Safety-Critical Heterogenous Multicore Technologies

Prof. Dr.-Ing. Dr. h. c. Jürgen Becker
Karlsruhe Institute of Technology – KIT
Why Multi-Core is Roadmap …

- Until ~2005 Processor Performance increase driven by:
  - Clock Speed
  - Execution Optimization
  - Cache & ILP

- Power Wall
- ILP Wall

→ Led to heterogenous Multi-Core Processors
→ Parallelism has to be exposed by Programmer

(source http://www.gotw.ca/publications/concurrency-ddj.htm)
Microelectronics: Various Processor Solutions

**IBM Cell Processor** (PS3)

**Motorola M68355 Microcontroller**

**SoC: Apple A7** (iPhone 5s)

**TI DSP 8847**

**System-on-Chip**
(Bluetooth SoC: Alcatel)
Trend in Processor Architectures

Microarchitecture Trend

- Multi-Core, Multi-Threaded
- Future Xeon™ Architecture
- Speculative Out of Order
- Pentium® 4 Architecture
- Pentium® Pro Architecture
- Pentium® Architecture
- Super Scalar
- Instruction Parallelism
- Era of Thread Parallelism
- Era of Instruction Parallelism
- Era of Trace Cache
- Pentium® 4 Architecture
- Pentium® Pro Architecture
- Pentium® Architecture
- Super Scalar

Same holds true for embedded processors!

→ Microarchitecture Trend provides solutions for automotive needs
Processor Breakthroughs ....

A major architecture disruption: multiprocessing and specialization will have a strong impact on software.

Technology limitations: perf. by parallelism no more by frequency => disruption in programming model, long term research challenges.

Multi-processing

Processor specialization

Moore’s law

Source: G. Edelin (Thales), 2009

Intel 8086

PowerQUICC II

Power

Time

1980
1990
2000
2015-2020 (?)

CISC era

RISC era

Architectural break point

Domain oriented architectures: eg with predictable performance to control the timeliness in RT critical applis, dynamic reconfiguration for adaptive, distributed critical architectures (multilevel RT composability)
Processors in Embedded Systems

- 15 Billion Systems in 2015
  - Ubiquitous
  - Networked
  - Additional Features necessary!
    - Autonomous Systems, etc.

- Innovative Application Models, Products & Services
  - Mobility (Automotive, Avionics, …)
  - Industrial & Medical Automation
  - Communication
  - Entertainment/Infotainment

Reliable and High-Performance Electronics
Automotive Security Systems ....

are more & more important!
CPS: Connected ECUs in Automotive

More than 80 ECUs in Upper Class Cars

More than 35% of Added Value of Passenger Car for Electrical/Electronic Systems

90% of all Innovations are based on Electronics

Software Part is quickly increasing

With Permission of DaimlerChrysler AG
Car-X-Communication: Motivation & Goals

-> especially in E-Mobility Scenarios, due to limited energy and range situations

- Higher vehicle safety
- Traffic management & optimization
- Infotainment & Value-added services

Vehicular Ad Hoc NETwork (VANET)
Latency requirements
Throughput and performance
Situation adaptiveness

Source: COMeSafety project
How to model processing and communication of heterogeneous automotive embedded systems including C2X communication systems?
Demonstrator Car-to-X Communication

Carmaker (visualization)

Simulation (ECU 3 .. N)

Scenarios:
- site warning
- broke down vehicle
- traffic jam

Vehicle (ECU1)
Car-X-Communication: Complexity

- Distributed functions
- Increasing communication
- C2X as opening of E/E architecture
- Secure communication in real-time (2500 msg/s)

Number of control units within a vehicle

<table>
<thead>
<tr>
<th>Year</th>
<th>Busses</th>
<th>Subbusses</th>
<th>ECUs</th>
<th>Sub-ECUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1986</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1995</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Source: Daimler)
Automotive Collaboration Platform
Overview of Car2X Multicore ECU

- Partitioning in Intra- and Inter-Car
  - Results in non-secure and secure domain
  - No access to secure domain from outside world
- Hardware and software have to be safe and secure
Scientific Development: Solution Concept:

-> Multi-Domain Design Methodology
Cyber-physical System: Vision Smart Mobility

Relevant information of current situations will be used in real-time by all traffic participants for a safe, efficient, environmentally friendly and comfortable mobility.

