Infineon's Multi-Core Automotive Microcontroller for Hard Real Time and Safety Applications

Dr. Albrecht Mayer
TU Darmstadt, May 2014
Challenging Embedded System

- Hard real time
- Harsh environment
- High reliability
- High safety
- Medium security
- Low cost

Welcome to Automotive!
Specifics

Performance
- Hard real time → meet all deadlines

Safety
- Minimize probability of faults
- Detect faults with high probability
- Provide proof for safety claims

Cost
- More than cents but no need for large bills
Safety on System Level

- **sensing**: TLE35584 Safety Power Supply
- **calculation**: 32-bit Lockstep MCU
- **data transfer**: CAN Transceiver TLE 6251D TLE7255G
- **sensor feedback**: TLE9180
- **sensing**: OptiMOS™-T2 40V 7x IPB180N04S4-01
- **diagnostics**: Rotor Position iGMR Sensor TLE 5009/12
- **memory protection (MPU)**: 3-Phase Driver IC TLE4998
- **fail safe**: TLE7255G
- **protection**: TLE4998
- **redundancy**: TLE7255G
- **diagnostics**: Watchdog
- **fail safe**: MC23xL
- **safety switch**: Watchdog

*Copyright © Infineon Technologies AG 2014. All rights reserved.*
Matlab/Simulink Representation of an Embedded System

- Control algorithm design
- Mapping to target platform
Top-Down Design + Code Generation

2.2.10 **Left Outer Actuator in Passive mode**

If the Left Outer actuator is in the Passive mode, and the Left Inner actuator is in the Active mode, transition the Left Outer actuator to the StandBy mode. Else transition the Left Outer actuator to the Active mode.

2.2.11 **Left Inner Actuator in Passive mode**

If the Left Inner actuator is in the Passive mode, and the Left Outer actuator is in the Active mode, transition the Left Inner actuator to the StandBy mode. Else transition the Left Inner actuator to the Active mode.
Timing in Embedded Systems

Source: Timing Architects Embedded Systems GmbH
Timing in Embedded Systems (without Scheduling)

Source: Timing Architects Embedded Systems GmbH
Timing in Embedded Systems (without Scheduling)

Source: Timing Architects Embedded Systems GmbH
Timing in Embedded Systems (with Scheduling)

Response Time

Source: Timing Architects Embedded Systems GmbH
Mixed Criticality

Homogeneous safety requirements

QM: Quality Management

Source: ATZ Electronic 1/2012 (Audi, Symptavision)
Rate Monotonic Scheduling (RMS)

Source: ATZ Electronic 1/2012 (Audi, Symptavision)
Criticality Aware Priority Assignment (CAPA)
CAPA + Period Transformation + Timing Protection

Source: ATZ Electronic 1/2012 (Audi, Symptavision)
Parallelize your software – start partitioning from a high level!

- Old / new functionality
- Fuel injection / engine and vehicle control
- Angle (e.g. ignition) / time bound tasks (e.g. diagnosis, CAN, temperature)
- Non-redundant / redundant software (for safety)
- ...

```
void function1(int arg) {
    // start independent block
    ...
    // end independent block
    for (;;) {
        // independent loop runs
    }
}
```

```
void function2() {
    
}
```

Task_10ms

Task_20ms

Independent task types

Independent tasks

Independent functions

Independent blocks within functions

- Advised: Start with split on a high level, and not on low level!
- Usually manual split necessary
- Constraint: Keeping real time and criticality
- To optimize: Scalability, maintainability, communication overhead, memory consumption, power
Problems to be addressed for multi-core software

Correctness
- Avoid deadlocks, race conditions, priority inversion
- Those problems can also occur on single core systems; however, they occur in particular in single core software ported to multicore architectures

Performance: communication overhead
- Latency: Meet scheduling constraints
- Throughput

Scalability
- Support not only one multicore device but a complete family of devices with different number of cores; keep software development costs under control
• Collision

[Diagram showing a process with overlapping timelines and a label "Blockage of Process"]

Source: Timing Architects Embedded Systems GmbH
Causality

Parallelization

Source: Timing Architects Embedded Systems GmbH
Causality

Blockage of Process

Source: Timing Architects Embedded Systems GmbH
What does Software Mapping mean?

- Map functionality to cores; map code and data to memories
- Major mapping criteria: Core performance, support for lock step, memory access latencies and sizes, access conflicts
Software Mapping Options

- Interrupt driven tasks on one core, time driven tasks on other(s)
- Distribute tasks across cores to balance performance
- With and without OS (computationally intensive functionality)
- Split by ASIL level
- Split by tier 1 / OEM
- Platform considerations
- Scalability considerations
Software Mapping Tool

Source: SymtaVision
Software Mapping Tool Example

Source: Timing Architects
TriCore AUO XYZ – Not a Multi-Core

Example earlier generation automotive microcontroller

- Two programmable cores but
  - Very different capabilities and roles in the system
- Analogous to the small DSPs (Oak etc.) in early GSM chips running only the comms stack
- Smaller core very much viewed as an accelerator by the programmer of the main core
  - could be replaced by hardwired engine
  - not sharing the application
### AURIX Family

