**Platform-based Design: an Automotive Example**

Alberto Sangiovanni Vincentelli  
The Edgar L. and Harold H. Buttner Chair of Electr. Eng. And Comp. Science  
University of California at Berkeley  
Co-Founder, Chief Technology Advisor and Board Member  
Cadence Design System  
Founder and Scientific Director  
PARADES (Cadence, Magneti-Marelli, ST)  
Work in collaboration with  
A.Ferrari, M. Baleani, L. Mangeruca (PARADES)  
And the Magneti-Marelli/ST Design Teams

---

**HW Micro-controller Architecture basic components**

- Processing Units (CPU)
- Memories
- I/O Units

---

June 9, 2000
An Example of Platform space: Micro-controllers

The criteria for selection are many...

- Cost
- Performance
- Flexibility
- Reliability
- Size
- Power
- Re-use
- ...

June 9, 2000
Power-Train Control System

- Electronic device controlling an internal combustion engine and a gearbox
- The goal
  - offer appropriate driving performance (e.g. torque, comfort, safety)
  - minimize fuel consumption and emissions
- Relevant characteristics
  - strictly coupled with mechanical parts
  - hard real-time constraints
  - complex algorithms for controlling fuel injection, spark ignition, throttle position, gear shift ...

Engine Management: Behavior

- Failure detection and recovery of input sensors (6 CFSMs + 1 Timer)
- Computation of
  - engine phase, status and angle (6 CFSMs + 6 Timers)
  - crankshaft revolution speed and acceleration (3 CFSMs + 1 Timer)
- Injection and ignition control law (18 CFSMs)
- Injection and ignition actuation drivers (56 CFSMs + 48 Timers)
Behavioral Validation

Simulation

- I/O traces of 4s of engine management
- Fault injection (sensors)
- About 20s on a 450MHz PII CPU with 256MB RAM

Single-Core Architectural Modeling

Hitachi SH2 or ARM7TDMI core based μ-controller
### Mapping to SH2: SW Execution time

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Execution Time (microsec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAP</td>
<td>TDC</td>
</tr>
<tr>
<td>CPU Load</td>
<td>2%</td>
</tr>
<tr>
<td>CPU Load Error</td>
<td>11.3%</td>
</tr>
</tbody>
</table>

### Core Exploration: ARM7TDMI

<table>
<thead>
<tr>
<th>Task Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (microsec)</td>
</tr>
<tr>
<td>GAP</td>
</tr>
<tr>
<td>CPU Load</td>
</tr>
</tbody>
</table>

**GAP**

**TDC**

**MTDC**

**Error**

**SH7055**

**ARM7TDMI**

**Speed Up**

**GAP**

**TDC**

**MTDC**

**Error**
**ARM7TDMI Code Size**

Code Size (published, publicly available, benchmarks):

- ARM7 (32 bit mode) is 25-35% larger than ARM7TDMI
- Hitachi SH2 is 20% larger than ARM7TDMI
- CPU32 (68K) is 30% larger than ARM7TDMI
- PPC is 60-100% larger than ARM7TDMI
- ST10/C167 is 15-35% larger than ARM7TDMI
- M-Core same code size

**I/O exploration (Sensor Management)**

- **Mapping A**
  - maximize software re-use and portability (i.e. not optimized for Hitachi architecture, only “essential” timers included)
  - 13 CFSMs + 5 timers
  - A1 two tasks, A2 three tasks

- **Mapping B**
  - optimize for the Hitachi architecture (i.e., utilize special purpose peripheral called ATU that makes more timers available)
  - 15 CFSMs + 7 timers
  - B1 two tasks, B2 three tasks

- **Mapping C**
  - minimize hw/sw communication introducing a new virtual hardware component (i.e. replaces ATU with additional functionality)
  - 18 CFSMs + 6 timers
**Mapping: performance**

<table>
<thead>
<tr>
<th>Mapping</th>
<th>A1</th>
<th>A2</th>
<th>B1</th>
<th>B2</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU load*</td>
<td>10.2%</td>
<td>10.3%</td>
<td>6.7%</td>
<td>6.8%</td>
<td>1.4%</td>
</tr>
<tr>
<td>IRQs</td>
<td>7739</td>
<td>7739</td>
<td>8626</td>
<td>8626</td>
<td>1528</td>
</tr>
<tr>
<td>Task switching</td>
<td>7616</td>
<td>7745</td>
<td>7616</td>
<td>7745</td>
<td>1673</td>
</tr>
<tr>
<td>Task number</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

