Software integration challenge
multi-core – experience
from real world projects

Rudolf Grave
17.06.2015

Embedded Multi-Core Conference
EMCC
16.-18.06.2015, Munich, Germany
Agenda

• About EB Automotive
• Motivation
• Constraints for mapping functions to cores
• AUTOSAR & MultiCore & Safety
• Summary & Outlook
**Software integration challenge multi-core**

**EB: Software and Services**

### Infotainment
Connected navigation software
- HMI tools for in-dash, digital instrument clusters and head-up displays
- Global software integration and engineering services

### Connected
- Connected experiences around urbanization and electrification
- Online diagnostics
- Software and content updates

### In-Car Infrastructure
- EB tresos – integrated software and tools, based on AUTOSAR standards
- Solutions for: operating systems, middleware, dependable communication
- Solutions for high integrity systems: reliability, functional safety and security
- Test & simulation

### Driver Assistance
- Software development for driver assistance functions
- Electronic horizon and test drive recording solutions
- Driver assistance algorithms and functions
Agenda

• About EB Automotive
• Motivation
• Constraints for mapping functions to cores
• AUTOSAR & MultiCore & Safety
• Summary & Outlook
Next generation of ECUs

- **Domain Controller** will be huge multi-core systems
  - ECU independent function
  - Connected to Actuator / Sensor ECUs
  - Reloadable functions
  - Connected with Automotive Ethernet

- **Domain ECUs** will be „small“ multi-core or single core systems
  - Hard real time
  - I/O handling
  - Safety functions

- **Motivation for change current E/E architecture**
  - Updatable ECUs
  - Dynamic Systems
  - Security
Motivation

Software integration challenge multi-core

ECU

Function 1

Func 1 CDD

Function 2

F2 Monitor

Function 3 (OEM)

Autosar BSW

Function 4 (Supplier)

Operating System
Software integration challenge multi-core

Agenda

- About EB Automotive
- Motivation
- Constraints for mapping functions to cores
- AUTOSAR & MultiCore & Safety
- Summary & Outlook
„Welcome to the jungle!“ (Herb Sutter)

- Automotive microcontroller follow IT systems with 5-8 years delay
- Appearing microcontroller
  - Different CPUs on one architecture
  - Combination of performance and peripheral cores
  - Combinations with DSP or GPUs
- Special microcontroller for special purpose

Example: Infineon AURIX TC27x

Two lockstep cores
- 1.6E
- 1.6P

One non-lockstep core
- 1.6P

Source: http://www.infineon.com/export/sites/default/media/products/Microcontrollers/32bit/BlockDiagram-TC27xT.png
Example: Infineon AURIX TC29x

One lockstep cores
- 1.6P

Two non-lockstep core
- 1.6P

Source: http://www.infineon.com/export/sites/default/media/products/Microcontrollers/32bit/TC29xT_Block_Diagram.JPG
Example: Freescale MPC5777M

Two computational core
- e200z7 (lockstep)
- e200z7

One peripheral core
- e200z4
Constraint: Safety Goals

Safety Goals

Functional Safety Requirements

Functional Requirements

Hardware

Technical Safety Requirements

Technical Requirements

Software

Req1: Freedom from Interference

Req2: OEM BB software shall run on lockstep core

Req3: Performance
Software integration challenge multi-core

Freedom from interference (ISO 26262)

In software partitioning system the freedom from interference argument between software component shall claim that:
• Shared resources are only used by one partition or a appropriate mechanisms are implemented
• Software partitioning is supported by hardware mechanism e.g. MPU
• Software mechanism ensuring FFI are implemented on highest level

Spatial FFI
“The data used by a one element shall not be changed by a another element. In particular, it shall not be changed by a non-safety related element.”

Temporal FFI
“One element shall not cause another element to function incorrectly by taking too high a share of the available processor execution time, or by blocking execution of the other element by locking a shared resource”

Exchange of information
“Exchange of information shall be defined as transport of data between software elements or hardware. (e.g. messages between ECUs or cores, data from non-volatile memory)”
Software integration challenge multi-core

Functional Constraints

- AUTOSAR basic software < v. 4.0.3 can’t be distributed on cores
- Reuse of legacy software
- Functional requirements
  - Response times
  - CPU load requirements
  - Startup times
  - Delay in Inter-Core, Inter-processor communication
- Access to peripherals
First idea: Group according to supplier

Core0 - lockstep required
- Function 1
  - CDD
- Function 2
  - Monitor

No lockstep core required
- Func 1
- F2

Function 3 (OEM)

Lockstep core required

High amount of communication with AUTOSAR BSW

High amount of communication with Function 2

No lockstep core required

Function 4 (Supplier)

Autosar BSW
Next idea

Core0 - lockstep
- Function 3 (OEM)
- Function 2
- F2 Monitor
- Function 4 (Supplier)
- Autosar BSW
- CPU Load > 100 %

Core1 - nonlockstep
- Function 1
- Func 1 CDD
- CPU Load = 25 %

Core2 - nonlockstep
- CPU Load = 0 %
Agenda

• About EB Automotive
• Motivation
• Constraints for mapping functions to cores
• AUTOSAR & MultiCore & Safety
• Summary & Outlook
AUTOSAR 4.0 – MultiCore

- AUTOSAR BSW runs on one core
- Safety OS MultiCore and Rte provides the ability to distribute AUTOSAR Software Components (SWC) to different cores
Singlecore Safety architecture

EB tresos Safety products enable
- mix of QM and Safety Software (also different ASIL level)
- integration of Black-Box Software (independent from ASIL-Level)
Os Applications

OS application
• Group of OS elements (ISRs, tasks, events, alarms,...)

Safety OS
• OS Applications are used to realize memory partition

MultiCore
• OS Applications are used for
  – Task to core mapping
  – ISR to core mapping
Multicore Safety architecture

Core0 - lockstep
- Function 3 (OEM)
- Function 2
- F2 Monitor
- RTE
- EB tresos TimE

Core1 - nonlockstep
- Function 4 (Supplier)
- RTE
- RTE
- RTE
- Autosar BSW

Core2 - nonlockstep
- Function 1
- RTE
- Func 1 CDD

EB tresos Safety OS Multicore
Agenda

• About EB Automotive
• Motivation
• Constraints for mapping functions to cores
• AUTOSAR & MultiCore & Safety
• Summary & Outlook
Summary: Software Architecture and early verification is the key to success

- Connect experts from functional development and infrastructure (Operating system and RTE)
- Functional expert can concentrate on algorithm
- Integration expert can concentrate on OS, RTE, Safety and performance
Outlook – “Reading the Crystal Ball”

• Multi-core systems are becoming a reality now and allow further consolidation of more functions on fewer ECUs

• Software system integration capabilities are key to success:
  – Taming the rising complexity
  – Building dependable, high-integrity systems

• Dynamic (service-oriented), dependable and high-integrity system architectures are required for future vehicles, e.g. autonomous driving. @EB you can evaluate / order such systems today (or soon) 😊

• Mixed operating system will come
Contact us!

automotive.elektrobit.com
Rudolf.Grave@elektrobit.com