June, 2010

Addressing Safety Standard Requirements for IEC61508 (SIL3) and ISO26262 (ASIL-D) with the MPC5643L 32-bit Power Architecture® Microcontroller

FTF-AUT-F0729

Markus Baumeister
Automotive System Engineer, Functional Safety
Introduction

Functional Safety and Automotive

► Increasing importance of functional safety:
  • New electronic systems open new opportunities for original equipment manufacturers (OEMs)
  • Public awareness due to surprising failures
  • Upcoming safety standard ISO 26262

► Functional safety costs money:
  • More components in the Electronic control Unit (ECU)
  • More complex system/SW due to failure detection
  • Additional work for safety assessment/certification

► Cost reduction by microcontroller (MCU) with integrated safety concept
  • Less components, SW simplification, safety documentation
  • MPC564xL won “Most innovative Microprocessor 2009” award of EDN
Introduction

► Presenter: Markus Baumeister, Automotive System Engineer
► Expertise: Functional Safety
► This session should last about two hours.
Session Objectives

► After completing this session you should be able to:
  • Integrate MPC564xL into your safety-relevant (SR) system concept
  • Decide in which mode MPC564xL to use
  • Know the respective software requirements
  • Roughly explain MPC564xL’s safety concept to an assessor

► Note: No hands-on programming session
  ⇒ Wed. 14:00; Gene Fortanely, Multicore Initiation: System Initialization for the MPC5643L
Agenda

► Example system: Electric Power Steering (EPS)
  • EPS with MPC564xL
► MPC 564xL as a Safety Element out of Context
► The safety standards
► MPC 564xL’s safety concept
► Using MPC564xL in decoupled mode
Electric Power Steering

Picture from: SAE TECHNICAL PAPER SERIES 1999-01-0401
by Dominke Peter and Ruck Gerhard ZF Lenksysteme GmbH

SBC=System Basis Chip

Freescale, the Freescale logo, AllVec, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc.,
Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MXC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SMARTMOS, TurboLink
and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
Assumed Safety Concept

► MCU checks sensors and actuators
  • Sensor data plausibility or redundant comparison
  • Closed loop actuator control

► Safety MCU does application-specific check of MCU behavior
  • Position sensor ↔ Torque sensor
  • Possibly further checks using SPI transmitted data
  • Check of safety MCU by main MCU

► Failure signaling by safety MCU
  • Disables power to motor and shortcuts motor coils to reduce resistance against mechanical moves
  • Independent clocking from main MCU
Issues with Concept

► Component count
  • Safety MCU
    ▪ Possibly second oscillator
  • Stuck-at propagation protection
  • Signals go to two components

► SW effort
  • Software for safety MCU required
    ▪ No false negatives
    ▪ No/very few false positives
    ▪ Might require synchronization with main MCU

► Possible Common Cause Failures requiring additional handling
  • Common power supply for MCU and safety MCU
  • Problem if safety MCU only snoops sensor information controlled by main MCU
Electric Power Steering

SBC=System Basis Chip

Picture from: SAE TECHNICAL PAPER SERIES 1999-01-0401 by Dominke Peter and Ruck Gerhard ZF Lenksysteme GmbH
EPS with MPC564xL

Picture from: SAE TECHNICAL PAPER SERIES 1999-01-0401
by Dominke Peter and Ruck Gerhard ZF Lenksysteme GmbH
Safety Concept

► Stand-alone MCU
  • Integrated detection of HW failures
  • Sensor and actuator correctness as before

► Minor crosschecks with SBC to ensure each other’s integrity
  • Voltage supervision
  • Simple watchdog

► Failure signaling by MCU and SBC
  • Same shut off mechanism for actuators as before

► MCU will be certified as “SIL 3 capable” easing system certification
Advantages

► Less components

► Less connections on printed circuit board (PCB)

► No distributed software system

► Part of safety case provided by documentation of “SIL 3 capable” MCU
Example system: Electric Power Steering
- EPS with MPC564xL

MPC 564xL as a Safety Element out of Context
- The safety standards
- MPC 564xL’s safety concept
- Using MPC564xL in decoupled mode
SIL 3 Capable?

