Freescale Technology Forum

Collaboration. Innovation. Inspiration.

July 14–16, 2009

AA129 - Reliability in Embedded Systems

Safety Standards and Self Tests

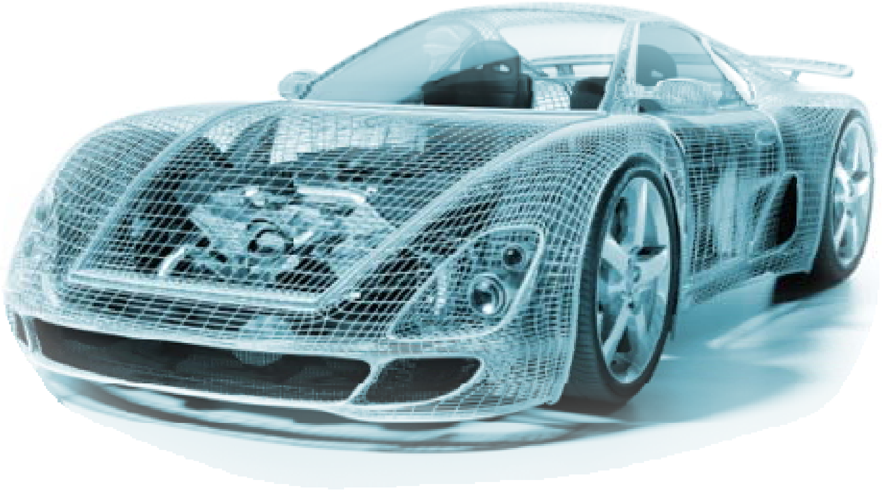
Christopher Temple

Automotive Systems Technology Manager

freescale™ semiconductor

Overview

- Introduction
- IEC61508 Safety Standard
- ISO26262 Safety Standard (draft)
- MCU Safety Continuum
- Basic Core Self-Test
- Summary



Introduction

Freescale Introduces Product Longevity Program

- ►The embedded market needs long-term product support, which allows OEMs to provide assurance to their customers
- ►Freescale has a longstanding track record of providing long-term production support for our products
- ►Freescale is pleased to introduce a formal product longevity program for the market segments we serve
  - For the automotive and medical segments, Freescale will manufacture select devices for a minimum period of 15 years
  - For all other market segments in which Freescale participates, Freescale will manufacture select devices for a minimum period of 10 years
- ►A list of applicable Freescale products is available at www.freescale.com.

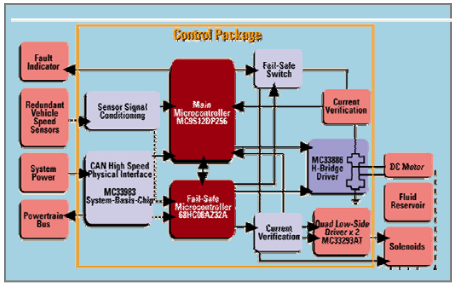
Automotive Safety and Functional Safety

Functional Safety Systems

Automotive Safety Systems

Active Safety Systems

Passive Safety Systems


Control Package
Fault Indicator
Redundant Vehicle Speed Sensors
System Power
Powertrain Bus
Sensor Signal Conditioning
CAN High Speed Physical Interface
MC33983 System-Basis-Chip
Main Microcontroller MC9S12DP256
Fail-Safe Microcontroller 68HC08AZ32A
Fail-Safe Switch
Current Verification
MC33886 H-Bridge Driver
Current Verification
Quad Low-Side Driver x 2 MC33293AT
DC Motor
Fluid Reservoir
Solenoids

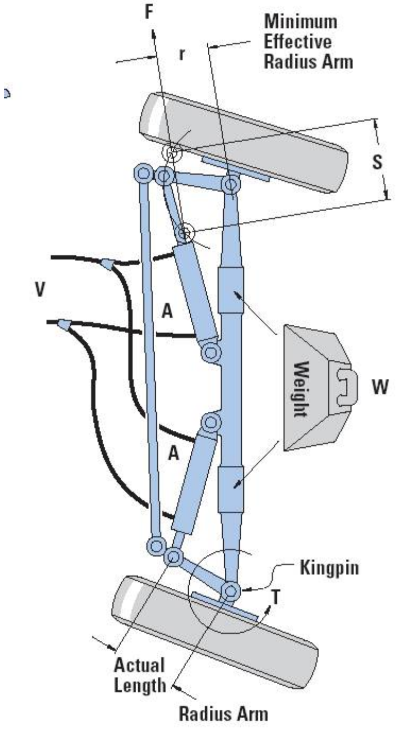
“Safety of persons in respect to accidents”

“Safety of persons in respect to (E/E/PES) system failures”

