Multicore Embedded Software Development on Virtual Platforms

Larry Lapides

EDP 2008, 18 April 2008

Agenda

- Embedded software is The Big Problem with SoC products
- Existing embedded software development solutions
- Open Virtual Platforms

The growing challenge

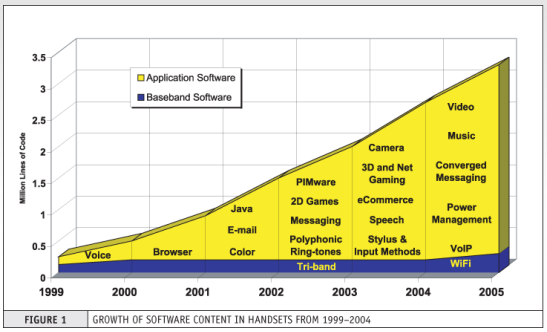
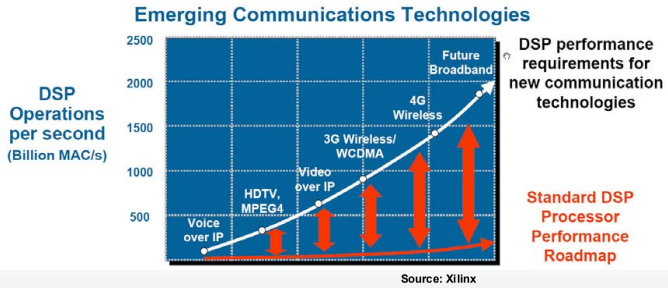


FIGURE 1 GROWTH OF SOFTWARE CONTENT IN HANDSETS FROM 1999–2004

Examining the economics of building next-generation mobile handsets with Linux
By: Bill Weinber, Jun. 14, 2005 11:00 AM, linux.sys-con.com

- SW content of electronic products grows dramatically
  - Millions and millions lines code
  - In 2007 SW dev costs exceeds HW design costs for SoC ICs

- and the software needs to run faster and faster to provide more and more functionality

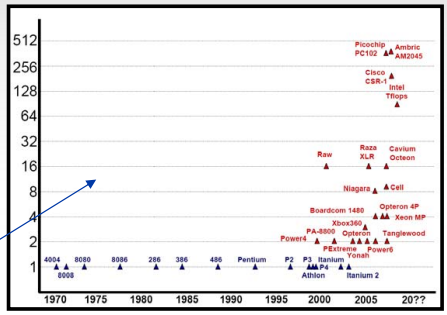


Source: Xilinx

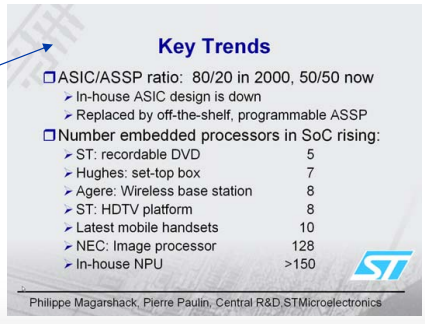
The real solution is Multi-Core

"Von Neumann is a poor use of scaling – all the energy is going on the communication between the processor and the memory. Its much better to use 20 microprocessors running at 100MHz than one at 2GHz"

