What Do Embedded Software Developers Really Think About Multicore Architectures

25 June 2009

David Stewart | CEO

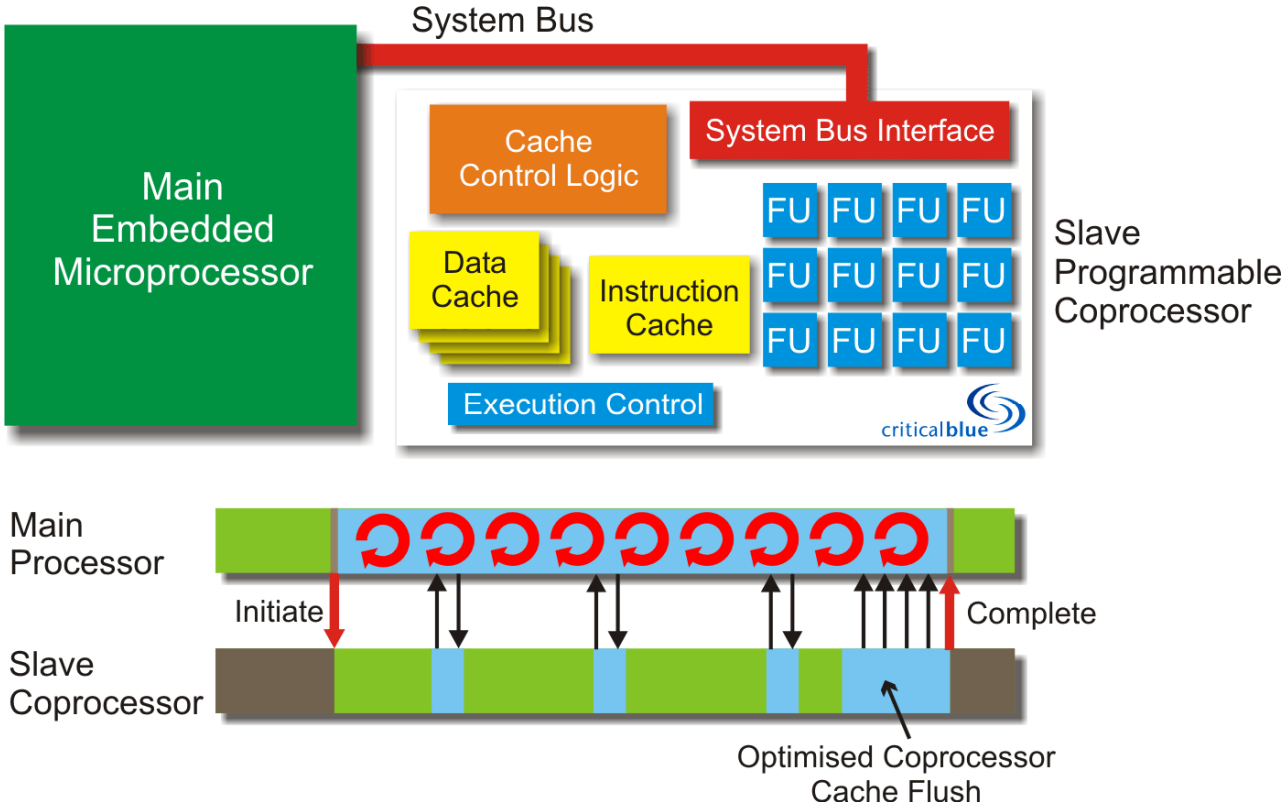
criticalblue
Accelerating Embedded Software

Agenda

- **Background**
- **Mainstream Multicore Programming**
- **What We Heard From The Market**

Background

- **2005: Introduction of our Cascade coprocessor synthesis product**



- **First indications that customers struggled with multicore software…**

criticalblue
Accelerating Embedded Software

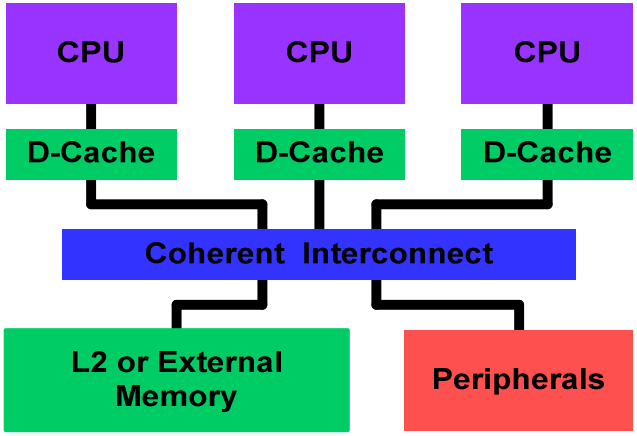
Multicore & Software: State of the Embedded Art (1)