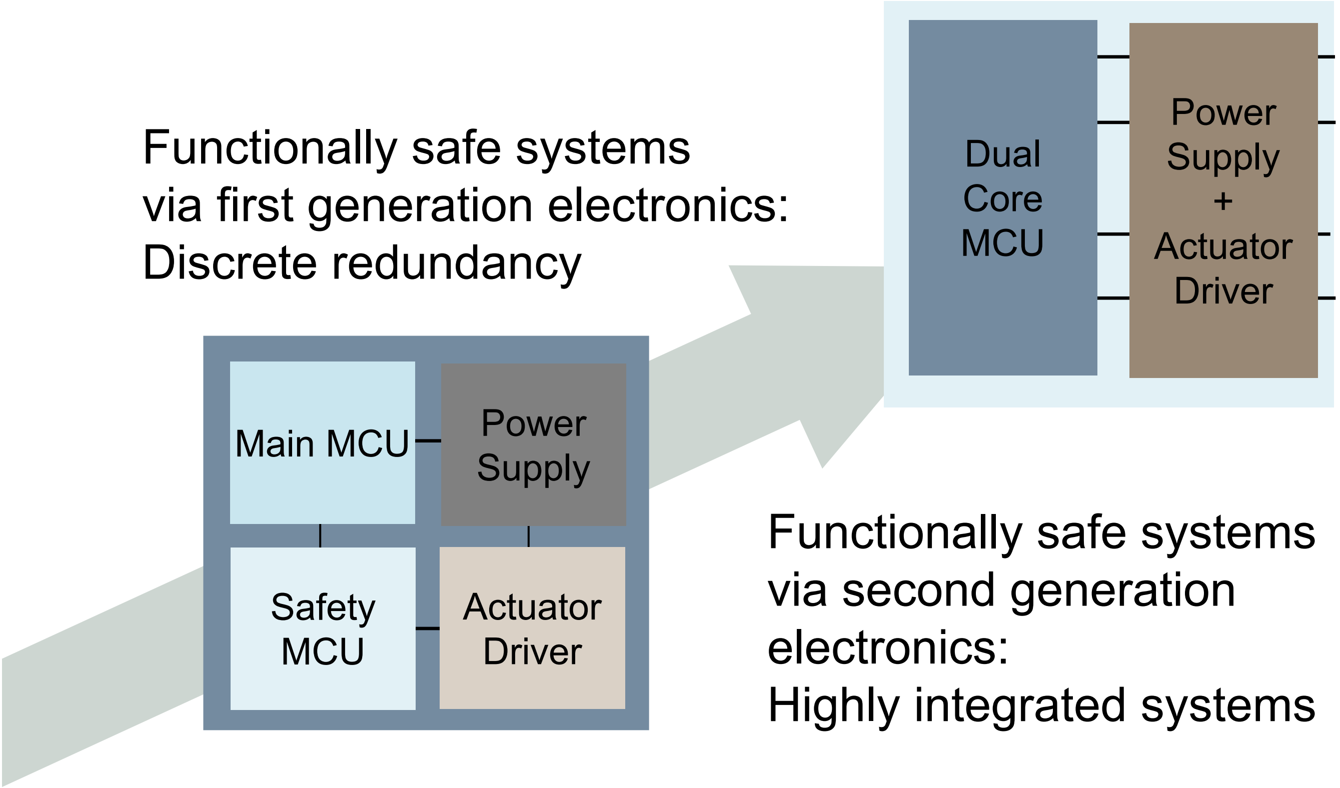
“Safety is freedom from unacceptable risk”

(IEC 61508)