Hugo de Man, IMEC

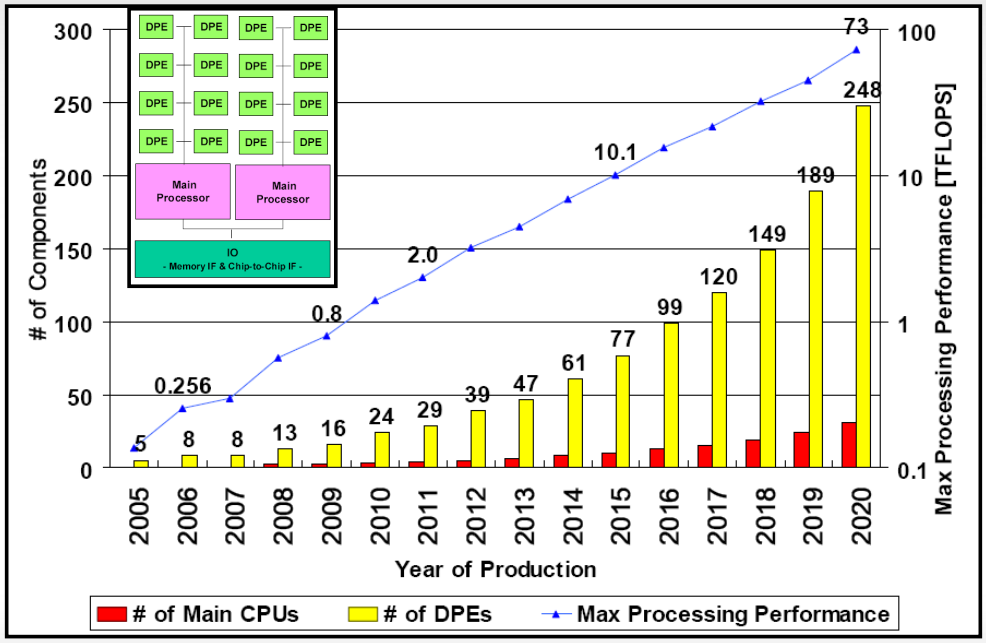


- Early movers have been building multi-core standard processors
- And more and more System on Chips (SoCs) and Platform chips are becoming multicore



Processor count predicted to increase dramatically

MULTICORE DESIGN SIMPLIFIED
Imperas



Source: ITRS 2006 Update
DPE: Data Processing Engine

Embedded software for MPSoCs: An extreme challenge!

MULTICORE DESIGN SIMPLIFIED
Imperas

"30 to 50 per cent of R&D budgets are spent on software, and the cost is rising 20 per cent a year. The software effort overtakes the hardware effort at 130nm."

Jack Browne, MIPS Technologies

"Some say we are at a crisis stage with the software side overwhelming the hardware side. Driving some of this is the proliferation of cores in system-on-chip (SoC) devices."

Steve Roddy, Tensilica

SW problems delay SoC revenues, impacting IP developer royalties

Productizing MPSoCs

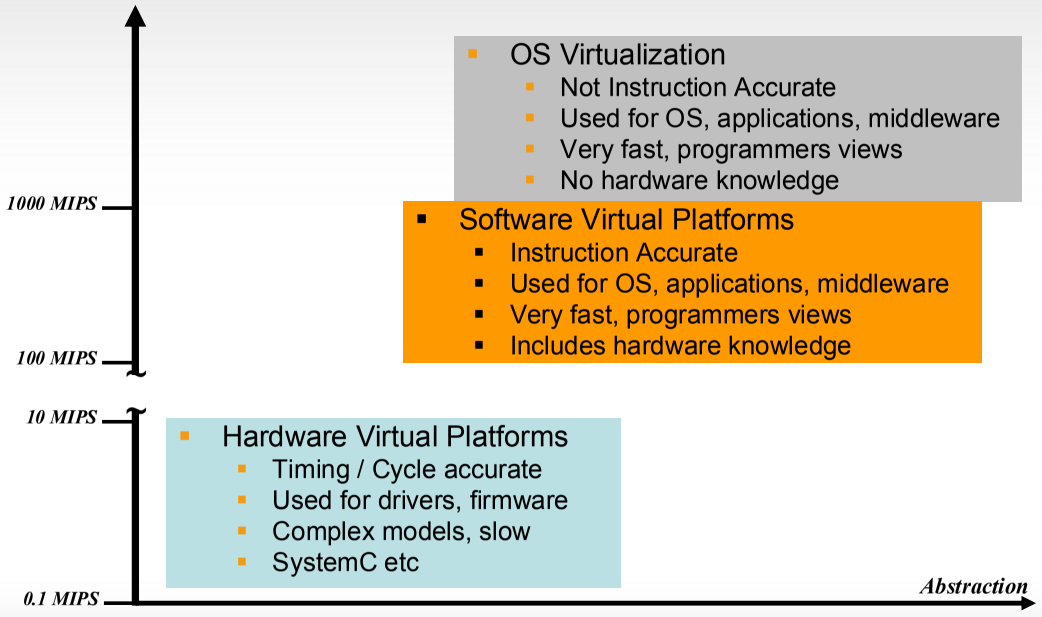
- So what does the future hold for software that is to run on these multi-core chips?
- Understanding what the future could look like, how can we make that better?