► (A)SIL=(Automotive) Safety Integrity Level
  • Defines necessary risk reduction
  • Valid only for a safety function of a system

► Safety Function
  • Function which “is intended to achieve or maintain a safe state for the EUC” (IEC 61508-4)
  • E.g., Steer car according to user input at steering wheel

► Individual components can not conform to a SIL

► Solution in new standards
  • Safety Element out of Context (ISO 26262)
  • Compliant Items/Element Safety Function (IEC 61508 Ed.2)
Safety Element out of Context (SEooC)

- Interactions with components outside of MCU

- Assumptions on interactions
  - Services expected from MCU
  - Services provided by external components
  - Software executed on MCU

- Integration of an SEooC
  - Check documented assumptions
  - Use safety metrics of element in system safety analysis
Assumptions made for MPC564xL

► HW assumptions
  • Externally supervised power
  • External simple watchdog
  • “Safety switch” connected to Error_out
  • PWM dead-time violation mitigator

► SW assumptions, e.g.
  • Triggering of external watchdog
  • I/O safety concept
  • Configuration checking

► Function assumptions, e.g.
  • Safe states

► All specified in “Safety Application Guide”
Excerpt of SW Assumptions for MPC 564xL

► Configuration tasks
  • Adapt failure reaction configuration of Fault Collection and Control Unit
  • Initialize usage of MPU and register locks
    ▪ Protection between cores as well as against lower SIL SW
  • Switch on clock monitoring
  • Initiate SW-triggered self-tests once

► Checking tasks
  • Periodic configuration register check
    ▪ DMA → CRC unit
  • Periodic Flash ECC logic test
    ▪ Pattern in Flash → DMA → CRC unit
  • Detection of spurious or missing IRQs caused by EMI

► I/O safety concept
Summary SEooC

► New standards allow certification of individual HW components
► Based on assumptions made during component design
  • Safety goals
  • External hardware
  • Executed software
► Deployed-in system must fulfill assumptions
► Very similar to current practice
  • Usage restrictions in “Safety Application Guide”
Example system: Electric Power Steering
  • EPS with MPC564xL

MPC 564xL as a Safety Element out of Context

The safety standards

MPC 564xL’s safety concept

Using MPC564xL in decoupled mode
Two relevant safety standards

- IEC 61508 (in revision)
  - Generic standard for functional safety of electronic systems
- ISO 26262 (in preparation)
  - ‘Derivate’ of IEC 61508 for automotive applications
  - Already in use although not complete

Goal

- Prevent unacceptable risk due to failures of equipment

Approach

- Reduction of Systematic failures (Prevention)
  - Human-introduced ‘bugs’
  - Constraints on development process
- Reduction of Random failures (Detection)
  - Failures due to aging, interference, …
  - Quantitative requirements via Safety metrics
Random Failures and their Handling

 ► Single Point Failure (SPF)
   • Immediate potential to cause a hazard
   • Quick detection or mitigation

 ► Latent Failure (LF)
   • Can become dangerous in conjunction with a second fault
   • Can aggregate
   • Periodic detection

 ► Common Cause Failure (CCF)
   • Causes several components to fail
   • Can possibly annul redundancy-based measures
   • Mitigation or quick detection
Quantitative Requirements of IEC61508 versus ISO26262

IEC 61508:
- Four Safety Integrity Levels (SIL)
- Two key metrics
  - Probability of dangerous failure per hour (PFH)
  - Safe Failure Fraction (SFF)
- Detailed requirements for CCF mitigation in upcoming edition

