Rationale for Multicore Architectures in Auto Apps

FTF-AUT-F0166

Joe Circello
Freescale Fellow, MSG Core and Platform Chief Architect
Agenda

- Review Technology and Core Architecture Constraints
- Consider Applicability of Multicore Architectures
- Review Freescale’s Current Multicore Automotive MCUs
- Market Challenges Driving 55 nm Cores and Platforms
- Consider Multicore Software Issues
- Summary
Abstract

Following the architectural trend in desktop computing, new devices for deeply embedded applications are increasingly adopting multicore architectures to address ever-increasing performance requirements while being constrained by power dissipation limitations.

This trend is reflected in various Freescale automotive microcontrollers ranging from "simple" loosely-coupled dual core architectures to more advanced next-generation 55-nm architectures that contain up to five Qorivva 32-bit processors in a single device.

This session discusses the rationale used in guiding the definition of these multi-core Qorivva 32-bit MCU architectures for the automotive application space, and discusses some of the software challenges to best use the microcontroller hardware.
Performance Scaling and Technology Challenges

- Clock rate improvements slowing: 40%/year → 12%/year
  - Pipelining has increased by factor of 4 in last decade
    - not possible in next decade

Source: UT Dept. Computer Science

8-10 FO4 Pipeline

Pipelining

Semiconductor Technology

Historical
Microarchitecture
Technology

40x
6.6x
10x
How Did We Get Here? Physics and Delays
**How Much Logic Can Be Touched in One Cycle?**

- Transistors continue to get faster, but wire delays begin to dominate
- Historical solution …
  - Divide and conquer with longer pipeline = Less work per cycle

<table>
<thead>
<tr>
<th>At $f$ MHz</th>
<th>At $6f$ MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
</tbody>
</table>
Time for a New Approach

• (Fred) Pollack’s Rule: The increase in performance is roughly proportional to the square root of the increase in complexity. That is:
  - 2x the core logic means a $\sqrt{2} \approx 1.4x$ performance increase
  - Advanced architectural features including:
    - Pipelining, multi-issue, VLIW, speculation, out of order execution …
• In 1999, Intel designers showed if trends in high-performance microprocessors continued, “by 2010 they’d burn as hot as the surface of the sun. The answer was clear: slow down the CPU’s clock and add more cores.” ¹
• Power dissipation: $P = CV^2f$
  - Product of capacitance (C), voltage squared ($V^2$), frequency (f)
  - Power densities exceeding Dennard’s scaling rule due to aggressive MHz increases and the reduction in voltage scaling effects
• Architecture trends from desktop, laptop and server spaces typically migrate into deeply embedded microcontroller spaces

Are Multicore Configurations the Answer?

- "Simple" options for higher performance
  - Double the core speed (1 CPU @ 2f MHz)
  - Double the cores (2 CPUs @ f MHz)
  - Double the performance (1 2xCPU @ f MHz)
- All approach twice the reference
  - Reference = 1 CPU @ f MHz
- **Multicore has better performance per mW**

![Performance vs. Time Graph]

- Reference = 1 CPU @ f MHz
- 2 CPUs @ f MHz
- 1 CPU w/ 2x gates @ 1.41f MHz
- 1 CPU w/ 3x gates @ 1.15f MHz
- 1 CPU w/ 4x gates @ 1.00f MHz

<table>
<thead>
<tr>
<th>Reference = 1 CPU @ f MHz</th>
<th>2 CPUs @ f MHz</th>
<th>1 CPU w/ 2x gates @ 1.41f MHz</th>
<th>1 CPU w/ 3x gates @ 1.15f MHz</th>
<th>1 CPU w/ 4x gates @ 1.00f MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate Performance</td>
<td>Power</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
More on Multicore in Embedded Applications

• Frequency scaling constraints
  - Package cost
  - FM frequency band
  - External component cost

• Multicore processors viewed as most viable approach to achieve required performance gains within power budgets