Examples:
- Green Mobility\[^{1}\]
- Accident-free driving\[^{1}\]
- Support for elderly people \[^{1}\]
- Adaptive route planning to the point of 4D harmonization \[^{2}\]

The requirements of **realtime information processing** capability and control of individual traffic participants increase considerably.

This vision is directly and technological inevitably linked to the realizable processing power of the individual mobile subsystems.

---

\[^{1}\] ZVEI - Nationale Roadmap Embedded Systems, 4.2 Mobilität, Mobilitätsszenarien 1-3
\[^{2}\] SESAR – Single European Sky ATM Research
Increase of Electronic Systems (HW + SW) is required

- to master complexity
- to meet environmental challenges
- to enhance competitiveness
- to improve cost efficiency

Degree of automation will directly depend on embedded computing power!
Project Focus of ARAMiS

Focus of the project:
Appropriate deployment of multicore systems and virtualization in the mobility domains automotive, avionics and railway especially for safety critical applications.

Challenges:
Deployment of multicore systems and virtualization for safety critical applications in mobility domains

Major Topics:
- Real-time,
- Performance,
- Reliability and availability,
- Safety,
- Security,
- Compatibility to existing concepts,
- Energy efficiency
Key Enabling Technology: Multicore

- Singlecore will not provide enough computing power in the future (scaling is over)
- Multicore is the best known solution that is able to provide sufficient performance

Future System Aspects and Characteristics

- Functionality is safety critical in many cases
- Systems are highly connected in networks
- Evolutionary Development (Legacy Code)
- User friendliness and smart ecosystems: Use of APPS
- Cyber Physical Systems with mixed criticality and dynamic reconfiguration of structures and services
Structuring the Multicore Challenges

- True parallelism
- Highly configurable
- Dynamic features
- Shared resources

Synchronization

Correctness

Complexity - management

Determinism

Segregation

Mixed criticality
COTS Multi-Core Processors

- Two or more processing cores
  & one central interconnect
- Shared peripherals & memory
  and common supply & clock

References:
Freescale Qorivva MPC5746M MCU Fact Sheet
Infineon Tricore AURIX TC27xx Product Brochure
Freescale QoriQ P4080 Reference Manual
Freescale i.MX 6Quad Reference Manual
Multicore in Safety-Related Systems

- **Synchronization**
  - Caused by true parallelism

- **Segregation**
  - Caused by shared resources

- **Complexity Management**
  - Highly configurable
  - Dynamic features
Starting Point: Scenarios & Requirements

AgendaCPS, SPES2020, BICC\textsuperscript{NET}, SESAR, ...

Society Context

Cross-Domain CPS/Smart Mobility Scenario
E1.1.4.X

Connected

Specialization

TP1

Automotive Scenarios
E1.1.1.X

Avionics Scenarios
E1.2.1.X

Railway Scenarios
E1.3.1.X

Requirements

TP2, TP6 (incl. TP3, TP4, TP5)
ARAMiS Demonstrators

- Safety
- Security
- Situation Awareness
- Cabin-Management
- DS34Rail
- Large Scale Software Integration
- Virtualized Car Telematics
- Deterministic Systems
- Open Systems

Avionics
Railway
Automotive
Rising Complexity Due To More ECUs

► More Software requires more computing power
► More computing power ...... means what?

Efficient control of increasing functionality while keeping the number of ECUs constant or even reducing it!
Multicore Challenge Virtualization

- **Virtualization** as key technology for usage of multicore platforms in embedded systems
  - **Consolidation** of functions
  - **Segregation** of applications with different criticalities/requirements regarding functional safety
  - **Integration/re-use** of existing software

- **Basic principles**
  - Sharing and segregation of computation
  - Sharing and segregation of memory
  - Secure communication between partitions

**Virtual machines for**
- different applications
- with different criticalities

**Monitoring to supervise and organize resource sharing**

**Server partition for shared resources**

**Dedicated interfaces for different VMs**

**System overview**

- **Hypervisor**
  - **Monitor**
  - **Runtime Environment**
  - **Multicore Hardware Platform**
  - **Applications**
    - Trusted VM: GENIVI
    - Untrusted VM: Android
    - Early & Safe VM: RTOS
    - Server VM: Linux
  - **Core**
  - **Memory**
  - **Coprocessor**
  - **Communication**
Demonstrator Platforms: LSSI Integration

