A flexible microcontroller architecture for fail-safe and fail-operational systems

Riccardo Mariani (YOGITECH SpA)
Thomas Kuschel (Toshiba Electronics Europe, Germany)
Hiroshi Shigehara (Toshiba Corporation Semiconductor Company, Japan)
Functional safety

- Safety is the freedom from unacceptable risk of physical injury or of damage to the health of people
  - Focus of functional safety is on system or equipments
    - For example, a microcontroller for braking systems

- International norms such as IEC 61508 and ISO 26262 are ruling functional safety
  - The goal of those norms is to reduce the risk from an unacceptable to an acceptable level

Measures to be adopted to guarantee functional safety of an electronic system are rated according its target safety integrity level (for automotive, the highest is SIL3 or ASILD)
Safety vs Availability

- Availability is the probability that the system will be functioning correctly at any given time.

- Safety and availability have different targets and could (but shouldn’t) be in contradiction:
  - “The safest train is the one that never starts”
  - The increase of transient faults (e.g. soft-errors) is a key concern for availability: most of modern systems would go to reset state to recover from a transient faults….. They are safe but not available!

- Safety and availability must be maximized in modern electronic equipments.
A typical electronic control unit

- **Aim**: Reduce costs to achieve functional safety and availability in accordance with IEC 61508 and ISO 26262
- **Solution**: Use of a SIL3/ASILD microcontroller based on a distributed and optimized tightly coupled redundancy
A distributed architecture

- centralized vs distributed architecture
  - a “centralized” safety architecture has a limited view about local failures and therefore it can just control input and outputs.
    - It has slow reactions to failures and poor knowledge of root causes.
  - a “distributed” safety architecture detects and controls failures locally, i.e. it follows the data process exactly where it occurs.
    - It has fast reactions to failures and wide knowledge of root causes.

Therefore, it can guarantee the correct behavior of a function better than a centralized architecture: in other words, it is more reliable.
Our goal

• Our goal is the implementation of a distributed & diverse architecture....

• ...but how to do it?
  – Each unit shall be monitored by an HW local supervisor
    • For complex units such as CPU, a deeper local supervision is achieved by
      dividing the supervisor itself in a set of modules with independent checkers
    • The supervisor shall detect a fault before the unit is compromised and it shall
      deliver a detailed fault information as fast as possible. To quickly dispatch
      fault information, the supervisors shall be connected in network
  – Each supervisor shall be diverse respect the supervised unit
    • To decrease common-cause and systematic failures

• How to specify these supervisors?
  – Each unit shall be analyzed in detail, till the gate level. To do that,
    a methodology is needed.
fRMethdoology principles

- fRMethdoology main principle:
  - The integrated circuit is partitioned in basic elements ("sensitive zones")
  - The risk of failure of each "sensitive zone" is computed as in FMEA theory:

\[
\lambda = f(\lambda_{\text{elem}}, C, D, F, DC)
\]

- The needs for corrective actions inside each unit (e.g. CPU) are identified and ranked: the correspondent distributed supervisor (e.g. fRCPU) is designed and optimized accordingly.
“faultRobust“ supervisors

• The basic architecture of a supervisor is:

  – An interface collects signals from the supervised unit
  – A set of “predictors” are estimating the expected behavior
  – A set of independent “checkers” are comparing the results
  – An error logger collects fault information and generates the alarms
Supervisors used in Test Chip

For peripherals (e.g. timers, DMA)

Optimized Tightly Coupled (OTC) Dual Core

For ARM CortexM3 CPU

For peripherals (e.g. timers, DMA)

Bus

MCU sub-systems

- CPUs
- memories
- busses
- peripherals
- diagnostic infos

To collect error information

For ARM AHB Bus

fRBUS

fRCPU

fRMEM

Memory

Peripheral

Peripheral

Bus

fRNET

To collect error information

Yogitech HW Diagnostic circuit

Toshiba HW Diagnostic circuit

2nd HiPEAC Workshop on Design for Reliability (DFR’10)
Gate count comparison

- The resulting cost of diagnostic for Optimized Tightly Coupled (OTC) fault supervisor is only about 24% of the cost of diagnostic for a dual-core lock-step.
Conclusions

• Achieving functional safety with a reasonable cost and without compromising availability is a key concern of many electronic equipments (automotive, medical, etc)

• Use of massive HW redundancy is too cost expensive while use of SW redundancy is not enough to detect faults such as transient faults (soft-errors)

• The use of a distributed, optimized and tightly coupled redundancy is an effective solution....