TÜV SÜD Rail

Multicore Devices in Safety Applications - Normative Aspects
Workshop Overview

- Definition
- System architectures
- Normative aspects
- Avoidance of systematic failures
- Fault Detection
A multi-core processor is a single computing component with two or more independent actual processing units (called "cores"), which are the units that read and execute program instructions. Source: wikipedia

Multi-core technology refers to CPUs that contain two or more processing cores. These cores operate as separate processors within a single chip. By using multiple cores, processor manufacturers can increase the performance of a CPU without raising the processor clock speed. Source: http://techterms.com/definition/multi-core

A multi-core processor is an integrated circuit (IC) to which two or more processors have been attached for enhanced performance, reduced power consumption, and more efficient simultaneous processing of multiple tasks. Source: http://searchdatacenter.techtarget.com/definition/multi-core-processor
Single Core versus Multi Core

- Core
- Cache
- Power Management
- Clock Management
- Interconnect (Crossbar / Bus / Network)
- Memory Controller
- Peripheral Controller
- I/O₁
- I/Oₙ
Single Core versus Multi Core

- Core 1
  - Cache
- Core n
  - Cache
- Power Management
- Clock Management
- Shared Cache
- Interconnect (Crossbar / Bus / Network)
- Memory Controller
- Peripheral Controller
- I/O
  - I/O_1
  - I/O_n
Managing Shared Resources

Source: http://cacm.acm.org/magazines/2010/2/69360-managing-contention-for-shared-resources-on-multicore-processors/fulltext
Different Architectures at Multicore Level
Different architectures in different industries - Automation

2 Channels (Multicore only in one channel)
Once channel with supervision (watchdog)
Different architectures in different industries - Avionic

Redundant system architecture (high availability)
Normative Aspects

- Safety Lifecycle, avoidance of systematic failures
- Failure modes
- Design related requirements, control of failures
- Verification
Avoidance of systematic failures

• According to IEC 61508-2:2010 no specific requirements for mass-produced electronic integrated circuits are defined.

• It is assumed that the avoidance of systematic failures is ensured by:

  1. Stringent development procedures
  2. Rigorous Testing
  3. Significant feedback from users

• If this assumption is not applicable the requirements for ASICs (see 7.4.6.7 and informative Annex F) will apply

See IEC 61508-2:2010 7.4.6.1 Note
Avoidance of systematic failures

• ISO 26262-10.2011 Annex A contains examples about how to deal with microcontrollers in the context of automotive applications

• Two example approaches to provide evidence related to avoidance of systematic failures during design of a microcontroller are shown:

1. Use measures defined in table A.8 of ISO 26262-10:2011
2. Providing rationale by field data of similar products
   See ISO 26262-10:2011 chapter A.3.7

• ISO/PDPAS 19451-2 : Application of hardware qualification
ASIC Safety Lifecycle

Design entry

1. Structured description
4. (V)HDL simulation
14. Observation of coding guidelines

Synthesis

20b. Static analysis of the propagation delay
24. Application of proven in use tools

Placement, Routing and layout generation

34. Design rule check (DRC)
35. Verification of layout versus schematic (LVS)

ASIC verification

Fault insertion testing (DC ≥ 90 %)

Test insertion and test pattern generation

27. Implementation of test structures
29. Simulation of the gate netlist

Chip Manufacturing

44. Test coverage
47. Burn-in test
Failure Modes

IEC 61508-2:2010 Annex A:
- DC fault model
- Change of information caused by soft-errors
- Dynamic cross-over for memory cells
- No definite failure assumption
- Common Cause faults

ISO 26262-5:2011 Annex D
- D.C. fault model
- Soft error model
- Wrong coding, wrong or no execution
- Dependent faults

ISO 13849-2:2012 Annex D
- Undetected faults
Common cause failure, dependent failure

- **ISO 26262-1, 1.22: dependent failures**
  - “Failures whose probability of simultaneous or successive occurrence cannot be expressed as the simple product of the unconditional probabilities of each of them”
    - Note: “Dependent failures include common cause failures and cascading failures“
  - **1.13: Cascading failure**: failure of one item causes other elements to fail
  - **1.14: Common cause failure**: failure of two or more elements resulting from a single event or root cause

- **IEC 61508-4**,
  - **3.6.9: dependent failure**: „failure whose probability cannot be expressed as the simple product of the unconditional probabilities of the individual events that caused it”
  - **3.6.10: common cause failure**: „failure, that is the result of one or more events, causing concurrent failures of two or more separate channels in a multiple channel system, leading to system failure”
  - **IEC 61508-6**: Common cause failure considerations (“β-factor”) include dependent failures
Design related requirements


- Annex E - requirements for integrated circuits (ICs) with on-chip redundancy e.g.:
  -> Separate physical blocks on substratum of the IC shall be established for each channel and each monitoring element
  -> The minimum distance between boundaries of separate physical blocks shall be sufficient
  -> The common cause potential of common resources such as boundary scan circuitries shall be analyzed.
  -> The estimated $\beta_{IC}$ shall not exceed 25%.

- ISO 26262 - ISO/PDPAS 19451-1: Application of concepts
  -> chapter 7 Multi-core components and ISO 26262
  -> Physical separation by using e.g. guard rings, separate wells,
Verification of diagnostics

IEC 61508-2:2010:

- Fault insertion testing (DC ≥ 90 %)

ISO 26262-5:2011

- Fault injection testing for ASIL C/D

Verification of diagnostics on IC level is not yet fully defined in the standards.
How can we help you?

www.tuv-sud.com/rail
rail@tuv-sud.com