<table>
<thead>
<tr>
<th></th>
<th>TQFP-80</th>
<th>TQFP-100</th>
<th>LQFP-144</th>
<th>LQFP-176</th>
<th>LFBGA-292</th>
<th>BGA-416</th>
<th>LFBGA-516</th>
</tr>
</thead>
<tbody>
<tr>
<td>9x Series</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>up to 8MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max. SRAM 2.75MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triple-Core</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7x Series</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>up to 4MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max. SRAM 472KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triple-Core</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6x Series</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>up to 2.5MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max. SRAM 752KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dual-Core</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4x Series</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>up to 2MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max. SRAM 96KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single-Core</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3x Series</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>up to 2MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max. SRAM 452KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lockstep-Core</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2x Series</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>up to 1MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max. SRAM 96KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lockstep-Core</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1x Series</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>up to 512KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max. SRAM 56KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lockstep-Core</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- TC222
- TC223
- TC224
- TC233
- TC234
- TC237
- TC24A
- TC264
- TC265
- TC267
- TC275
- TC277
- TC297
- TC298
- TC299
AURIX – HW measures supporting safety

- Redundant, spatial separated peripherals
- Bus Monitoring Unit
- Safe DMA
- Safe SRI
- SRAM ECC (DECTED with enhancements to detect multi bit failures)
- Flash ECC (SECDED with enhancements to detect multi bit failures)
- Lockstep core
- Memory protection core
- Memory protection peripherals
- Safe Interrupt Processing
- Flexible CRC Engine (FCE)
- IO Monitor
- Clock Monitoring
- CPU self tests (90% Latent Fault Metric)
Safety Critical Challenges

- Typically need to show freedom from (non-intentional) interference
  - Spatial Isolation &
  - Temporal Isolation
- Spatial isolation is solvable through protection mechanisms
- Temporal isolation is more difficult when sharing a resource
  - this is typically *implicit* sharing; i.e. an interconnect or a memory controller
Recap

- Hard real time – Meet deadlines
- Safety – Proof your claims

→ Worst Case Execution Time (WCET)

Multi-core specifics:

- Minimize interference
- Upper bound for interference
Design Guidelines for predictable multicore architectures


1. Fully timing compositional architecture (with bounds)
2. Disjoint instruction and data caches
3. Caches with LRU replacement policy (not all cores)
4. A shared bus protocol with bounded access delay
5. Private caches
6. Private memories, or only share lonely resources

Masters seem to share the same bus → “bus is the bottleneck”

Wiring/layout seems to be modest

Wiring seems to be linear increasing with number of participants
Block Diagrams showing Crossbars as Busses are misleading (2/2)

- All masters have a direct connection to all slaves
  - Low latency to a specific slave
  - Only arbitration at the slave between different masters
- Wiring/layout becomes a big issue
- Wiring complexity is increasing with M*S
Debugging with DAP2 + ED: Trace at “JTAG tooling” Price Point

- Just ED (same package) and DAP connector on board
- New: logic analyzer functionality for peripherals and pins
- New: continuous Compact Function Trace (CFT) via DAP2
- Trace tooling on every engineer’s desk

2013-11-15

Copyright © Infineon Technologies AG 2013. All rights reserved.
Observation Points
SRI_CPU1
CPU0
SPB
used for example on next slide
MCDS Trace Example: Unaligned 64 bit write CPU0 to DSPR CPU1

<table>
<thead>
<tr>
<th>TimeR</th>
<th>Ticks</th>
<th>Origin</th>
<th>Data</th>
<th>Operation</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>CPU0</td>
<td>89AB</td>
<td>W 16</td>
<td>600000000</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>SRI_CPU1</td>
<td>89AB</td>
<td>W 16 SV</td>
<td>600000000</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>CPU0</td>
<td>123456780EF0CD00</td>
<td>W 64</td>
<td>600000002</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>CPU0</td>
<td>00000001</td>
<td>W 32</td>
<td>D0009BFC</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>SRI_CPU1</td>
<td>CD00</td>
<td>W 16 SV</td>
<td>600000002</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>CPU0</td>
<td>80208020</td>
<td>W 32</td>
<td>F003D318</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>SRI_CPU1</td>
<td>56780EF0</td>
<td>W 32 SV</td>
<td>600000004</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>CPU0</td>
<td>1234567800000010</td>
<td>W 64</td>
<td>600000002</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>SRI_CPU1</td>
<td>1234</td>
<td>W 16 SV</td>
<td>600000008</td>
</tr>
<tr>
<td>22</td>
<td>5</td>
<td>SPB</td>
<td>80208020</td>
<td>W 32 SV</td>
<td>F003D318</td>
</tr>
<tr>
<td>26</td>
<td>4</td>
<td>SRI_CPU1</td>
<td>0010</td>
<td>W 16 SV</td>
<td>600000002</td>
</tr>
<tr>
<td>30</td>
<td>4</td>
<td>CPU0</td>
<td>00000000</td>
<td>W 32</td>
<td>D0009BFC</td>
</tr>
</tbody>
</table>

- Parallel trace at several observation points
- Very accurate time information with ticks
Multicore, MCDS, ED + DAP: Trace tooling for every engineer everywhere
Do you see the bug?
Nothing has happened (yet)
Outlook Embedded Health

- Complex systems have bugs and diseases
- Some of them will have no symptoms
- Standardized, automated examinations needed
- Automatic highlighting of anomalies
- No hassle and low cost solutions ... enable broad and early usage
Summary

- Requirements for automotive multi-core

- Deadlines – not throughput!

- Software mapping is the challenge

- Success factors: HW Architecture + Tooling +
ENERGY EFFICIENCY
MOBILITY
SECURITY

Innovative semiconductor solutions for energy efficiency, mobility and security.