<table>
<thead>
<tr>
<th>SIL 1</th>
<th>SIL 2</th>
<th>SIL 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFH [1/h]</td>
<td>&lt;10^-5</td>
<td>&lt;10^-6</td>
</tr>
<tr>
<td>SFF (HFT=0)</td>
<td>&gt;=60%</td>
<td>&gt;=90%</td>
</tr>
<tr>
<td>SFF (HFT=1)</td>
<td>-</td>
<td>&gt;=60%</td>
</tr>
</tbody>
</table>

ISO 26262:
- Four Automotive SILs (ASIL)
- Three key metrics
  - Probability of violation of safety goal (PVSG)
  - Single Point Fault metric (SPFM)
  - Latent Fault Metric (LFM)
- General requirements for CCF analysis

<table>
<thead>
<tr>
<th>ASIL B</th>
<th>ASIL C</th>
<th>ASIL D</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVSG [1/h]</td>
<td>&lt;10^-7 (recom.)</td>
<td>&lt;10^-7</td>
</tr>
<tr>
<td>SPFM</td>
<td>&gt;90%</td>
<td>&gt;97%</td>
</tr>
<tr>
<td>LFM</td>
<td>&gt;60%</td>
<td>&gt;80%</td>
</tr>
</tbody>
</table>

Note: Table adopted for typical automotive application
Example system: Electric Power Steering
- EPS with MPC564xL

MPC 564xL as a Safety Element out of Context

The safety standards

MPC 564xL’s safety concept

Using MPC564xL in decoupled mode
MPC564xL and the Failure Classes

► Single Point Failure (SPF)
  • Structural redundancy
    ▪ Core, cache, bus, DMA, INTC, watchdog, RAM-Ctrl, Flash-Controller
  • Information redundancy
    ▪ ECC on system RAM and Flash, EDC on Cache

► Latent Failure (LF)
  • HW-Self test
    ▪ Memory, logic, some peripherals
    ▪ 90% coverage

► Common Cause Failure (CCF)
  • Measures according to IEC61508-2 Ed.2 Annex E
  • Supervision of clock, power and temperature
  • Independent safety clock
  • Independent failure signaling
MPC5643L Safety Elements – Module View

**Sphere of Replication:**
- Replicated e200Core
- replicated eDMA
- redundant INTC, SWT, etc
- redundant MMU
- RC Units at Gates to non redundant sphere

**XBAR + MPU:**
- Redundant
- RC Units at Gates to non redundant sphere

**Clock Monitoring:**
- Detects and mitigates clock disturbances
- PLL

**Timer:**
- eTimer0 channels “isolated”

**ADC:**
- On Line Assisted Hardware BIST

**Fault Collection Unit:**
- detects when errors have occurred
- indicates error to external
- independent of software operation
► Example system: Electric Power Steering

► MPC 564xL’s safety concept

► SPF detection: Lockstep Mode

► SPF mitigation: ECC & Multiplexing

► Failure reaction control: FCCU

► I/O safety concept
Sphere of Replication – 2oo2 principle

- same input data
- replicated processing
- different output data
  \[\Rightarrow\text{error}\]
Sphere of Replication – What to Replicate

CPU1

Bus

RAM Ctrlr

Peripheral Bridge

RAM

I/O

CPU2

➤ Only the core
Sphere of Replication – What to Replicate

- Only the core
- Most of the comp. subsys
Sphere of Replication – What to Check

CPU1

Bus1

RAM Ctrlr 1

Peripheral Bridge 1

CPU2

Bus2

RAM Ctrlr 2

Peripheral Bridge 2

RAM

I/O
Sphere of Replication – What to Check

Sphere of Replication

CPU1

Bus1

RAM Ctrlr 1

Peripheral Bridge 1

CPU2

Bus2

RAM Ctrlr 2

Peripheral Bridge 2

RAM

I/O
MPC564xL’s Safe Mode of Operation: LockStep Mode

- **MCU mode** which allows SIL3 with minimal software overhead
- Software executes automatically on both cores
- Application sees one logical core
- Checkers (RC) guarantee detection of non-CCFs when redundant channels are merged
- Failure handling in FCCU
- Selected via shadow bit in Flash during boot