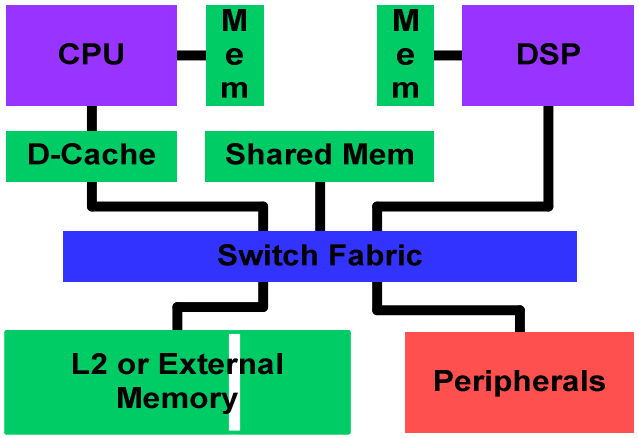
Desktop/Server/Telecom

Homogeneous
Shared memory
Coherent memory
eg x86 dual/quad
Sun UltraSparc T2

Embedded

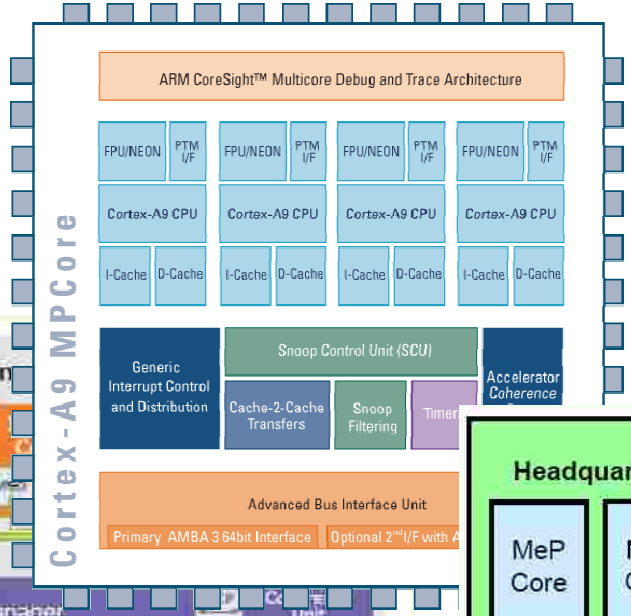
Heterogeneous
Distributed memory
Independent memory
eg TI OMAP
Most baseband ICs



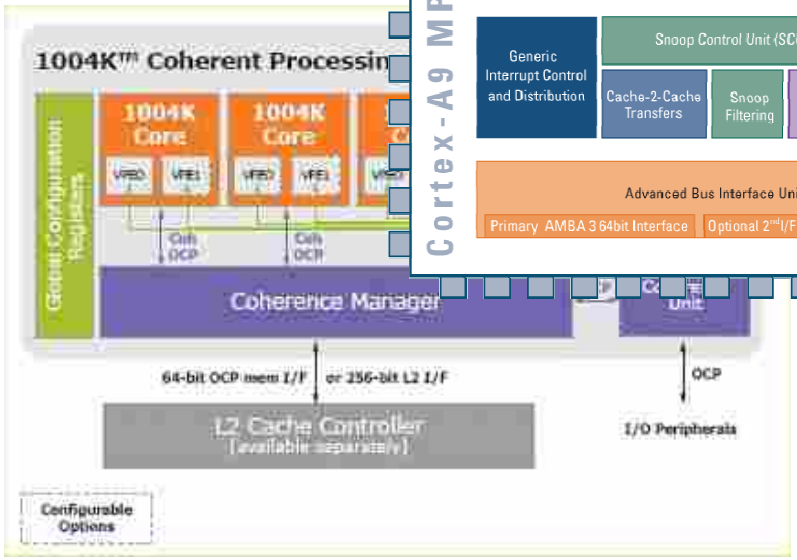


Multicore & Software: State of the Embedded Art (2)

