IEC-61508 certification of mixed-criticality systems based on multicore and partitioning

Ada Europe 2015
Madrid (23rd June)

Jon Perez
jmperez@ikerlan.es
01 Context

02 Multicore is what you need / what you will have

03 The business need and opportunity

04 The wind turbine example

05 Conclusions and lessons learnt
01 Context
Some Research Projects: Multicore & mixed-criticality
Keynote in a nutshell

**Market Pull**
- Reliable
- Scalable
- Available

**Technology Push**
- CPU
- CPU
- CPU
- CPU

**Product H2020**
- Wind Turbine
- Train
- Electric Car
“Modern electronic systems used in industry (avionics, automotive, etc.) combine applications with different security, safety, and real-time requirements. Systems with such mixed requirements are often referred to as mixed-criticality systems”.

[Baumann, 2011]

“The integration of applications of different criticality (safety, security, real-time and non-real time) in a single embedded system is referred as mixed-criticality system”.

[Perez, 2014]
Multicore is what you need...
Multicore is what you will have...
2\textsuperscript{nd} International Conference Automotive Embedded Multi-Core Systems.

Roadmaps:

**32-bit Body Electronics MCU Roadmap**

- **Fado/Bolero 90nm Products**
  - MPC5668/6/7/E
    - 2\textsuperscript{6}x20, 110MHz,
      Up to 3M Flash, 598k RAM
    - Flex, Ether, MLB, CAN, LIN
  - MPC5646/5/4C
    - 2\textsuperscript{4}x20, 120MHz,
      Up to 3M Flash, 259k RAM,
    - Flex, Ether, Security,
      CAN, LIN
- **55 nm Next Gen Products**
  - MPC5646/7/0G
    - 2\textsuperscript{4}x20, 120MHz,
      3M SM Flash,
    - Flex, Ether, Security,
      MLB, USB, CAN, LIN

**Integrated Gateways**

- MPC5668E
  - 2\textsuperscript{6}x30, 110MHz,
    Up to 2M Flash, 598k RAM
  - Flex, Ether, MLB, CAN, LIN

**Mid-High BCM**

- MPC5607/6/5/3B
  - 2\textsuperscript{6}, 64MHz,
    Up to 1.5M Flash, 96k RAM
  - CAN, LIN

**Entry BCM**

- MPC5604/3/2/1B/C
  - 2\textsuperscript{6}, 64MHz,
    Up to 512k Flash, 49k RAM
  - CAN, LIN

- MPC5602/1D
  - 2\textsuperscript{6}, 49MHz,
    Up to 256k Flash, 16k RAM
  - CAN, LIN

**Source:** [www.freescale.com](http://www.freescale.com)
Generic purpose multicore

ZYNQ.
UltraSCALE+
Zynq UltraScale+™ MPSoC

Quad-Core ARM® Cortex®-A53
Dual-Core ARM® Cortex-R5
ARM® Mali-400MP
H.265 Video Codec Unit

Smarter Network
- Bandwidth optimized
- High-end processors with offload

Smarter Vision
- H.265 Video Codec
- 2D and 3D graphics

Smarter Control
- Lowest power and cost
- Small density and footprint

Smarter Vision & Network
- Kintex-based Fabric
- Bandwidth optimized

Smarter Control & Vision
- Artix-based Fabric
- Lowest power, cost and density

Source: www.xilinx.com
IEC-61508-3 Annex F (Informative) – “Techniques for achieving non-interference between software elements on a single computer”

- Independence of execution should be achieved and demonstrated both in the spatial and temporal domains.
  - Spatial: 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: 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 of some kind

- The term “independence of execution” means that elements will not adversely interfere with each other’s execution behaviour such that a dangerous failure would occur.”
Threats to be considered and managed
Threats to be considered and managed

- **Temporal & Spatial independence**, e.g., Shared resources (e.g., memory, cache, bus, interrupts) [1]

Which is the time-scale of the temporal interference?