**LSM**
LockStep Mode
MPC564xL’s Safe Mode of Operation: LockStep Mode

- **MCU mode** which allows SIL3 with minimal software overhead
- Software executes automatically on both cores
- Application sees one logical core
- Checkers (RC) guarantee detection of non-CCFs when redundant channels are merged
- Failure handling in FCCU
- Selected via shadow bit in Flash during boot

**LSM**
LockStep Mode

not visible to software
Example system: Electric Power Steering

MPC 564xL’s safety concept

- SPF detection: Lockstep Mode
- SPF mitigation: ECC & Multiplexing
- Failure reaction control: FCCU
- I/O safety concept
Increasing Safety and Availability: Error Correction

► SRAM is largest contributor of (transient) error rate
  • Ca. 2000 FIT on MPC564xL

► Simple detection would lead to low availability

► SEC/DED ECC
  • Masks 1 bit errors

► Problem: Multi Bit Upsets (MBU)
  • Rate is not negligible
  • 2MBU decreases availability
  • >2MBU decreases safety

Neutron-caused MBU percentages of different 90nm technologies [Internal Report]
Additional Countermeasures Against MBUs

Solution: Column Multiplexing
- Spreads logical bits over physical ones
- MBU flips only one logic bit per ECC-protected word
- MUXing-factor depends on expected MBU size
- MPC564xL uses 8 times column MUXing
Safety Concept for Cache

► No data cache
► I-Caches are duplicated to ensure high diagnostic coverage
  • Additional EDC
► To improve availability:
  • EDC detects errors
    ▪ Erroneous cache lines invalidated and an exception is raised
  • Lock-stepped cores propagate EDC errors
    ▪ Invalidation of cache line in both cores
  • Both cores re-fetch
    ▪ Avoids the execution flow of one core to drift away
Example system: Electric Power Steering

- MPC 564xL’s safety concept
  - SPF detection: Lockstep Mode
  - SPF mitigation: ECC & Multiplexing
  - Failure reaction control: FCCU
  - I/O safety concept
FCCU Concept and Purpose

► Fault Collection and Control Unit (FCCU)
  • Provide independent failure reaction
  • Supervise critical control signals
  • Allow configurable failure reactions

► Configurable and graded fault control:
  • Internal reactions
    ▪ No internal reaction
    ▪ IRQ
    ▪ Reset
  • External reaction
    ▪ Reported to the outside world via output pin.
FCCU Finite State Machine: Ensuring Internal Reaction

► Internal reaction IRQ gives chance to mitigate error with SW
► Danger: incorrect SW execution due to failure
► FCCU state machine checks correct error recovery

- On error, FCCU moves to the **ALARM** state or to the **FAULT** state, depending on the user configuration.
- **ALARM** state is kept for a programmable timeout. If error is not recovered, FCCU moves to **FAULT** state.
- Actions in **ALARM** and **FAULT** state are configurable.
Path Redundancy on Critical Error Reaction

► Detected *critical* errors are forwarded independently to
  • Fault collection and control unit *and*
  • Reset Generation Module (RGM)

► Additionally:
  • The state of the RGM is forwarded to the FCCU
  • The FCCU forwards an additional reset request to the RGM

► Decreases possibility of common cause failures on the safety path

► Both need to be configured
Example system: Electric Power Steering

- MPC 564xL’s safety concept
  - SPF detection: Lockstep Mode
  - SPF mitigation: ECC & Multiplexing
  - Failure reaction control: FCCU
  - I/O safety concept
### Safety Mechanisms for Peripherals: SPI Example

