Overview

- Traditional System Development: A use case
  - Traditional design methodology leads to system architecture problems being discovered late in the development cycle
    - Leads to high re-work and long delays
  - HdS is a single system distributed over multiple heterogeneous processors interacting through a HW platform
    - HW/SW system testing required to validate

- System Development Using Virtual System Prototypes
  - Design → Implement → Test → Integrate → Re-design
    - Before the HW is manufactured
    - Significantly reduces risk and delay
  - Requirements of a virtual system prototype
    - Accuracy
    - Speed
3GPP Femtocell design: A Case Study

- Challenge: Design of a highly integrated UMTS 3GPP Release 6 femtocell base station
  - Low cost heterogeneous processor core design
- Features:
  - High Speed Packet Downlink and Uplink (HSDPA/HSUPA)
  - Radio Access Network “(RAN) in a box” collapsed stack architecture;
  - Session Initiation Protocol (SIP) enabled;
  - IP Multimedia Subsystem (IMS) enabled;
  - Media stream trans-coding;
Candidate Architecture

- 10/100 Ethernet
- SDRAM Interface
- I/D TCM
- ARM 926e
- I/D Cache
- DMA Controller
- DMA Controller
- I/D Memory
- TI c64x DSP
- I/D Cache
- RF Front End
- Analog DBB Front End
- 24 element DSP Processor Array
- Stream Buffer
- HW accelerator blocks

Bus Interconnect

ARM Developers’ Conference & Design Pavilion 2007
Candidate Architecture

Time Critical Software (UMTS PHY + framing):
- Sample rate processing
- Chip rate processing
- Symbol rate processing
- Interface to HW accelerator blocks
- Management and scheduling
- MODEM control and configuration interface
- High performance MAC-hs and MAC-b, etc.

10/100 Ethernet
SDRAM Interface
DMA Controller
I/D Memory
TI c64x DSP
I/D Cache

RF Front End
Analog DBB Front End
DSP Processor Array

Stream Buffer
HW accelerator blocks

159x129 Time Critical Software (UMTS PHY + framing):
• Sample rate processing
• Chip rate processing
• Symbol rate processing
• Interface to HW accelerator blocks
• Management and scheduling
• MODEM control and configuration interface
• High performance MAC-hs and MAC-b, etc.
Candidate Architecture

Real Time Software:
- Video and Audio trans-coding
- Stream encryption/decryption

Transport Interface:
- 10/100 Ethernet
- SDRAM Interface

Front End:
- RF
- Analog DBB
- RF Front End

Memory:
- I/D Memory
- I/D Cache

Processor:
- ARM 926e
- TI c64x DSP

DMA Controller

I/D TCM

Bus

DMA Controller

Bus

Stream Buffer

DSP Processor Array

HW accelerator blocks
Candidate Architecture

10/100 Ethernet
SDRAM Interface
I/D TCM
ARM 926e
DMA Controller
DMA Controller
I/D Memory
TI c64x DSP
I/D Cache

(Semi) Elastic Software running on Linux OS:
- IPsec/IPv4/IPv6 stack
- Services: Telnet, HTTP, FTP
- SIP client and server
- Network management services
- System control and configuration (inc. DMA setup)
- RAN signaling stack: SGSN/RRC/RLC/MAC

ARM Developers’ Conference & Design Pavilion 2007
Software Driven Design

- Majority of functionality is implemented in software:
  - Time critical data processing and control;
  - Real time data processing;
  - Elastic signaling and services.

- **Hardware Dependent Software:**
  - Hardware architecture must be sufficiently dimensioned to enable software to satisfy system requirements;

- Software is distributed over several heterogeneous processors;

- All of the software components need to collectively meet the system requirements;
Traditional System Development

Mistakes made here, are found here

Reference architectures
Ad-hoc modeling and analysis
Incumbent knowledge
HW/SW partitioning + algorithms

Basic simulated unit testing:
Using ISS, HW stubs, etc.

Unit testing
System testing

ARM Developers’ Conference & Design Pavilion 2007
Problem 1: User configuration over HTTP interface causes uplink packet loss

**Symptom:** For full rate uplink channel configurations, successive cache misses on the ARM926e cause stream buffer overflow.

**Cause:** Occurs when user configures system using HTTP interface and system is receiving at full uplink data rate. HTTP server not being in ARM926e I-cache causes bus contention.