ps ns msec second


Threats to be considered and managed

- Complex (new) hardware components, e.g., Core interconnect fabric
- Lack of detailed documentation


Threats to be considered and managed

- Worst Case Execution Time (WCET)
Threats to be considered and managed

- Interference among safety related and non safety related functions, e.g.,
  - Safe configuration
  - Safe startup and boot
  - Safe shutdown
  - Exclusive access to peripherals
  - Resource virtualization

- Diagnosis

The need and opportunity
Impact Perspective – The right scale

[Diagram showing the growth in rotor diameter over time, with specific sizes marked: 15 m Ø in '85, 112 m Ø in '93, 126 m Ø in '97, 160 m Ø in '01, and 250 m Ø in '05. The diagram also includes a note for the Airbus A380 wing span of 80 m and a question mark for future growth.]

A modern off-shore wind turbine dependable control system manages [1,2]:

- **I/Os**: up to three thousand inputs / outputs.
- **Function & Nodes**: several hundreds of functions distributed over several hundred of nodes.
- **Distributed**: grouped into eight subsystems interconnected with a fieldbus.
- **Software**: several hundred thousand lines of code.


Source: [www.almstrom.com](http://www.almstrom.com)
Automotive domain:

- The software component in high-end cars currently totals around 20 million lines of code, deployed on as many as 70 ECUs [1].
- Automotive electronics accounts for some 30% of overall production costs and is rising steadily [1].
- A premium car implements about 270 functions that a user interacts with, deployed over 67 independent embedded platforms, amounting to about 65 megabytes of binary code [2].

(On-board) railway domain:

- The ever increasing request for safety, better performance, energy efficient, environmentally friendly and cost reduction in modern railway trains have forced the introduction of sophisticated dependable embedded systems [1].
- The number of ECUs (Electric Control Units) within a train system is of the order of a few hundred [2,3].
- Groups of distributed embedded systems:
  - Train Control Unit.
  - Railway Signalling (e.g. ETCS).
  - Traction Control.
  - Brake Control.
  - Etc.

The wind turbine example


Perez, J. and A. Trapman (2013). Deliverable D7.2 (Annex) - Wind power case-study safety concept, FP7 MultiPARTES.
Introduction – Context Diagram

- Safety
- Non Safety Related

- HMI & COMS
- Supervision

- Safety Protection
- Safety Relay

- Speed Sensor (s)

- Sensor (s)
- Actuators
- Subsystems

< Safety Chain >

Output relay pitch control

ETHERCAT
Introduction – Proposed solution

Safety Relay

Safety Protection

Supervision

HMI & COMS

Output relay pitch control

< Safety Chain >

Speed Sensor (s)

Sensor (s)

Activators

Subsystems

ETHERCAT
### Safety Concept - Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR_WT_4</td>
<td>The &lt;Protection System&gt; safety function must activate the “safe state” if the “rotation speed” exceeds the “maximum rotation speed”</td>
</tr>
<tr>
<td>SR_WT_5</td>
<td>The &lt;Protection System&gt; safety function must ensure “safe state” during system initialization (prior to the running state where rotation speeds are compared)</td>
</tr>
<tr>
<td>SR_WT_6</td>
<td>&lt;Protection System&gt; safety function must be provided with a SIL3 integrity level (IEC-61508).</td>
</tr>
<tr>
<td>SR_WT_7</td>
<td>The safe state is the de-energization of output “safety relay(s)”</td>
</tr>
<tr>
<td>SR_WT_8</td>
<td>Output “safety relay(s)” is(/are) connected in serial within the safety chain.</td>
</tr>
<tr>
<td>SR_WT_9</td>
<td>A single fault does not lead to the loss of the safety function: HFT=1 and Diagnostic Coverage (DC) of the system &gt;= 90% (according to IEC-61508).</td>
</tr>
<tr>
<td>SR_WT_10</td>
<td>The reaction time must not exceed PST (SW_WT_14)</td>
</tr>
<tr>
<td>SR_WT_11</td>
<td>Detected ‘severe errors’ lead to a “safe state” in less than PST (SW_WT_14)</td>
</tr>
<tr>
<td>SR_WT_12</td>
<td>The “rotation speed” absolute measurement error must be equal or below 1 rpm to be used by &lt;Protection System&gt;. If measurement error ≥ 1 rpm it must be neglected</td>
</tr>
<tr>
<td>SR_WT_13</td>
<td>The “Maximum Rotation Speed” must be configurable only during start-up (not running)</td>
</tr>
<tr>
<td>SR_WT_14</td>
<td>The Process Safety Time (PST) is 2 seconds</td>
</tr>
</tbody>
</table>
Safety Concept – The approach