Evolution of Functional Safety Approaches



Functionally safe systems
via first generation electronics:
Discrete redundancy



Main MCU

Power Supply

Safety MCU

Actuator Driver

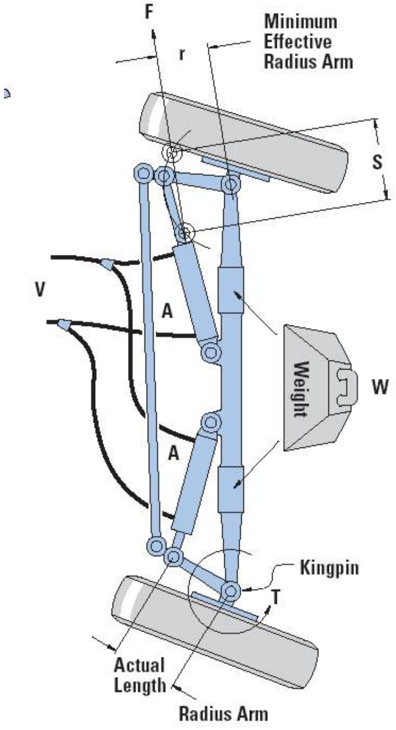
Dual Core MCU

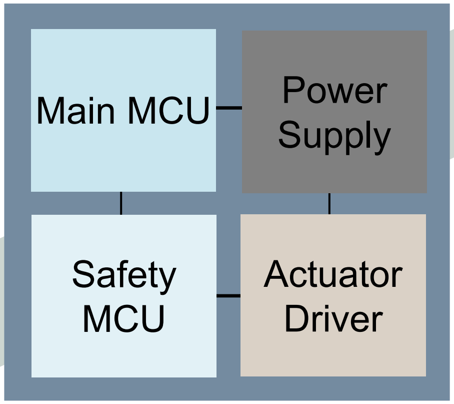
Power Supply + Actuator Driver

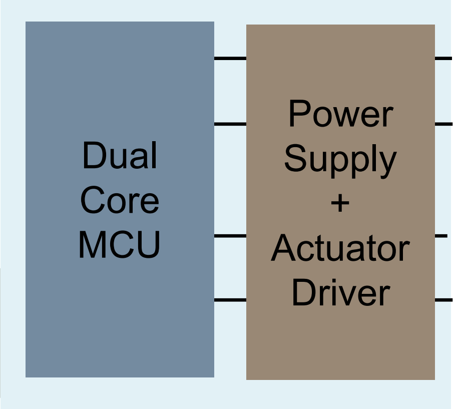
Functionally safe systems
via second generation
electronics:
Highly integrated systems

From Components to Integrated Systems

- In the past strong separation of system level and discrete component level
- Good visibility of structural system details







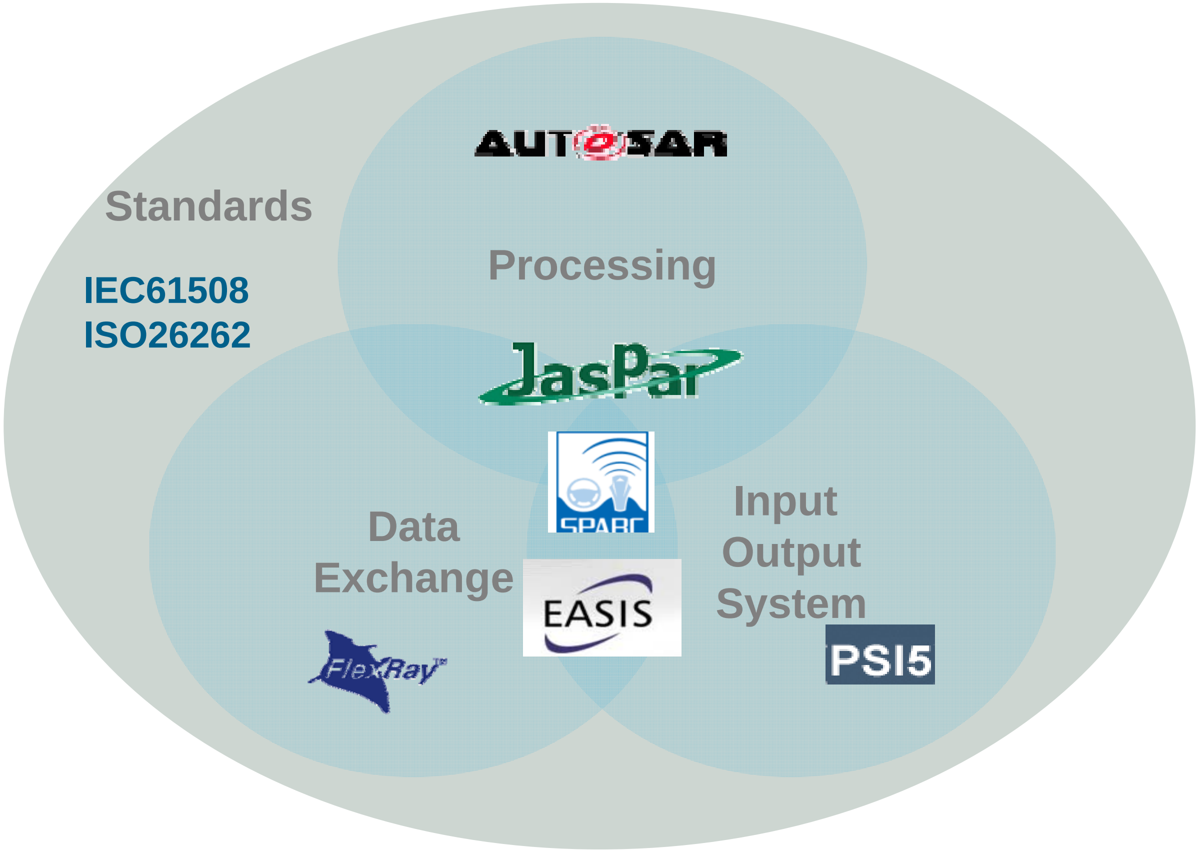
- Now complete systems are being condensed to single integrated components
- Structural system details embodied in integrated components

Systems and Standards

Semiconductor manufacturers are moving towards *safety systems* suppliers

Industry-wide *cooperation* and *standardization* emerging to harmonize system related aspects across industry