<table>
<thead>
<tr>
<th>Failure mode</th>
<th>Cause (MCU internal)</th>
<th>Safety mechanism (MCU level)</th>
<th>Cause (External to MCU)</th>
<th>Coverage of MCU mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrong addressing</td>
<td>d.c. failure in client selection register</td>
<td>comparator (feedback written value)</td>
<td>d.c. failure in address lines or sensor input (for CS bridging: one sensor dominates the other when answering); soft error in sensor selection indication register</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>soft error in client selection register</td>
<td>register hardening or ECC protection</td>
<td></td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>d.c. failure in chip select driver</td>
<td>reencoding of pad states</td>
<td></td>
<td>depends on how strong fault feeds back into pad voltage</td>
</tr>
<tr>
<td>&quot;Babbling idiot&quot;</td>
<td>stuck at 1 of chip select output</td>
<td>reencoding of pad states</td>
<td>stuck at 1 of sensor CS input</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>stuck at 1 in client selection register</td>
<td>reencoding of pad states</td>
<td>stuck at 1 of sensor internal selection logic or registers</td>
<td>none</td>
</tr>
<tr>
<td>Corrupted Value</td>
<td>soft error in shift register</td>
<td>IP supervision</td>
<td>Any sensor failure</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>clock too fast</td>
<td>clock supervision; IP supervision</td>
<td></td>
<td>none</td>
</tr>
</tbody>
</table>

**MCU safety mechanisms DO NOT cover external faults!**

System level techniques are needed.
### Safety Mechanisms for Peripherals: SPI Example (cont’d)

<table>
<thead>
<tr>
<th>Failure mode</th>
<th>Cause (External to MCU)</th>
<th>Safety mechanism (system level)</th>
<th>Cause (MCU internal)</th>
<th>Coverage of system level mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrong addressing</td>
<td>d.c. failure in sensor selection indication register</td>
<td>double read/write OR sensor ID</td>
<td>d.c. failure in client selection register</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>soft error in sensor selection indication register</td>
<td>double read/write OR sensor ID</td>
<td>soft error in client selection register</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>d.c. failure in address lines or sensor input</td>
<td>double read/write OR sensor ID</td>
<td>d.c. failure in chip select driver</td>
<td>yes</td>
</tr>
<tr>
<td>&quot;Babbling idiot&quot;</td>
<td>stuck at 1 of sensor CS input</td>
<td>double read/write OR sensor ID OR application checksum</td>
<td>stuck at 1 of chip select output</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>stuck at 1 of sensor internal selection logic or registers</td>
<td>double read/write OR sensor ID OR application checksum</td>
<td>stuck at 1 in client selection register</td>
<td>yes</td>
</tr>
<tr>
<td>Corrupted Value</td>
<td>Measurement failure</td>
<td>double read/write</td>
<td>soft error in shift register</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>Comm logic failure</td>
<td>double read/write OR application checksum</td>
<td>clock too fast</td>
<td>Depends on source of clock failure</td>
</tr>
</tbody>
</table>

System level safety mechanisms DO cover MCU internal faults … PROVIDED THAT independency of redundant operations is enforced
Example System Level Mechanism: PWM Out with Read-back

► PWM output
► Read back into eTimer
► Internal or external read-back path
  • Internal: uses Pad logic
  • External: external connection to another pin
► Synchronization of eTimer acquisition with PWM output via CTU
► Application software must implement the read-back comparison
  • Read back values stored in CTU FIFO
  • Must be checked against PWM configuration (e.g. period and duty cycle)
MCU vs. System Level Safety Mechanisms: Summary

► MCU HW safety mechanisms
  • Can detect failures on I/O modules level
  • Impact on silicon area, power consumption (and possibly performance)
  • No/limited coverage against faults outside of the MCU
    ▪ E.g. bonding, wiring or sensor failures

► System level safety mechanisms
  • Needed whether MCU mechanisms are provided or not
  • Guarantee coverage of faults both inside and outside MCU
    ▪ Independence of the two I/O channels is required

Replicated I/O modules for system level safety mechanisms
Summary - MPC564x Safety MCU Overview