Automotive Multicore Qorivva 32-bit MCUs

• New challenge
  - Software for multicore processors

\[ P = CV^2f \]
Processor Configuration Taxonomy

Processor Core Hardware Configurations

Uniprocessor, Single Core
- Dual Core, Shared Memory
  - Asymmetric
  - Symmetric
- Multi-Core, Shared Memory
  - Asymmetric
  - Symmetric
- Many-Core
  - Dual-Core, Lockstep
  - Symmetric

Freescale’s Current and Near-Term Auto MCU Focus
A Long History of Multicore Innovations

• Originally described as programmable coprocessors
  - 1989: MC68302 = 68000 + RISC Comm Processing Module
  - 1993: MC68360 = ‘020 + Quad Integrated Comm Controller
  - 1995: MPC821 = PowerPC + QUICC
    Power QUICC I, II, II Pro, III, … today’s multi-core QorIQ
  - 1995: MC68332 = ‘020 + Time Processing Unit (TPU)
  - 1998: MPC555 = Power PC 555 + TPU
  - 2004: MPC5554 = e200z6 + 2x eTPUs

• Migrating to more “traditional” dual cores + shared memory
  - 2000: Customer-specific dual V4 ColdFire core
  - 2004: S12DX512 = S12X + XGATE
  - 2007: MPC5510 = e200z0 + e200z1
## Today’s Multicore Automotive MCUs

### I/O Processor Solutions

- **CPU offloading of specific low-level I/O tasks**
- **Peripheral emulation and flexibility**
- **Parallel gateway communications processing**

### High-Performance Symmetric Multiprocessor SoC

- **Increased MIPS/MHz**
- **Hardware support for both SMP and AMP applications**
- **Improved current consumption and EMC performance**

### S12X

<table>
<thead>
<tr>
<th>S12X</th>
<th>XGATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus Switch</td>
<td>Mem</td>
</tr>
<tr>
<td>Periph</td>
<td>Mem</td>
</tr>
</tbody>
</table>

### Qorivva MPC5554

<table>
<thead>
<tr>
<th>Power Architecture e200z6</th>
</tr>
</thead>
<tbody>
<tr>
<td>eTPU</td>
</tr>
<tr>
<td>Crossbar</td>
</tr>
<tr>
<td>Periph</td>
</tr>
</tbody>
</table>

### Qorivva MPC5510

<table>
<thead>
<tr>
<th>Power Architecture e200z1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Architecture e200z0</td>
</tr>
<tr>
<td>Crossbar</td>
</tr>
<tr>
<td>Periph</td>
</tr>
</tbody>
</table>

### Qorivva MPC5676R

<table>
<thead>
<tr>
<th>Power Architecture e200z7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossbar</td>
</tr>
<tr>
<td>eTPU</td>
</tr>
<tr>
<td>Periph</td>
</tr>
</tbody>
</table>

### Qorivva MPC5643L

<table>
<thead>
<tr>
<th>Power Architecture e200z4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Architecture e200z4</td>
</tr>
<tr>
<td>Crossbar</td>
</tr>
<tr>
<td>Periph</td>
</tr>
</tbody>
</table>
Typical I/O Processor Use Cases

Peripheral Emulation

I/O Processor

GPIO Port

Multiple smart PWM channels emulated on GPIO port

ADC Inputs

Low Power

Full chip cyclic wakeup

Smaller core (IOP) or 2 cores running, reducing “on” time

Current Consumption and EMC

Single core, high frequency operation

Main core running at reduced speed and IOP periodically helping
Main Core + IOP = Qorivva MPC551x

- MCU family targeted at central body and gateway applications
  - Shared memory, hardware semaphores
  - 0.5 – 1.5 MB flash
  - 32 – 80 KB RAM
  - $f_{\text{max}} = 80$ MHz
- Dual e200zX Power Architecture® cores (z1, z0)
  - Common instruction set architecture (ISA) with binary compatibility
  - Different microarchitectures with different capabilities
- Multiple low-power sleep modes
The Ultimate Gateway: Qorivva MPC5668G

- Gateway controller: Manage communications networks and data traffic
  - CAN, FlexRay, LIN networks
  - MOST ring, external access
- MCU highlights
  - Dual Power Architecture cores
    - 116 MHz e200z6, 58 MHz e200z0
    - > 250 DMIPS performance
  - 2 MB flash
  - 592 KB RAM
  - Integrated Ethernet, FlexRay, MLB controllers
SMP + Lockstep Safety = Qorivva MPC564xL