◊ Safety concept based on ‘common practice in industry’

◊ Serves as a reference, not detailed

◊ Analogous safety concept using heterogeneous multicore and hypervisor

◊ The MultiPARTES contribution
DUAL-PROCESSOR – 1oo2

Safety techniques (IEC-61508 SIL3):
- 1oo2
- HFT=1 and DC >= 90 %
- Dual diverse sensors
- Dual independent safety relays connected in serial
- Dual Diverse Processors:
  - ‘P0’ safety functions only
  - ‘P1’ mixed functionalities
  - ‘P0/P1’ independent safety relay
  - Local diagnosis and reciprocal comparison by software (‘P0/P1’)
- Communication: EtherCAT and ‘safety over EtherCAT’
DUAL-PROCESSOR – 1oo2

Scalability limitations:

- The number of functionalities continues to increase (real-time, safety and non-safety)
- Usage of fan not allowed (reliability issue)
- ‘P1’ Processor performance capability reaches a limit...
Increased Scalability:
• Add additional processors (P2, P3, etc.) to provide required computation performance

Reduced Reliability:
• The overall system reliability and availability is reduced...
PARTITIONED

Safety Concept – (B - ‘Multicore partitioning’)

Is it feasible to developed a ‘partitioned’ solution?:

- Usage of a certifiable hypervisor.
- System partitioning (safety, real-time and non real-time partitions).
- Interference freeness of non-safety partition with safety partitions, and lower criticality levels with higher criticality levels.
‘Partitions’ mapped to a multicore processor:
- Heterogeneous quad core.
- Dual diverse cores for safety partitions.
- Partitioning and multicore allocation enables resource usage and performance maximization while ensuring interference freeness.
Safety Concept – (B - ‘Multicore partitioning’)
Safety Concept – (B - ‘Multicore partitioning’)

◊ Scheduling (IEC-61508-3 Annex E):
  • Static cyclic scheduling algorithm.
  • Pre-assigned guaranteed time slots.
  • Defined at design time.
  • Synchronized based on the global notion of time.

◊ Diagnosis:
  • The partition should be self contained and should provide safety life-cycle related techniques and platform independent diagnosis abstracted from the details of the underlying platform.
  • The hardware provides autonomous diagnosis and diagnosis components to be commanded by software.
  • The hypervisor and associated diagnosis partitions should support platform related diagnosis.
  • The system architect specifies and integrates additional diagnosis partitions required to develop a safe product taking into consideration all safety manuals.

Conclusions and lessons learnt
Conclusions and lessons learnt

◊ It is feasible to achieve SIL3 IEC-61508 / Pld ISO-13849 with COTS multicore, partitioning and current safety standard versions.

◊ Temporal independence and isolation:
  • Temporal isolation simplifies the safety argumentation but... Temporal independence does not necessarily require temporal isolation.
  • The lack of complete temporal isolation and rare (undocumented) temporal events could reduce the availability of the system but should not jeopardize safety (fault avoidance and control).

◊ The same strategy can be extended to different domains with safety standards that use IEC-61508 as reference standard.
  √ Wind Turbine, IEC-61508 SIL3 and ISO-13849 Pld
  √ Railway signaling, SIL4 EN-5012X using PTA (Probabilistic Time Analysis)
  ◊ Working with automotive domain case study ASILC ISO-26262