Industry-wide Initiatives for Standards



Key Safety Standards

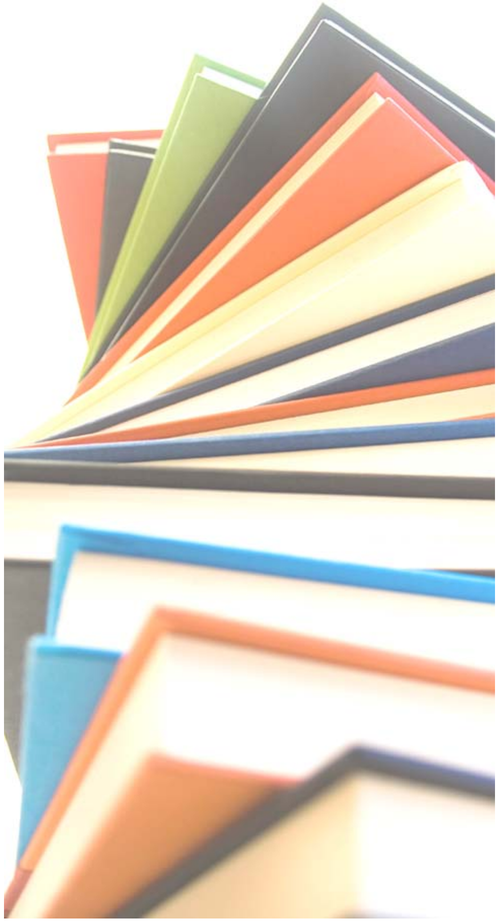
- IEC61508 (today)
- IEC61508 v2 (emerging)
- ISO26262 (emerging)

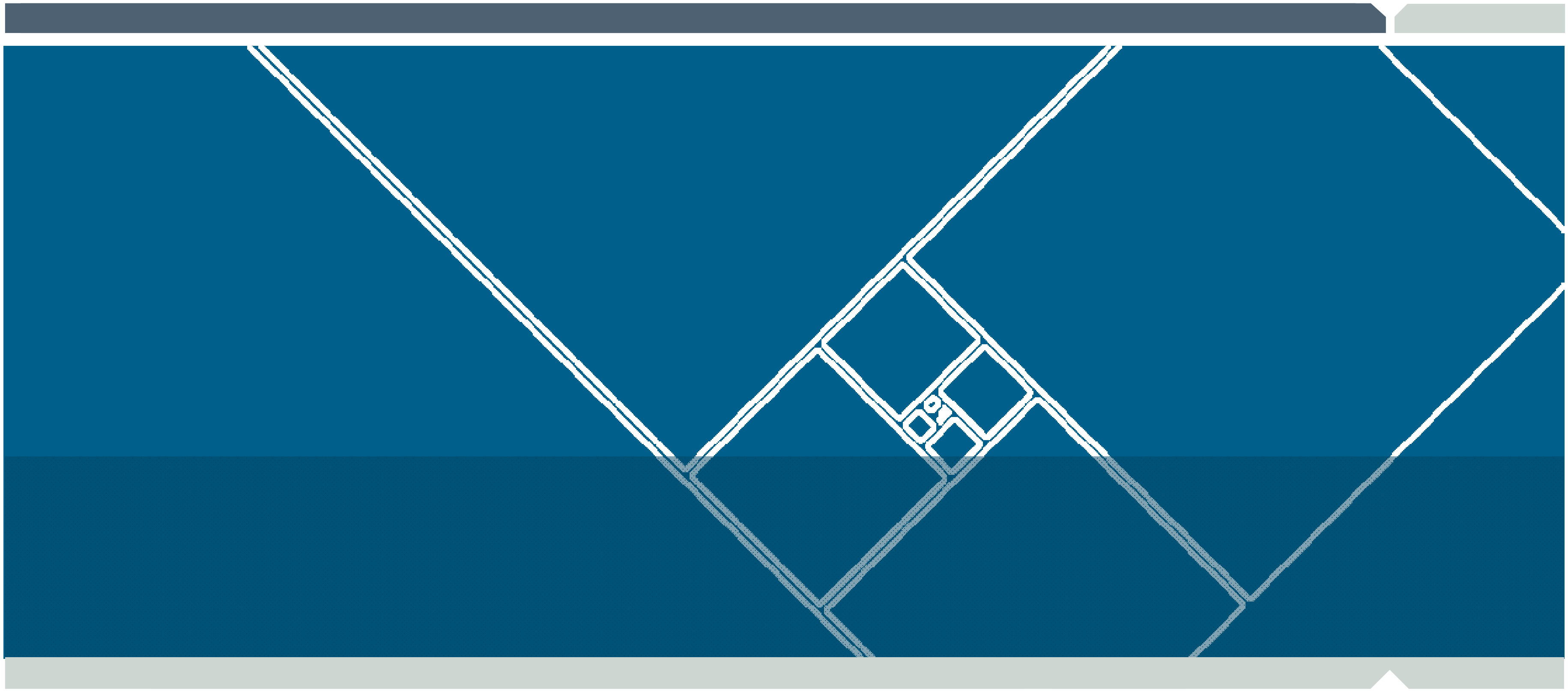
Industry Wide Initiatives

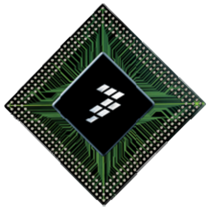
- AUTOSAR
- JasPar
- FlexRay™ Networking
- PSI5
- EC Funded SPARC, EASIS