BMW platform
- Integrating powertrain components
- Non-intrusive approach
- Using Hypervisor
- Using AUTOSAR stacks inside partitions/VMs

Audi platform
- Integrating chassis components
- Intrusive approach
- Using monolithic AUTOSAR stack and native methods for integration
Virtualization

Levels of Abstraction and/or Virtualization

→ There is always effort somewhere - No solution for free!
Deployment Approach:  
-> Constructive & Analytic Part:

Separation of Concerns, Modeling and Extensive Automation

► Reduced complexity of individual engineering subtasks
► Better work division for teams of domain experts
► Increased reusability by low coupling and high cohesion
► Flexible re-deployment with tool support (generators, analyzers, schedulers, …)
Impact of Microcontroller Architecture

Specific architectural properties of microcontroller products gain importance regarding efficient implementation of parallelization and deployment.

- Different computing power per core may need attention.
- Other properties like safety capabilities may also have to be taken into account for deployment decisions.
- Cores may have dedicated resources as well as access to global resources with different access speed.
Conclusions & Outlook

**Exceptional Complexity:**

**Hardware/Software Platforms of Multicore Systems**

- **Two essential Goals in Multi-Core Context of Safety-Critical Systems**
  - Efficient utilization of **Parallelism & Performance**
  - Preservation and enforcing of **Predictability & Determinism**

- **Detailed analysis with modeling and tooling necessary**
  - **Understanding**, **Classifying** and **Separating** of issues
  - **Simulation** and **Evaluation** of design parameters and decisions, selection of suitable implementation strategies

- **Design and implementation of concepts as extension to existing hardware/software architectures**
  - **Software:** Operating System, Middleware, Virtualization Layer
  - **Hardware:** On-Chip Communication, Sharing of I/O & Co-Processors with Scheduling & Bandwidth

Derivation and evaluation of requirements for future hard- and software architectures
Multi-Core Technologies ...

Moore’s Law:
continues, ...
But: clock rates AND
cost/transistor does
not scale any longer!

Multicore Solutions:
homogenous/heterogenous

Reliability &
Certification

Performance &
Domain Orientation

Frequency &
Power Wall

Memory Wall

2005

ALL Layers:

System
(micro) architecture
Logic
Device

2015
ALMA - ALgorithm parallelization for Multicore Architectures

Motivation:

- Multiprocessor System-on-Chip (SoC) become more and more common
- Programming them suffers from a complex tool-chain and programming process

Project Goals:

- New Tool Chain for efficient mapping of applications on multiprocessor platforms from high level of abstraction
- Use of high level programming language Scilab
- Hide the Complexity of the underlying multi-core architecture to the end user

Coordination:
Prof. Dr.-Ing. Dr.h.c. Jürgen Becker

Funded by:
European Union

ITIV Spin-off from ALMA:
emmtrix Technologies
www.emmtrix.com

30.10.15
Funded by: European Union

ITIV Spin-off from ALMA: emmtrix Technologies www.emmtrix.com
“Leveraging Embedded Multicore Computing”
performance • portability • cost saving • seamless integration
Multicore Parallelization is today mainly manually realized and is very time and cost intensive

⇒ High Development Cost (△ Resource Effort △ Cost)
⇒ Long „Time-to-Market“
⇒ Innovations are blocked!
The new emmtrix tool will address actual problems in Multicore Programming and Parallelization and will open new Innovations!

- Shorter Development Time ($\triangle$ Ressource Effort $\triangle$ Cost)
- Shorter "Time-to-Market"
- New Innovations will be enabled!
Thank You!
Further Information

Contact:
Prof. Dr.-Ing. Dr. h. c. Jürgen Becker
Karlsruhe Institute of Technology (KIT)
Institute for Information Processing Technologies (ITIV)
Engesserstr. 5
76131 Karlsruhe
Germany
Tel.: +49 721 - 608 – 42502
Email: becker@kit.edu