Conclusions of the Fault Tolerant Spaceborne Computing Employing New Technologies

May 28-30, 2008

Sandia National Laboratories

Erik DeBenedictis (Sandia)
Larry Bergman (JPL)
Richard Stempien (MITRE)

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

JPL MITRE Sandia National Laboratories

Conclusions

- PCI-Express and RapidIO seen as both acceptable (GigE too)
- Developing “boards” compatible with one/both interface standards would [define a de-facto architecture/not be a wasted effort] and a target for the software community
  - Software community would use the hardware above as a target for yet to be developed middleware
- [Software and hardware WG outtakes here]
- The workshop brought together a number of people in software fault tolerance and lead to further collaborations among them.
- Meet again in a year
- Create password protected electronic forum

JPL MITRE Sandia National Laboratories

DRAFT

Hardware Working Group

Sandia Computer Science Research Institute
Notes and Conclusions May 27, 2008

Inventory

Radiation hardened hardware may be classified into the following:

1) Image processing
2) High speed positioning -floating point
3) Data routing (I.e. TSAT) -effectively a "router in the sky"
4) Command & data handling -many lines of code, such as for an emergency escape plan from a rocket. (I.e. this type is driving GN&C, situational awareness)

It has been commented that availability does not necessarily mean affordability.
Some products mentioned:

Available now:

-1553b is still being used, but not desired
-16kb Carbon Nanotube memory array
-1993 EEPROM
-Virtex 5
-Space Micro -technology has been developed for a 4Gb flash cube.. (100krad wo shield & 120 MeV SEU, SEFI and SEL)
- Seakr: Virtex4
-Seakr: Gbit Phy
-BAE Rad750
-BAE L2 Cache

Available soon:

-40Mb Carbon Nanotube array (dec 08)
-Seakr Iris modem-router. Ethernet level. Also __to Ethernet.
-CRAM
-BAE/ACHRONIX RH FPGA (CNT fabric added in later phase)

Not available:

-FPGA with ISP
-Serdes without IP difficulties

Scenarios for architectures

It is generally agreed that there are few ways for the space hardware community to develop standards (architecture standards and interconnect standards for example). Changing current practices is expensive, and is simply not in the interest of individual companies. Standards can only be considered when they will save money for all involved.

Roadmap

1) Nonvolatile memory: NRAM, CRAM
2) There is a demand for large memories, up to 1Gb.

3) SDram. Present processors require 400Mhz DDR2, and this speed is insufficient. JPL has been testing and found 2k consecutive errors.

Gaps or action items

Although funding comes on a year-by-year basis, the space industry, by its nature, requires long-term projections. There is a problem in the industry because the funding is coming from five year projection, but program managers are locking themselves into 10-15 year projections. After five years it may be that the program is not very strategic, but program managers keep pushing because there is no funding to switch tracks.

Discussion of multi-core development

It is expected that the demands of the space community will continue to follow the commercial community; both have an interest in producing smaller, more power-efficient, faster hardware. For example, small laptops and cell phones are creating a demand for power and heat solutions. The gaming market is creating a demand for faster processing.

However, the space community has several limitations:

-RH requirements will restrict density of transistors. At some point space hardware will probably have to branch away from FETs and switch to carbon nanotubes

-Power/temperature restrictions will keep the space industry from making quick leaps to multi-core processing. When multiple cores do come into view, the space community will also need to develop our own version of multi core hardware (for example, there would be no demand for a processor with 1000 cores each the size of an arithmetic unit)

Note that there are three categories of multi-core processors, which must be considered separately. At this point no single type is emerging above the others as the best commercial solution.

Hierarchical -cores are indistinguishable

Array -cores different and therefore cannot be swapped

Pipelined -data feeds from one core to the next (good for performance but redundancy is more difficult)

Discussion of Interconnects

At this time RapidIO is generally the best option, even though it rests on a shorter software legacy than PCI. PCI express is much more attractive (PCIx will be skipped over), but it is presently not available to the space community because it was developed with no concern for long term reliability. A graceful degradation has not been considered. Radhard timetriggered Ethernet is being developed (10Gbps range) and will perhaps make PCI obsolete, although some kind of RapidIO will still be necessary.