Role of Safety Standards

- ►Standards are emerging as a framework to establish metrics and value network
- ►IEC61508
  - V1 since late 1990s, V2 announced
  - Safety lifecycle defined
  - Recommended and mandatory practices
- ►ISO26262
  - Current draft, release expected ~2011
  - Refinement of IEC61508 to comply with needs specific to the application sector of E/E systems within road vehicles







IEC61508 Safety Standard

The Seven Parts of IEC 61508

- 1: General Requirements
- 2: Requirements for electrical / electronic / programmable electronic safety-related systems *(means HW)*
- 3: Software Requirements
- 4: Definitions and abbreviations

normative

- 5: Examples of methods for the determination of safety integrity levels
- 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3
- 7: Overview of techniques and measures

NORME INTERNATIONALE
INTERNATIONAL STANDARD
CEI IEC 61508-1
Première édition
First edition
1998-12

PUBLICATION FONDAMENTALE DE SÉCURITÉ
BASIC SAFETY PUBLICATION

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité –
Partie 1:
Prescriptions générales –

Functional safety of electrical/electronic/programmable electronic safety-related systems –
Part 1:
General requirements

IEC
Numéro de référence
Reference number
CEI/IEC 61508-1:1998

NORME INTERNATIONALE
INTERNATIONAL STANDARD
CEI IEC 61508-2
Première édition
First edition
2000-05

PUBLICATION FONDAMENTALE DE SÉCURITÉ
BASIC SAFETY PUBLICATION

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité –
Partie 2:
Prescriptions pour les systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité

Functional safety of electrical/electronic/programmable electronic safety-related systems –
Part 2:
Requirements for electrical/electronic/programmable electronic safety-related systems

IEC
Numéro de référence
Reference number
CEI/IEC 61508-2:2000

NORME INTERNATIONALE
INTERNATIONAL STANDARD
CEI IEC 61508-3
Première édition
First edition
1998-12

PUBLICATION FONDAMENTALE DE SÉCURITÉ
BASIC SAFETY PUBLICATION

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité –
Partie 3:
Prescriptions concernant les logiciels

Functional safety of electrical/electronic/programmable electronic safety-related systems –
Part 3:
Software requirements

IEC
Numéro de référence
Reference number
CEI/IEC 61508-3:1998

NORME INTERNATIONALE
INTERNATIONAL STANDARD
CEI IEC 61508-4
Première édition
First edition
1998-12

PUBLICATION FONDAMENTALE DE SÉCURITÉ
BASIC SAFETY PUBLICATION

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité –
Partie 4:
Définitions et abréviations

Functional safety of electrical/electronic/programmable electronic safety-related systems –
Part 4:
Definitions and abbreviations

IEC
Numéro de référence
Reference number
CEI/IEC 61508-4:1998

NORME INTERNATIONALE
INTERNATIONAL STANDARD
CEI IEC 61508-5
Première édition
First edition
1998-12

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité –
Partie 5:
Exemples de méthodes de détermination des niveaux d'intégrité de sécurité

Functional safety of electrical/electronic/programmable electronic safety related systems –
Part 5:
Examples of methods for the determination of safety integrity levels

IEC
Numéro de référence
Reference number
CEI/IEC 61508-5:1998

NORME INTERNATIONALE
INTERNATIONAL STANDARD
CEI IEC 61508-6
Première édition
First edition
2000-04

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité –
Partie 6:
Lignes directrices pour l'application de la CEI 61508-2 et de la CEI 61508-3

Functional safety of electrical/electronic/programmable electronic safety-related systems –
Part 6:
Guidelines on the application of IEC 61508-2 and IEC 61508-3

IEC
Numéro de référence
Reference number
CEI/IEC 61508-6:2000

NORME INTERNATIONALE
INTERNATIONAL STANDARD
CEI IEC 61508-7
Première édition
First edition
2000-03

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité –
Partie 7:
Présentation de techniques et mesures

Functional safety of electrical/electronic/programmable electronic safety-related systems –
Part 7:
Overview of techniques and measures

IEC
Numéro de référence
Reference number
CEI/IEC 61508-7:2000

How does IEC61508 define Functional Safety?

- Safety
  - “freedom from unacceptable risk”
- Risk
  - “combination of the probability of occurrence of harm and the severity of that harm”
- Harm
  - “physical injury or damage to the health of people either directly or indirectly as a result of damage to property or to the environment”
- Functional safety
  - “part of the overall safety relating to the equipment under control (EUC) and the EUC control system which depends on the correct functioning of the electrical/electronic/programmable electronic (E/E/PE) safety-related systems, other technology related safety-related systems and external risk reduction facilities”

Quantitative Requirements of IEC61508

- IEC 61508
  - Four Safety Integrity Levels (SIL)
  - Two key metrics
    - Probability of dangerous failure per hour (PFH)
    - Safe Failure Fraction (SFF)
  - Hardware redundancy in formulas (HFT)

