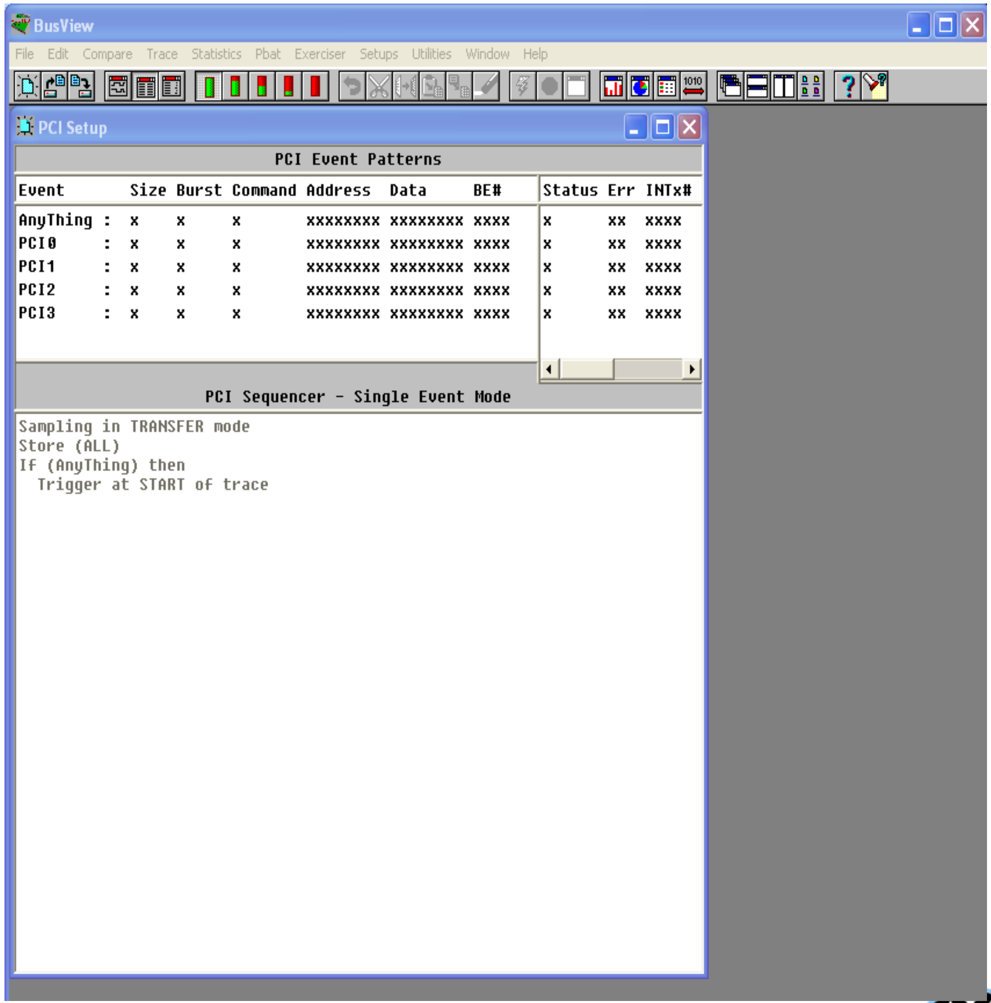
Q&A

Backup slides

PCI Investigation

- V-Metro BusView
- PCI Bus Analyzer
  - Trigger Events
  - State machines
  - Buffers
  - PCI Interrupts
- Nothing unusual seen in multiple PCI traces.

- Still CPU state unknown..


BusView
File Edit Compare Trace Statistics Pbat Exerciser Setups Utilities Window Help
PCI Setup

PCI Event Patterns

Event		Size	Burst	Command	Address	Data	BE#	Status	Err	INTx#
AnyThing	:	x	x	x	xxxxxxxx	xxxxxxxx	xxxx	x	xx	xxxx
PCI0	:	x	x	x	xxxxxxxx	xxxxxxxx	xxxx	x	xx	xxxx
PCI1	:	x	x	x	xxxxxxxx	xxxxxxxx	xxxx	x	xx	xxxx
PCI2	:	x	x	x	xxxxxxxx	xxxxxxxx	xxxx	x	xx	xxxx
PCI3	:	x	x	x	xxxxxxxx	xxxxxxxx	xxxx	x	xx	xxxx

PCI Sequencer - Single Event Mode

```
Sampling in TRANSFER mode
Store (ALL)
If (AnyThing) then
  Trigger at START of trace
```

PCI Interrupt module

- Converted a 3U module to show pending PCI interrupts from backplane
- After hang, some interrupt lines were seen pending low i.e. not acknowledged
- It was useful to know which interrupts were pending, but did not lead us to the cause.

JTAG

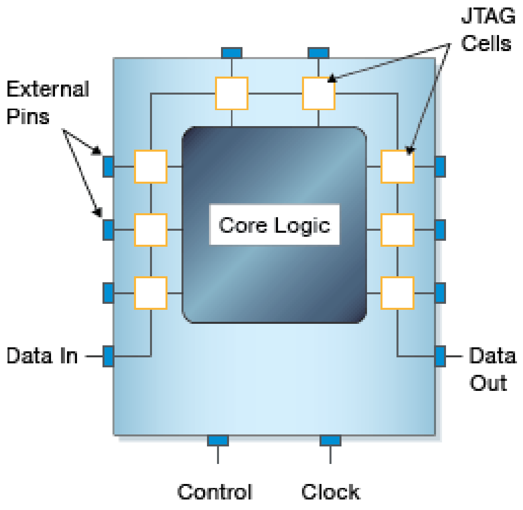


Figure 1 - Simple JTAG device

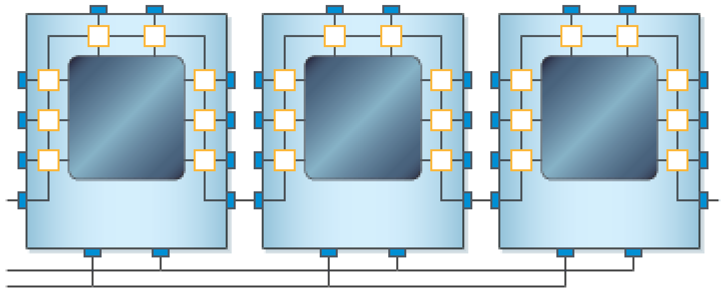


Figure 2 - Simple JTAG chain

http://www.xjtag.com/

JTAG: Joint Test Action Group (now IEEE standard)