Software Working Group Report Out Summary

Preliminary
(to be further expanded after workshop)

May 29, 2008

Larry Bergman
Hans Zima

Software Working Group
Members

- Marti Bancroft (US Gov)
- Larry Bergman (JPL) *
- Greg Cieslewski (UFL)
- Steve Crago (ISI)
- Joe Coughlan (ARC)
- Rich Doyle (JPL)
- Steve Duran (JSC)
- Daniel Gallegos (Sandia)
- Adam Jacobs (UFL)
- Zbigniew Kalbarczyk (UIUC)
- Michael McDougall (GrammaTech)
- Scott Michel (Aerospace Corp)
- Darrel Raines (JSC)
- Jim Sherrill (ClearSpeed)
- Joe Shultz (GE Global Research)
- Peter Szilaqyi (Reservoir Labs)
- Jack Stocky (JPL)
- Henry Stone (JPL)
- David Wangerin (Aerospace Corp)
- Hans Zima (JPL)

* Facilitator / scribe

Preface

- The Software Working Group discussed software and systems architecture challenges for the multi-core era in three contexts:
  - Challenging spacecraft systems **requirements** for which no easy solution is known
  - **Opportunities** for introducing paradigm shifts in spacecraft avionics or systems capability
  - Challenging **software engineering** problems, exacerbated by multi-core

Software Challenge Domains

- Systems architecture
- Design tools and environments
- Technology Infusion Issues
- Project / Programmatic / Political / Cultural

Software Challenges
Systems Architecture (1)

- Faults must be traceable to a specific hardware and/or software defect to be able to adequately diagnose failures in operations
  - Possible major impact on introducing high level abstractions into architecture
  - Want to have determinism when doing V&V and analyzing faults
- Could multi-core be used to do 3-way (or more) TMR (or other ABFT) on the same chip?
- Need a programming model for highly reliable embedded multi-core systems architectures

Software Challenges
Systems Architecture (2)

- Distributed Systems Challenge: State synchronization of multiple multi-core computers required for human-rated systems (and possibly others as well)
  - Both software and hardware implications
- Need to be able to selectively turn off hardware features that are not needed (saves power and reduces overhead)
- Reducing software complexity, development time (cost) and V&V may require feeding more requirements to hardware community than has been in the past

Software Challenges
Systems Architecture (3)

- Need to support 0, 1, and 2 level fault tolerance (human rated systems). 2 level is not required in all situations.
- Are SEU error models well known enough to devise the right software architecture (may suggest more research in fault detection/propagation/containment)?

Software Challenges

Design Tools and Environments (1)

- Should there be a multi-agency certification program put in place to eliminate “common cause” bugs in compilers, execution environments, V&V tool suites, etc?
- Would an “Open Source like” (limited to US orgs) distribution of common spacecraft software elements accelerate the infusion the adoption of multi-core?
- Need for intelligent set of tools for the support of key features of the programming model and the associated languages – for program development, debugging, and execution

Software Challenges
Design Tools and Environments (2)

- AI technology needed for the support of architecture and application-aware compilation (DARPA's recent AACE BAA is an important step in this direction)
- Automatic support for porting of legacy codes
  - this is also a problem requiring sophisticated AI technology (it has never been solved adequately in the HPC environment
  - how important is this problem? (New mission software often developed from scratch)
- Generalization of V&V technology to runtime verification and application-specific, knowledge-based fault tolerance strategies (possibly based on introspection)

Software Challenges
Technology Infusion Issues

- Need to devise a low risk incremental approach to infusing autonomy into spaceborne systems
  - Real payoff in terms of lowering operational costs, enhancing vehicle health management, improving reaction time in science/intelligence instruments observations, and ultimately, control applications such as EDL
  - Challenge in retaining legacy code/systems while accomplishing new technology infusion
  - Need to demonstrate utility of multicore to drive adoption
- Distributed systems challenge (both hardware and software): A need to express spacecraft **state** and **intent** (to ground operations personnel and/or crew members) to ease infusion of autonomy into manned and unmanned space systems, and provide incremental infusion steps

Software Challenges
Policy, Programmatic, Project, Cultural

- A strong university education program is needed for highly reliable multi-core embedded systems software
- Project Managers must see the need for multi-core before committing to accept the risk of a new technology
  - Are there any unique applications that multi-core can do that single core cannot?
- Real dilemma: software tools and methods lag hardware systems by 5-7 years
  - Need to close the gap, perhaps by looking at new programming paradigms
  - Commercial sector will help but not completely
- Real Issue: Reducing the time to develop, test, review, … new software uploads for spacecraft (presently 6-9 months or more)