**Solutions:**
1) Re-dimension bus interconnect (re-spin);
2) Increase ARM926e I-TCM (re-spin);
3) Increase stream buffer size (re-spin);
4) Reduce maximum uplink speed (cut specification)
5) Limit system configuration when operating at rated load (reduce features);
Problem 1 post-mortem

- Elastic software components (i.e. user configuration application) have had an adverse effect on time critical software components (i.e. MODEM and packet processing);
- Indirect interaction between software running on ARM926e (elastic) and MODEM DSP array (time critical);
- Analysis in system architecture stage failed to consider worst case behavior for asynchronous events under full load
  - Difficult to predict analytically because elastic (application) software footprint is not known at architecture design stage
Problem 2 found at integration stage

Problem 2: Cannot support full rate video trans-coding when all AMR voice channels are active

Symptom: Full rate video trans-coding drops too many frames (violates QoS constraint).

Cause: DSP does not have capacity to support video trans-code algorithm, reading and writing from external SDRAM, and full set of AMR voice CODECs simultaneously.

Solutions:
1) Consider moving AMR voice CODECs to MODEM DSP array (software re-design: but MODEM array does not have sufficient capacity – need a re-spin);
2) Add additional RAM block to DSP and use DMA transfer to reduce cycles required to read and write to packet buffers located in external SDRAM (re-spin);
3) Reduce feature set;
Problem 2 post-mortem

- Fundamental architecture design mistake
  - Was not found until algorithm was mapped to implementation;
- Assumption was made that voice CODECs could be implemented in MODEM DSP array (i.e. a Plan B):
  - Assumption was found to be wrong too late;
  - Needed to fall back on Plan B;
  - … BUT … Plan B was never an option!
- System SW partitioning mistake cannot be fixed because HW design cannot adapt to changing SW partitioning
  - Resulted in fundamental failure;
  - Requires HW architecture re-design = massive delays.
Problem 3 found at integration stage

Problem 3: Mobile originated voice calls are dropped

Symptom: Mobile originated calls are sporadically terminated.

Cause: Unknown, still trying to debug using JTAG interfaces and third party handsets.

Solutions:
1) No real solution, multiple debug iterations and tracing cycles as breakpoint are “single shot”.

ARM Developers’ Conference & Design Pavilion 2007
Lessons Learned

- A software driven embedded systems solution is:
  - A single software system distributed across multiple heterogeneous cores and supporting peripherals.
- Elements of the solution:
  - Interact with each other across processor cores in implicit and complex ways.
- The behavior of the combined solution is:
  - Difficult to analyze by conducting experiments on each component in isolation.
- Minor changes in software partitioning can:
  - Require major modifications to the hardware architecture.
- It is too late to identify architectural mistakes at the integration stage!
- Software engineers rely on a debugging environment that gives them complete control of the SW execution:
  - BUT the only way to give SW engineers a debugging environment where SW remains in “lock step” with HW is a using a HW-SW virtual platform.
Virtual Prototype Driven System Development

- Requirements
- Architecture + Algorithms
- Software Requirements
- Software Design
- Implementation
- Hardware Requirements
- Hardware Design
- Implementation
- Manufacture
- System testing

Implement → system validate → re-design

Mistakes made here, are already fixed before here
Virtual Prototype Requirements

Elastic (application) software:
Timing behavior of micro-architecture (e.g. Cache, MMU) is more important than static instruction timing cycle accuracy.

Real time software:
High level of instruction cycle timing accuracy + micro-architecture behavior required.

Time critical software:
Precise level of instruction cycle timing accuracy required.

All SW components interact via the HW platform.

Accuracy requirements are different for each system component. However, the entire system model has accuracy requirements.

ARM Developers’ Conference & Design Pavilion 2007
Virtual Platforms

A virtual platform is a “lock step” model of:
   - More than one heterogeneous processor core;
   - A set of peripheral HW models;
   - The complete HW/SW system.

Virtual platforms:
   - Enable HdS to be system tested on the hardware platform before the hardware is available;
   - Models the critical behavior of the hardware platform;
   - Facilitates identification of fundamental architectural problems;
   - Provides an opportunity for architecture re-design and HW/SW re-partitioning;
   - Can be used for architecture exploration and optimization;
   - Is a tool that can be used to debug hard to find problems with the final product.
Conclusions

- Software Driven Embedded Systems require new tools to successfully integrate the SW into the HW;
- Virtual platforms used to develop Software Driven Embedded Systems need to:
  - Needs to be fast to run real software loads;
  - Be able to model multiple heterogeneous processor cores in “lock step”;
  - Need to accurately model critical timing and behavior of the hardware architecture;
- Modern Software Driven Embedded Systems design using virtual platforms extends agile development techniques beyond SW development to the complete development of the HW/SW system:
  - Design → Implement → Test → Integrate → Re-design