*Deferred ISR scheme used

**Flash and Code Size Trend**

- **Discrete Flash**
- **Embedded Flash**
- **Code Size**

June 9, 2000
**Flash Speed**

![Memory Speed Chart](chart.png)

- Embedded: 60Mhz (ISSCC@1999)
- External: 20-30Mhz (AMD/STM/INTEL)

---

**Platform Cost**

- **External Flash:**
  - Cost = Cost(Flash IC) + Cost(MCU) + IntegrationCost(Flash, MCU)
  - Micro-controller likely to be I/O bounded

- **Embedded Flash:**
  - Cost=Cost(MCU) + IntegrationCost(MCU)

**Other issues: speed, reliability, re-use, ...**
**Embedded Flash Solution**

**Embedded Flash:**
- Speed: embedded is faster
- Cost: embedded is cheaper
- Size: micro-controllers with higher code density reduce system cost since size is mostly dictated by Flash

---

**Power-Train Control System**

Extend behavior to gearbox control

ARM7/SH2 core overloaded

**Solutions:**

- **Increase Clock Frequency**
  - Performance bounded by memory (FLASH)
  - A cache memory must be introduced.
  - Higher clock frequency (80Mhz) might have a worse EMI impact.

- **Increase Parallelism**
  - Instruction Level: VLIW (larger code size)
  - Processor Level: **Multiprocessor**
The Dual-Arm Architecture

A symmetric dual processor architecture with a high-bandwidth interconnection network among processors, memory, and I/O sub-systems

The micro-controller has been designed as a collaborative effort among:

- PARADES for architecture concepts
- Magneti-Marelli for peripherals and requirements
- ST for detailed architecture and IC design
- Accent for X-bar switch and Interrupt Controller detailed design
- Cadence for system exploration and evaluation tools and methodology

Excellent example of supply-chain integration in electronic system design for next generation platforms!
**Why dual-core?**

A simple back of the envelope calculation

- **Estimated size:**
  - (32KB.RAM + 512KB.FLASH) ~ 5.8M Trs. equivalent to 65-75% of chip area
  - (ARM7TDMI+Debug+IC+...) < 200K Trs. equivalent to less than 4% of chip area
- The MCU size is driven by FLASH size

**Dual-core solution:**
- 4% increment of MCU cost
- twofold performance

---

**Dual-ARM VCC Architectural Model**
**Behavioral Mapping**

- **Engine Management** $\rightarrow$ **CPU1** (as single core)
- **SeleSpeed** $\rightarrow$ **CPU2**

**Interrupt Latency**

- XBAR arbitration policy: round robin and 1 clock owner cycle
- Best Case: 15 clks
- Worst Case with 1 master: 26 clks
- Worst Case with 2 masters: 37 clks
- Worst Case with 3 masters: 42 clks
- A ‘typical ‘ value of 27 clk (by simulation)
**Interrupt Latency (2)**

![Graph showing response time (CLKs) over simulation time (sec).]

**Platform Definition**

![Diagram illustrating platform definition and exploration.]

---

June 9, 2000
Conclusions

- Complex application of system design methodology: automotive power-train control
- Several architectures (different CPU, different I/Os, different memory organization, e.g. external vs. internal flash) evaluated
- New dual-core architecture developed and validated for Magneti Marelli applications by joint design teams (PARADES, Magneti-Marelli, ST)

The Past, the Present and the Future of the Dual-core Platform

- Architecture Conceived in PARADES in March 1999
- First Discussion with M.M. and ST in April 1999
- Specification Team with M.M. and ST started in Sept'99
- Performance exploration started in Dec'99
- Close specification in Jan’00 (IRQ list)
- FPGA Prototype in Oct ’00
- Tape out Jan ’01 (0.18 μm)
- First Silicon 2001
- Architecture spins for different applications 2001-2002
- MM power-train controller Start-Of-Production mid-end 2003