Current SoC (Software on Chip!) methodology

- Start developing application software in parallel with hardware
  - But cannot rigorously simulate software
  - Some static analysis tools used
- Get hardware prototype
  - The hardware is always far too late in the product development cycle
  - Also it is a limited, unreliable and not up to date platform
- Then there is a real challenge in verifying and debugging MP software
  - Lack of controllability, visibility, precision
  - And poor MP support
- More and more teams scrabbling about looking for a better solution
- Moving to Virtual Platforms for earlier testing of software, better software quality, lower software maintenance costs

Embedded software verification is 10-15 years behind hardware verification!

Virtual platform types



Virtual platforms market today is fragmented, and diverging

- Approaches
  - Roll your own – from scratch in C/C++/SystemC
  - Contract product provider as service to build models of IP and platform
  - Use commercial solution to write your own models
  - Existing open source
- Today's alternatives: advantages
  - Get a virtual platform
- Today's alternatives: disadvantages
  - Takes too long to develop
  - High risk
  - Lack of control for service model
  - Proprietary solutions
  - No interoperability
  - Cost of deploying simulation environment
  - Different solutions for different models/IP architectures
  - Designed for single cpu model not embedded MP platform

⇨ This fragmentation is not a good situation if virtual platforms are the enabler for MPSoCs

What is needed?

- **Open way of modeling needed for Virtual Platforms**
  - Targeted at Instruction Accurate Software Virtual Platform need
  - Easy to use, high level, 100s of MIPS
  - Covers complexity of current designs easily
  - Built-in interoperability for models from different developers
  - Proven and in use technology
- **Methodology that leads to an ecosystem**
  - Ability to enable model builders to protect IP
  - Interfaces that enable the growth of tool chains
  - Tools that provide verification, debug, analysis of embedded sw
- **Backwards compatible with legacy solutions**

- A solution for the people, by the people, of the people

Introducing
OPEN VIRTUAL PLATFORMS

- Software virtual platform infrastructure should be free and freely available
- Imperas is sharing, making public, and making free our simulation infrastructure technologies with the intention of establishing a common, open standard platform for software virtual platforms for software developers
- Imperas will support and manage the OVP site, and will contribute much of our innovation to keep this infrastructure evolving
- Participation of organizations and individuals around the world is critical to the success of OVP

OVP is …

- **Modeling APIs**
  - Publishing of C OVP APIs for Processor, Peripheral, and Platform modeling
  - Documentation & header files
- **OpenSource library of models**
  - C source of models written to C OVP APIs
  - Processor models of ARM, MIPS, OpenRisc OR1K (more later)
  - Peripheral models of standard embedded devices
  - Example embedded platforms in C, C++, SystemC
- **Free OVP reference simulator**
  - Runs processor models up to 500 MIPS
  - Interfaces to GDB, Eclipse via RSP/socket
  - Callable with C/C++/SystemC wrapper
  - MP Capable
  - Can encapsulate existing processor models

Open Virtual Platforms - Ecosystem (February 2008)