- Dual Core
- Lock-Step
- Sphere of Replication
- ECC
- Protected Memories
- Replicated Peripherals
- BIST
- Fault Collection Control Unit
Example system: Electric Power Steering

- EPS with MPC564xL

MPC 564xL as a Safety Element out of Context

The safety standards

MPC 564xL’s safety concept

Using MPC564xL in decoupled mode
A Key Question

Can one size fit all?
1. Calibration of Ratings via Reference Architecture

<table>
<thead>
<tr>
<th>Rating</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing penalty (core level)</td>
<td>1 Assuming $T_{\text{consolidation}} &lt;&lt; T_{\text{app_task}}$</td>
</tr>
<tr>
<td>Energy factor (core level)</td>
<td>1</td>
</tr>
<tr>
<td>Execution latency</td>
<td>1</td>
</tr>
<tr>
<td>Software complexity factor</td>
<td>Low</td>
</tr>
<tr>
<td>Detection of software errors</td>
<td>0</td>
</tr>
<tr>
<td>Utilization factor for non-safety tasks</td>
<td>1</td>
</tr>
</tbody>
</table>

Simplex ... fault-free per design demonstrable

Complex ... fault-free per design not demonstrable
2. Dual Core Lockstep

- **Core 1**
- **Core 2**
- **Aux 1**
- **Aux 2**
- **Aux 3**

### Rating (approx.) | Comment
--- | ---
**Processing penalty (core level)** | ~ 0.5
**Energy factor (core level)** | ~ 2x 2 cores
**Execution latency** | 1 Execution time equivalent to single core
**Software complexity factor** | Low Looks like single core from SW perspective
**Detection of software errors** | 0
**Utilization factor for non-safety tasks** | ~ 0.5

Simplex … fault-free per design demonstrable
Complex … fault-free per design not demonstrable
3. Dual-Core Lockstep & Diverse Tasks in Time Redundancy

<table>
<thead>
<tr>
<th>Rating (approx.)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing penalty (core level)</td>
<td>~ 0.25 Assuming ( T_{\text{consolidation}} &lt; T_{\text{app_task}} )</td>
</tr>
<tr>
<td>Energy factor (core level)</td>
<td>~ 4x 2 cores x 2 executions in ‘series’</td>
</tr>
<tr>
<td>Execution latency</td>
<td>~ 2x</td>
</tr>
<tr>
<td>Software complexity factor</td>
<td>Medium Diverse SW, but single linear execution flow</td>
</tr>
<tr>
<td>Detection of software errors</td>
<td>1</td>
</tr>
<tr>
<td>Utilization factor for non-safety tasks</td>
<td>~ 0.5</td>
</tr>
</tbody>
</table>
4. Dual-Core & Diverse Tasks in Parallel

<table>
<thead>
<tr>
<th>Rating (approx.)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing penalty (core level)</td>
<td>~ 0.5x</td>
</tr>
<tr>
<td>Energy factor (core level)</td>
<td>~ 2x</td>
</tr>
<tr>
<td>Execution latency</td>
<td>~ 1x</td>
</tr>
<tr>
<td>Software complexity factor</td>
<td>High</td>
</tr>
<tr>
<td>Detection of software errors</td>
<td>1</td>
</tr>
<tr>
<td>Utilization factor for non-safety tasks</td>
<td>1</td>
</tr>
</tbody>
</table>

Simplex … fault-free per design demonstrable
Complex … fault-free per design not demonstrable
MPC564xL’s Second Mode of Operation: Decoupled Parallel Mode

- MCU mode which allows SIL3 with software replication
- CPU cores and subsystems run independently and are visible to application
- Checker units (RC) are disabled in this mode
- Latent faults and CCF still handled in hardware
- Additional SPF s need to be handled in software
- Channel isolation using MMU, MPU & I/O-Bridge
MPC564xL in DPM and the Failure Classes