	SIL 1	SIL 2	SIL 3
PFH [1/h]	$<10^{-5}$	$<10^{-6}$	$<10^{-7}$
SFF (HFT=0)	>=60%	>=90%	>=99%
SFF (HFT=1)	-	>=60%	>=90%

Note: Table adopted for typical automotive application

Quantitative Requirements of IEC61508

- IEC 61508
  - Four Safety Integrity Levels (SIL)
  - Two key metrics
    - Probability of dangerous failure per hour (PFH)
    - Safe Failure Fraction (SFF)
  - Hardware redundancy in formulas (HFT)

	SIL 1	SIL 2	SIL 3
PFH [1/h]	$<10^{-5}$	$<10^{-6}$	$<10^{-7}$
SFF (HFT=0)	>=60%	>=90%	>=99%
SFF (HFT=1)	-	>=60%	>=90%

Note: Table adopted for typical automotive application

Safety Integrity Levels

- SIL: “discrete level for specifying the safety integrity requirements of the safety functions to be allocated to the E/E/PE safety-related systems, where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest”
- Approaches to determine the SIL
  - Quantitative methods: such as via probability of a dangerous failure per hour for continuous mode of operation
  - Qualitative methods: such as risk graph or hazardous event severity matrix

Quantitative Requirements of IEC61508

- IEC 61508
  - Four Safety Integrity Levels (SIL)
  - Two key metrics
    - Probability of dangerous failure per hour (PFH)
    - Safe Failure Fraction (SFF)
  - Hardware redundancy in formulas (HFT)

	SIL 1	SIL 2	SIL 3
PFH [1/h]	$<10^{-5}$	$<10^{-6}$	$<10^{-7}$
SFF (HFT=0)	>=60%	>=90%	>=99%
SFF (HFT=1)	-	>=60%	>=90%

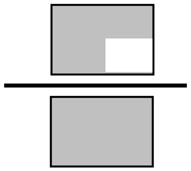
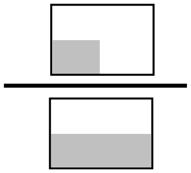
Note: Table adopted for typical automotive application

Key Metrics

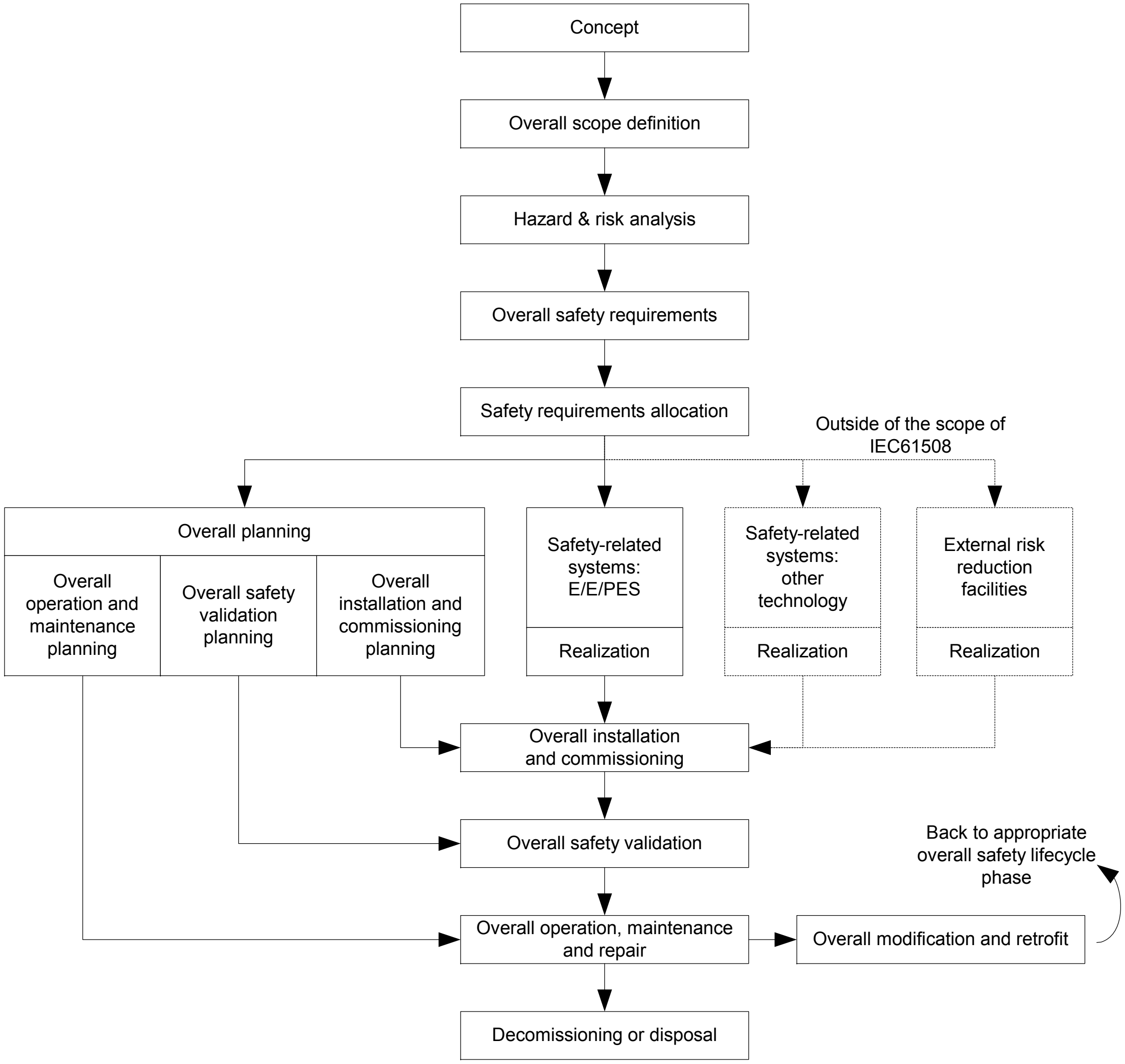
- Probability of dangerous failure per hour (PFH)
  - Target values depend on mode of system (low demand versus *high demand/continuous*), complexity of system (Type A (simplex) versus *Type B (complex)*) and additional *customer requirements*
- Safe Failure Fraction
  - the ratio of the average ***rate of safe failures plus dangerous detected failures*** of the system ***to the total average failure rate*** of the system