CALYPTO

AZUL SYSTEMS

MIPS TECHNOLOGIES

Sigmatix

denali

Jennic

criticalblue

Tensilica

NOVAS

Sibridge Technologies

eve

carbon design systems

FORTE DESIGN SYSTEMS

beyond SEMICONDUCTOR

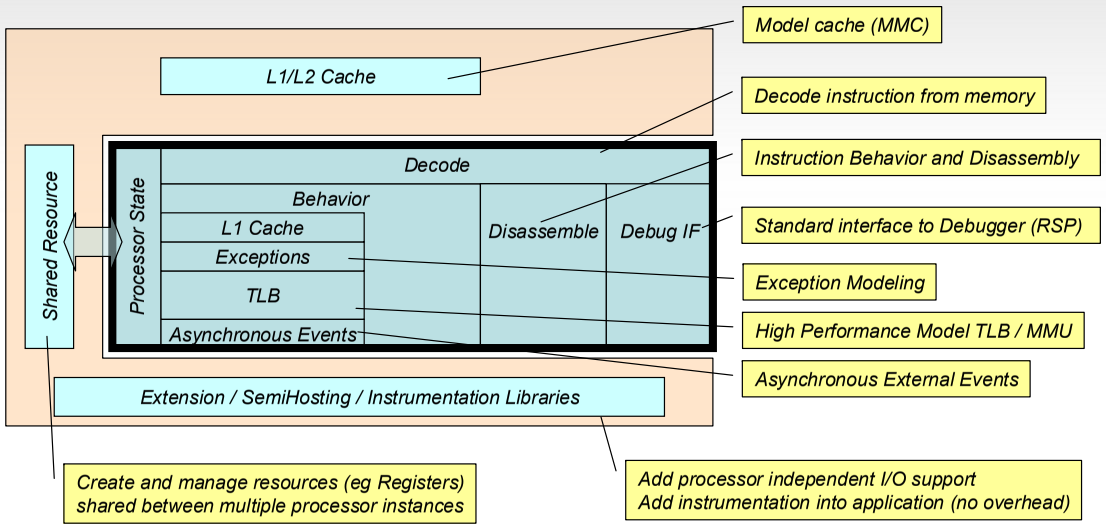
Brian Bailey Consulting

Prof. Don Thomas CMU

OVP technology

- Provided as open source through www.OVPworld.org and sourceforge
  - Capability to model processors, components & platforms
- OVP models: processor
  - C models using API
  - JIT (Just In Time) code morphing technology for up to 500 MIPS speed
  - Documentation, methodology, examples + support = develop Instruction Accurate models in 6-8 weeks
  - Encapsulation tools allow existing processor models to be integrated in hours
- OVP models: peripherals: easy C API
- OVP models: platform: easy C API
- Models have C, C++, and SystemC wrapping

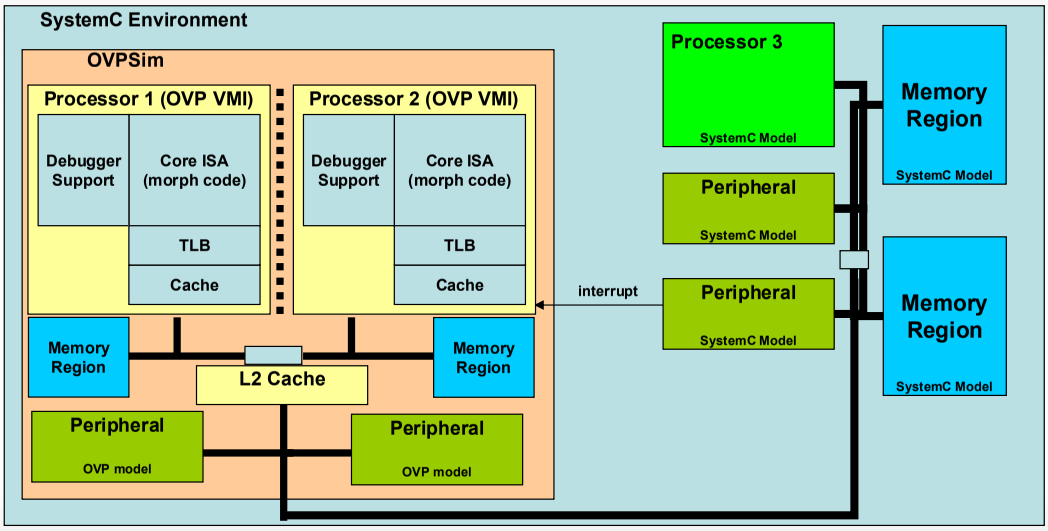
Components of OVP processor models



=> complete, proven technology and methodology

Easily transition to OVP

- Existing processor models can be encapsulated for OVP, and run in OVPsim
- OVP platforms can run under C, C++, SystemC, as well as standalone
- Platforms can be defined in SPIRIT IP-XACT compatible XML



OVP Summary

- The key to efficiency for future software development environments is an Open Virtual Platforms solution – especially with the move to MP
- The Imperas technology donation and formation of OVP kick starts a new phase in embedded software development
- OVP provides a fantastic OpenSource solution to be the foundation of the next generation of software development environments
- Committed collection of partners developing the ecosystem