► Single Point Failure (SPF)
  • Structural redundancy
    ▪ Core, cache, bus, DMA, INTC, watchdog, RAM-Ctrl, Flash-Controller
  • Information redundancy
    ▪ ECC on system RAM and Flash, EDC on cache

► Latent Failure (LF)
  • HW self-test
    ▪ Memory, logic, some peripherals
    ▪ 90% coverage

► Common Cause Failure (CCF)
  • Measures according to IEC61508-2 Ed.2 Annex E
  • Supervision of clock, power and temperature
  • Independent safety clock
  • Independent failure signaling
DPM Software Architectures

► Standard Software replication
  • Symmetric redundancy
  • Run safety-related SW twice
  • Compare results (on both cores)
  • SW effort for synchronization & comparison only

► Master-Checker architecture
  • Asymmetric redundancy
  • Second SW variant (checker)
    ▪ Cannot control thus simpler
  • SW effort for “1½” application implementations
DPM Software Architectures (2)

► Independent preprocessing
  • Partial redundancy
  • Failures in preprocessing discovered/masked in safety processing
  • Better usage of performance

► MCU sharing
  • External redundancy
  • Safety actually achieved by external measures (e.g. ASIC)
  • Two cores used for software isolation (ISO 26262)
Avoid single point of failure in MCU

- Easy if several parallel actuators exist
- Easy if an actuator is intelligent enough to decode e.g. a CRC
Alternative: Use Actuator Feedback Loop

► Previous solutions not always possible
  • Single, non-intelligent actuator
  • No protection of transmission line

► Use feedback to check
  • Correct command requested
    ▪ Control point 1
  • Correct command sent
    ▪ Control point 2
  • Correct command executed
    ▪ Control point 3

► Part of I/O safety concept
► Might require additional I/O
Peripheral modules are replicated

In DPM: Path to them partially shared
  • In LSM: Replicated

Possible measures
  • Sensor diversity
  • Online self-test
    ▪ Read all SR I/O modules
    ▪ Read all bits
    ▪ Exercise all relevant address bits
    ▪ Write
Additional Software Measures in DPM

► Decoupled parallel mode disables HW redundancy checking

► Additional measures necessary to replace it
  • SW architecture (shown)
  • Other measures for
    ▪ Prevention of channel interference, e.g.
      – Watchdog (only use for attached core)
      – Ram controller (preferably use half of RAM “near” to core)
    ▪ SPF detection, e.g.
      – I/O-Bridges (software test for wrong addressing/data mangling)
      – Crossbar on I/O access (same as above)
      – DMA checks

► White paper available to explain possible measures
  ▪ Keyword: “MPC564xLWP” on freescale.com
## Comparison of Different Dual-core Modes for Selected Fault Examples

<table>
<thead>
<tr>
<th>Mode</th>
<th>Dual-core w/o safety</th>
<th>Dual-core w/o safety + SW Rep</th>
<th>Dual-core w/o safety + SW Div</th>
<th>DPM</th>
<th>DPM + SW Rep</th>
<th>DPM + SW Div</th>
<th>LSM</th>
<th>LSM + SW Div</th>
<th>Adapted LSM</th>
<th>Adapted LSM + SW Div</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPU-Fault (Single channel fault)</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>INTC fault (stop fault)</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Voltage too low (CCF)</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CAN clock (safe fault)</td>
<td>Annoyance failure not caught</td>
<td>Reaction highly SW dependent</td>
<td>Shutdown</td>
<td>Graceful degradation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW fault</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Modes possible with MPC564xL**
Session Summary

MPC564xL

► SIL 3 computing on one chip
  • Removes need for Safety MCU

► Several measures to fulfill IEC 61508/ISO 26262 requirements
  • Structural & information redundancy
  • HW self-tests
  • CCF countermeasures

► Flexible usage
  • LSM mode for high functional safety
  • DPM mode for increased performance
    ▪ Shifting some safety jobs to SW

► Available now
  • Engineering samples
  • Qualified samples expected Q1 2011