Safe Failure Fraction and Diagnostic Coverage

Safe, detected	Safe, undetected
Dangerous, detected	Dangerous, undetected

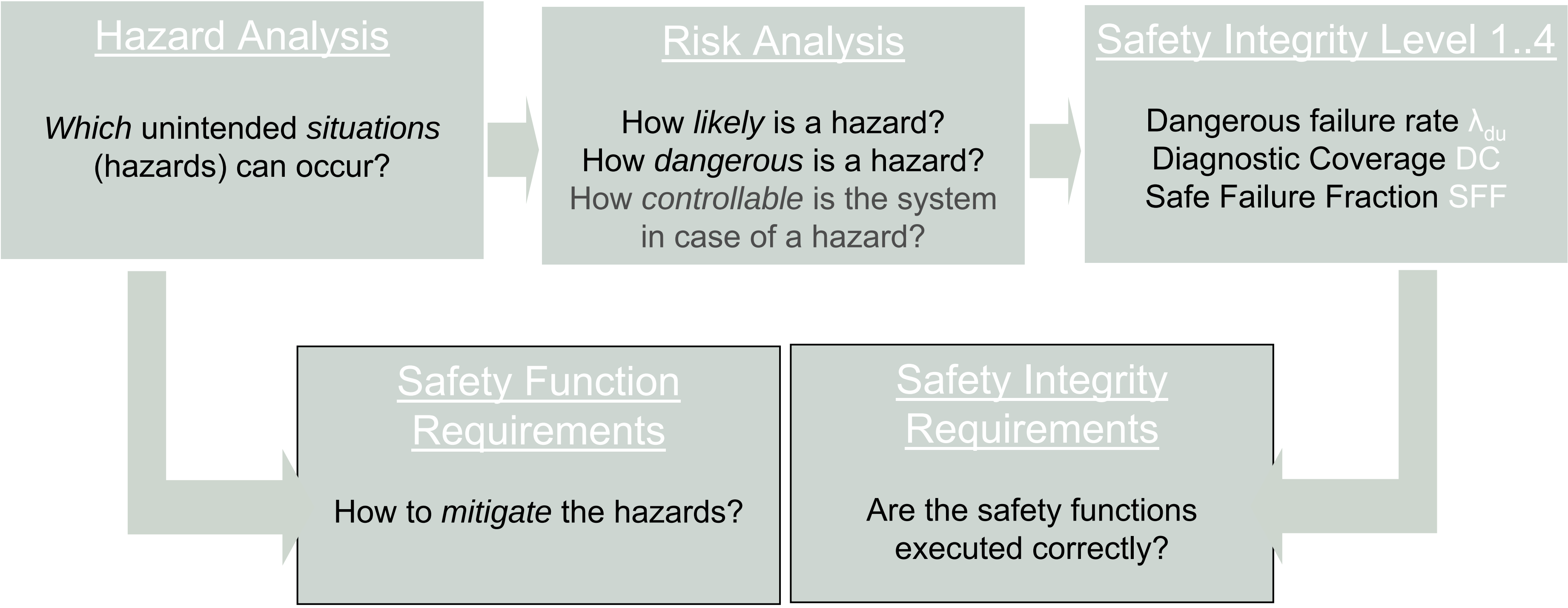
- Safe Failure Fraction=  Diagnostic Coverage= 
- Note: SFF is computed from the RATES (approx. probabilities) of the different failure classes
  - $SFF = (\sum\lambda_S+\sum\lambda_{DD})/ (\sum\lambda_S+\sum\lambda_{DD} +\sum\lambda_{DU})$
  - Where:
    - $\sum\lambda_S$: total rate of safe failures
    - $\sum\lambda_{DD}$: total rate of dangerous detected failures
    - $\sum\lambda_{DU}$: total rate of dangerous undetected failures

