Future of System Level Design Panel Discussion

Jack Kouloheris
Sr. Manager, Embedded Systems
IBM T. J. Watson Research Center

Oct. 3 2003
I. No Silver Bullet

- Designing complete systems (which are increasingly systems-on-a-chip) is just plain hard and getting harder.
- We are faced with complexity levels that are rising as fast or faster than Moore’s law.
- Complexity is driven not only by the increase in the raw number of transistors on a chip, but also by the fact that chips now have:
  - Analog elements
  - RF elements
  - Multiple programmable elements (with RT constraints)
- Systems are networked together into larger systems (e.g. autos)
- Not to mention mechanical and thermal design which are traditionally in separate CAD silos.

Before we worry about the future, let’s get today’s SLD tools useable and working
IBM STB 04XXX Integrated STB Controller

- True system on a chip
  - Integrates CPU, peripherals, HW audio/video/graphics, analog functions
  - All STB functions with exception of RF tuner and demodulator
  - Multiple programmable elements
    - PPC 405 CPU & caches
    - DSP for Audio decoding
    - Programmable 2D graphics
    - Pico code for Video Decoder
  - Mixed Signal
    - 2 PLLS (audio and video)
    - Video DACs
STB Controller Block Diagram

- 4 programmable elements with different ISAs
- Analog elements (PLLs, DACS)
- Both internally designed and licensed IP – over 30 cores
- Real time performance constraints
- SW in analog feedback loops
- Accurate power estimates needed early to choose cheapest pkg
STB – HW/SW Coverification tools

- ET3 Simulation of entire SOC
  - Used to verify proper connection of cores
  - Accurate performance/bus BW measurements
  - Can run limited end-to-end simulation
  - Compressed data in -> video out
  - Test cases on PPC processor -> video out
- Run code on previous generation chip
- Standalone ISA simulator (DSP)
Problems with methodology

- ET3 expensive, shared resource. Can’t practically boot/run OS and real applications
- Older platforms don’t have real time performance of new chips – can’t debug performance issues
- Standalone simulators don’t include system level latency issues, interaction between cores

How well does it work?
- 2nd, 3rd generation chips generally 1 pass designs (may have B RIT)
- delays from chip availability to driver kits available – slows time to market
- difficult debugging environment (esp for SW)

Why not use SimOS/CoWare/etc?
No resources to develop models for all cores/ no library
Some bugs may only be found in gate level model (needed to boot Linux on gate level model to find a particular bug)
How do we prove SW model matches RTL model? (no equivalent of boolean comparison), no way to run enough test cases.
II. The great thing about standards – there are so many of them!

- No standard platform in Industry – multiple incompatible frameworks from academia and CAD vendors
- No standard modeling language (C/C++? SystemC? SystemVerilog?)
  - Despite recent efforts, CAD industry continues to fight.
- Extremely poor integration in commercial CAD frameworks
  - Most CAD vendors “SLD” tools continue to be conglomerations of acquired point tools linked by PERL (etc) scripts and format transformations that core dump when you sneeze.
- No libraries of models available for existing cores
  - Need to be able to acquire models in right format for purchased as well as in-house IP
  - Support for legacy IP
- Resource constraints in tight times – should management apply limited engineering staff to write VHDL/Verilog (I.e. do the design) or write the high-level model?
  - Frequently cannot afford both – too few engineers to build models at multiple levels of abstraction
III. The coming crisis in embedded SW

Or

SW is Hard and Hardware is Soft

• Systems (and SOCs) becoming increasingly programmable
  • Support multiple standards
  • Save increasingly expensive mask sets
  • Too many transistors!
  • “Faster time to market”

• BUT
  • Is SW really easier than HW?
Embedded Programmable Devices

3 Embedded devices / person worldwide in 2010

Source: Gartner 2002: Microprocessor, Microcontroller and Digital Signal Processor Forecast Through 2005
Rapid Increase in Electronic Components in Automobiles

Since 1999 significant increase in industry focus on software quality/security to ensure driver safety and overall automobile IT security (e.g. from hackers)

"Drive by Wire" - fully electronic acceleration

Source: Industry Tradeshow Data, 2002
Electronics Will Be an Increasing Fraction of a Vehicle’s Production Costs

"Automotive S/W and Electronics Briefing - IBM Automotive Industry Team, May, 2002"
Increasing Software Complexity

Rapid growth will continue in software complexity for high value industries such as aerospace, consumer electronics, network, industrial and automobiles

Typical product, circa 1995:
100,000 lines of code. Standalone, unmanaged, fixed function

Typical product, circa 2001:
1,000,000 lines of code. Networked, managed, upgradable

A tenfold increase in software content

Source: Wind River, BCG
Hardware – Growth of Complexity and Productivity

Source- Sematech
Implies Manpower CAGR of 30%
Software – Growth of Complexity and Productivity

Source: Applied SW Measurement (Capers Jones), IEEE
HW vs SW development productivity

**Hardware - Chip Complexity and Productivity Growth**

- Logic Trans./Chip (100s):
  - **CAGR 58%**

- Productivity - Trans./ Man Month:
  - **CAGR 21%**

**Software - Lines of Code**

- Prod - LOC/MM:
  - **58% CAGR**

- 3-7% CAGR
Embedded Systems Developers – HW vs. SW


Software developer population growing at 8.3% vs 4.0% for hardware

Source: VDC Embedded Systems Market Analysis 2002/3
Summary – The SW Crisis

- The net effect of these two factors – lower SW productivity improvement and increase in SW content – is driving demand for SW developers at a higher rate than for HW developers. If improvements are not made in SW productivity, it will soon be (if it isn’t already) the prime bottleneck in system level design.

- SW costs, amortized over all units, can be a substantial part of system cost. Example: We are told that the HW cost of the processor in a RIO MP3 player was $9 while the SW development cost, spread over all the millions of units made, was $12 per unit!

- As discussed above, new vehicles may contain millions of lines of code. Software may account for 30% of all problems (dominating HW defects), and leads to bad press. This can be costly both in $$ and public image.

- A key part of addressing the future of SLD is addressing the SW development crisis.
Conclusions

- Need to make today’s tools practically useable by industry
- Need to continue to make progress on standards and availability of high level models
- Need for robust simulation at all levels of abstraction (and mixed levels of abstraction) will continue to increase
- Need to come up with ways to link logical correctness of high level model to RTL design
- Need to address SW productivity and reuse at least as effectively as HW has done else SW will limit speed of innovation.