• Targeted at chassis + apps requiring high performance and/or safety
  - IEC61508, ISO-26262 standards

• MCU highlights
  - Dual 120 MHz e200z4 cores
    ▪ 2-way superscalar dispatch
    ▪ 2 configurations
      • Dual-core lockstep for safety
      • Independent CPUs for high performance
  - 1 MB flash
  - 128 KB RAM
  - Integrated functional safety
e200zX Core Roadmap

Common instruction set architecture,
Different microarchitecture implementations

- **e200z0**
  - 80 MHz
  - FPU Only
- **e200z1**
  - 80 MHz
  - VLE Only
- **e200z3**
  - 80 MHz
  - VLE
- **e200z4**
  - 150 MHz
  - Dual Issue + VLE
- **e200z5**
  - 200 MHz
  - Up to 32K cache
  - FPU
  - SIMD
- **e200z6**
  - 144 MHz
  - Up to 32K cache
  - FPU
  - SIMD
- **e200z7**
  - 300 MHz
  - 10-stage pipeline
  - Dual Issue + VLE

Time

Performance / Features
Next-Generation 55 nm Low Power Challenges

- Very aggressive market power targets
  - OEMs and Tier 1s asking for ~40% reduction
    - Allow eco-sensitive “green” designs to coexist with new features
  - Process variation: worst-case current >> average current
    - Automotive requires guarantees, not average
    - Performance vs. power tradeoff in worst case

- Requires various design implementation techniques
  - Leakage: Power gating, multiple low-power modes …
  - Dynamic: Clock gating, reduced signal toggling, optimized clock trees …
  - Power management complexity

- And architectural simplifications
55 nm Qorivva 32-bit Auto Microcontrollers
Philosophy Driving Core Architecture Changes

• To maintain the Qorivva 32-bit MCU leadership position:
  - Focus on “differentiators” while reducing complexity
    ▪ Power
      • Adoption of tightly-coupled processor local memories, two-way SA caches
      • Reduce overall core gate counts
    ▪ Performance
    ▪ Functional safety
      • Selective replication operating in delayed lockstep
      • End-to-end Error Correcting Code (e2eECC) implementations
  - Simplify software and strengthen the tool chain
  - Revisit feature set: use available area/power for key auto needs
    ▪ MMU is superseded by MPU; hierarchical core and system MPUs
    ▪ Application-specific DSP capabilities + scalar FPU leverages dual issue (superscalar {ld/st + FPU} dispatch)
e200zX Performance vs. RunIDD Comparison

The e200 cores shown in this chart are (left-to-right):
- z0, z1, z3, z6,
- z420, z446,
- z720, z750

Performance, RunIDD are calculated at product frequency [MHz].
Next-Generation Platform Architecture Changes

- Where appropriate, adoption of a dual platform shell SoC
  - Driven by power management concerns
  - Computational platform shell operating at \( f \) MHz
    - Main core(s), optional safety checker core
    - Flash + SystemRAM
  - Peripheral platform shell, operating at \((f/2)\) MHz
    - I/O processor, alternate bus masters (DMA, FlexRay, Ethernet …)
    - Slave peripheral subsystems
  - Intelligent bus bridging gaskets provide inter-shell connections
  - Hierarchical memory and memory protection system
    - Memories: \{I,D\}-$, TCMs > Flash + SystemRAM > Inter-shell Xfrs
    - MPU: Core MPU > System MPU > Peripheral Access Control
Next-Generation Qorivva 32-bit Auto MCUs

- Consider the following “generic” high-end multi-core Qorivva 32-bit MCU:
  - Computational platform shell running at $f$ MHz
    - Two main cores + safety checker core operating in delayed lockstep
      - Cores with local I- and D-caches and/or tightly-coupled {I,D}MEMs
    - Shared flash memory with calibration data remapping capabilities
    - Shared SystemRAM
  - Peripheral platform shell running at $f/2$ MHz
    - IOP core with local cache(s) and/or tightly-coupled {I,D}MEMs
    - Alternate bus masters including DMA, FlexRay, Ethernet …
    - Security module: a “microcontroller within a microcontroller”
      - Qorivva e200zX core + security accelerators + local memories
    - Slave peripheral subsystems
Example 55 nm Qorivva 32-bit MCU Core Platform