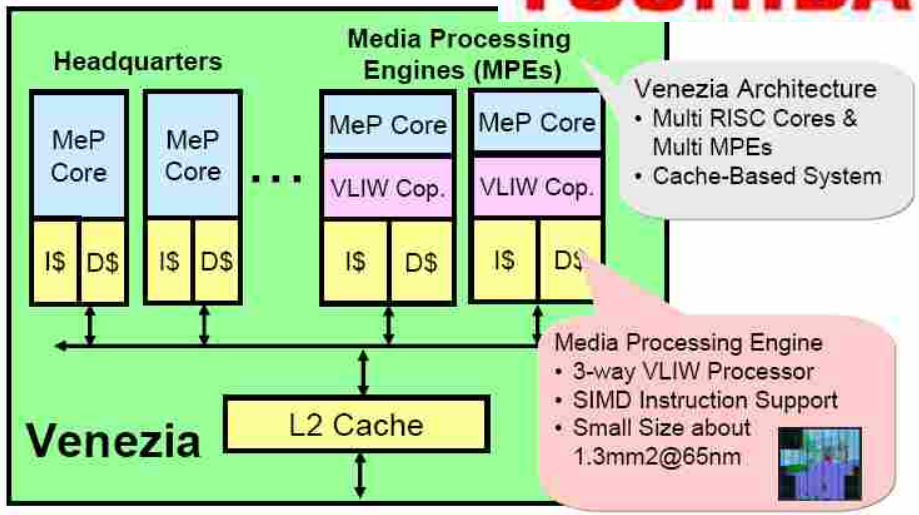
ARM



- ***Regular C language***
- ***Threading API***
- ***1-4 cores***
- ***Shared memory***



TOSHIBA



MeP: Media embedded Processor

MIPS TECHNOLOGIES

Multicore & Software: State of the Embedded Art (3)

- Putting lots of cores on silicon is relatively simple
- There are many clever memory and communication architectures
- However… the mass market of software developers haven't yet adopted multicore programming…
- Because…
- Software developers didn't ask for multicore!
- Software developers don't care about multicore block diagrams
- Software developers need motivation to migrate aggressively to multicore

criticalblue
Accelerating Embedded Software

What Did We Do Next?

- We developed some multicore analysis features
- We found some early adopters
- We started to realize that this issue was generic
- We wondered what a practical methodology and flow would look like
- We went out and asked a lot of questions
- We collated the feedback

- And this is what we heard….

criticalblue
Accelerating Embedded Software

Feedback From The Market (1)

- **We love the C language**
  - It's not that C is the perfect parallel language
  - 100s of millions of lines of existing code
  - Existing code maintenance is a huge overhead
- **We love our Compiler/IDE/ISS**
  - Companies typically standardize on a particular toolchain
  - Huge investment in the developers' desktop
  - Real world C/C++ is less portable than you might think
- **We need motivation to move software to multicore**
  - Benefits need to be clear before code changes are made
  - Multithreading simple to implement, hard to understand

criticalblue
Accelerating Embedded Software

Feedback From The Market (2)

- **Do not synthesize my code!**
  - Maintaining sequential code is hard enough already
  - Requirement to retain control of source code changes
  - Identify minimal set of code changes to deliver benefit
- **We must have confidence in code changes**
  - Check that the code modifications deliver expected benefits
  - Check that no new problems were added
  - Check that the code is 'safe
- **We need education in multicore programming**
  - Lots of FUD, but also plenty of products in the market
  - Plenty of buzzwords, less clear explanation
  - How does this relate to <u>my</u> code?

criticalblue
Accelerating Embedded Software

Collating The Data

1. Analyze – characterize existing code. Identify parallelism determining issues

What might a practical and pragmatic flow look like?

2. Explore – without code changes, evaluate alternative concurrent scenarios

3. Implement – select parallelization strategy and write code in existing IDE

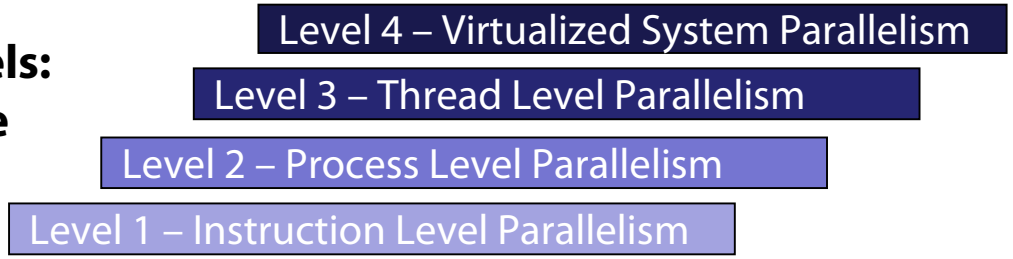
4. Verify – check for code safety, race, threading and synchronization issues

5. Tune – inspect for further performance opportunities on target platform

criticalblue
Accelerating Embedded Software

Providing the Foundation for Multicore Deployment

Parallelism Attainment Levels: A framework for multicore adoption



A focus on three areas to enable practical and effective solutions that allow development teams to attain required levels of multicore adoption:

- **Prism:** Tools to guide practical parallel programming starting with existing unmodified sequential code
- **Multicore Association MPP Founder:** Industry derived multicore programming practices, for portability, reuse, and efficiency
- **Processor Company Partnerships:** Collaboration with platform and processor companies to streamline and optimize design flows

criticalblue
Accelerating Embedded Software

Summary

- **We've made a start at a practical & pragmatic path for sequential to parallel software migration….**
- **Learn more at www.criticalblue.com**

- **Give us your feedback to make Prism even better!**