- Top-level core platform consists of two platform shells
  - **Computational** shell containing main cores + memories
  - **Peripheral** shell containing I/O processor + other bus masters + connections to peripheral subsystems
- Provides structure that best manages the power dissipation requirements and safety aspects of family
  - Supports multiple operating frequencies + hierarchical on-chip memory organization
  - Safe operation including end-to-end ECC, balancing system performance versus power dissipation
Multicore Software Architectures

• Macro software architecture issues
  - Static assignment of tasks to cores
    ▪ Examples: Peripheral interrupt service routines to IOP
    ▪ Powertrain partitioning of angle vs. time tasks to different CPUs

• Micro software architecture issues
  - Allocation of tasks and variables to hierarchical memory spaces
    ▪ Three-level hierarchical memory system
      • Processor-local memories > system flash and RAM > inter-shell accesses
      • All memory regions are accessible to all processors, but:
        - Performance implications depending on the memory’s location
    ▪ Data coherency
    ▪ Task synchronization and interprocessor signaling
• AUTOSAR R4.0 adds support for multicore architectures

• Hardware assumptions
  – CPU hardware
    ▪ Core identification, atomic read, write and read-modify-writes
    ▪ Same instruction set architecture, same data representation
    ▪ CPU-to-CPU interrupt mechanism
  – Memory
    ▪ Shared flash and RAM, single address space

• Software limitations
  – Tasks are statically assigned to specific cores
  – Resource allocation supported on a per-CPU basis
  – No shutdown, restart of individual cores
Proposed Software Partitioning

Operating System
- System Services
- Memory Services
- Communication Services
- Onboard Device Abstraction
- Memory Hardware Abstraction
- Communication HW Abstraction
- Microcontroller Drivers
- Memory Drivers
- Communication Drivers

Application Layer
- Application Software Component
- Application Software Component
- Application Software Component

Application Programming Interface (API)
- Basic Software
- Complex Driver
- I/O Drivers
- I/O Hardware Abstraction

Microcontroller

Static assignment:
- Main Core
- Second core
- Core choice depends on load balancing
## Optimizing Qorivva 32-bit MCUs

<table>
<thead>
<tr>
<th></th>
<th>MPC500</th>
<th>MPC5{5,6}00</th>
<th>MPC5700</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core Architecture</strong></td>
<td>PowerPC Classic</td>
<td>BookE + VLE</td>
<td>Next Gen 55 nm</td>
</tr>
<tr>
<td><strong>Optional CPU Units</strong></td>
<td>FPU, Decompress</td>
<td>FPU, SPE (DSP)</td>
<td>VLE</td>
</tr>
<tr>
<td><strong>Implementation Focus</strong></td>
<td>Single core</td>
<td>Single + dual core</td>
<td>FPU, LSP, MPU</td>
</tr>
<tr>
<td><strong>SoC Microarchitecture</strong></td>
<td>Single-master, core-centric</td>
<td>Multi-master, platform-centric</td>
<td>Multicore, lower power, performance</td>
</tr>
</tbody>
</table>

Optimizing architectures for leading performance and efficiency
55 nm Qorivva 32-bit MCU Summary

• Built upon Freescale’s long history of multicore innovations and the industry’s most powerful automotive architecture to offer exceptional scalability to enable a new generation of smarter, safer, more connected vehicles

• Spanning from uniprocessor to multicore variants, Qorivva 32-bit MCUs provide increased performance, security and safety for the latest vehicle applications

• Now more than ever, Freescale is a technology partner and supplier the auto industry can turn to for innovative solutions that meet their performance, efficiency, reliability, quality and cost objectives

Session materials will be posted @ www.freescale.com/FTF

Look for announcements in the FTF Group on LinkedIn or follow Freescale on Twitter