IEC61508 Safety Lifecycle



Concept
Overall scope definition
Hazard & risk analysis
Overall safety requirements
Safety requirements allocation
Outside of the scope of IEC61508
Overall planning
Overall operation and maintenance planning
Overall safety validation planning
Overall installation and commissioning planning
Safety-related systems: E/E/PES
Realization
Safety-related systems: other technology
Realization
External risk reduction facilities
Realization
Overall installation and commissioning
Overall safety validation
Back to appropriate overall safety lifecycle phase
Overall operation, maintenance and repair
Overall modification and retrofit
Decomissioning or disposal

Outline for Designing a Safe System



Hazard Analysis

Which unintended *situations* (hazards) can occur?

Risk Analysis

How *likely* is a hazard?
How *dangerous* is a hazard?
How *controllable* is the system in case of a hazard?

Safety Integrity Level 1..4

Dangerous failure rate λ_{du}
Diagnostic Coverage DC
Safe Failure Fraction SFF

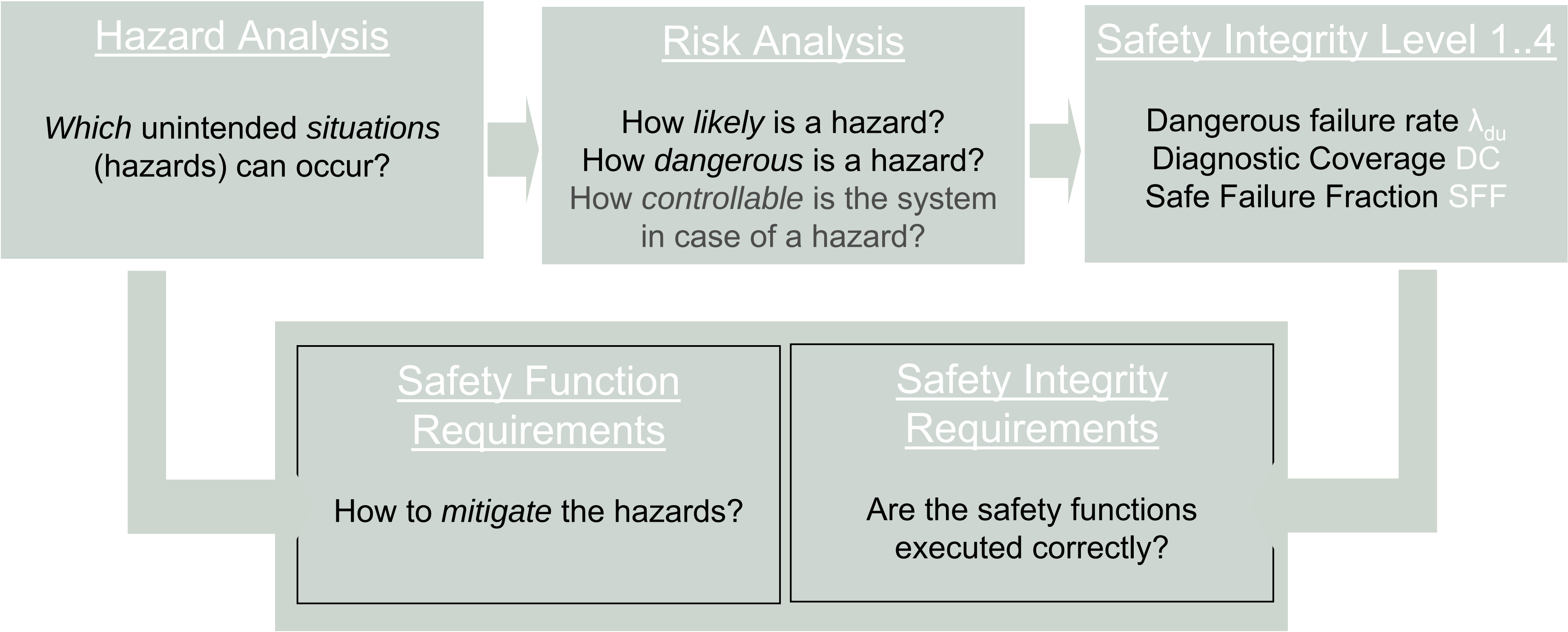
Safety Function Requirements

How to *mitigate* the hazards?

Safety Integrity Requirements

Are the safety functions executed correctly?

Outline for Designing a Safe System



Hazard Analysis

Which unintended *situations* (hazards) can occur?

Risk Analysis

How *likely* is a hazard?
How *dangerous* is a hazard?
How *controllable* is the system in case of a hazard?

Safety Integrity Level 1..4

Dangerous failure rate λ_{du}
Diagnostic Coverage DC
Safe Failure Fraction SFF

Safety Function Requirements

How to *mitigate* the hazards?

Safety Integrity Requirements

Are the safety functions executed correctly?

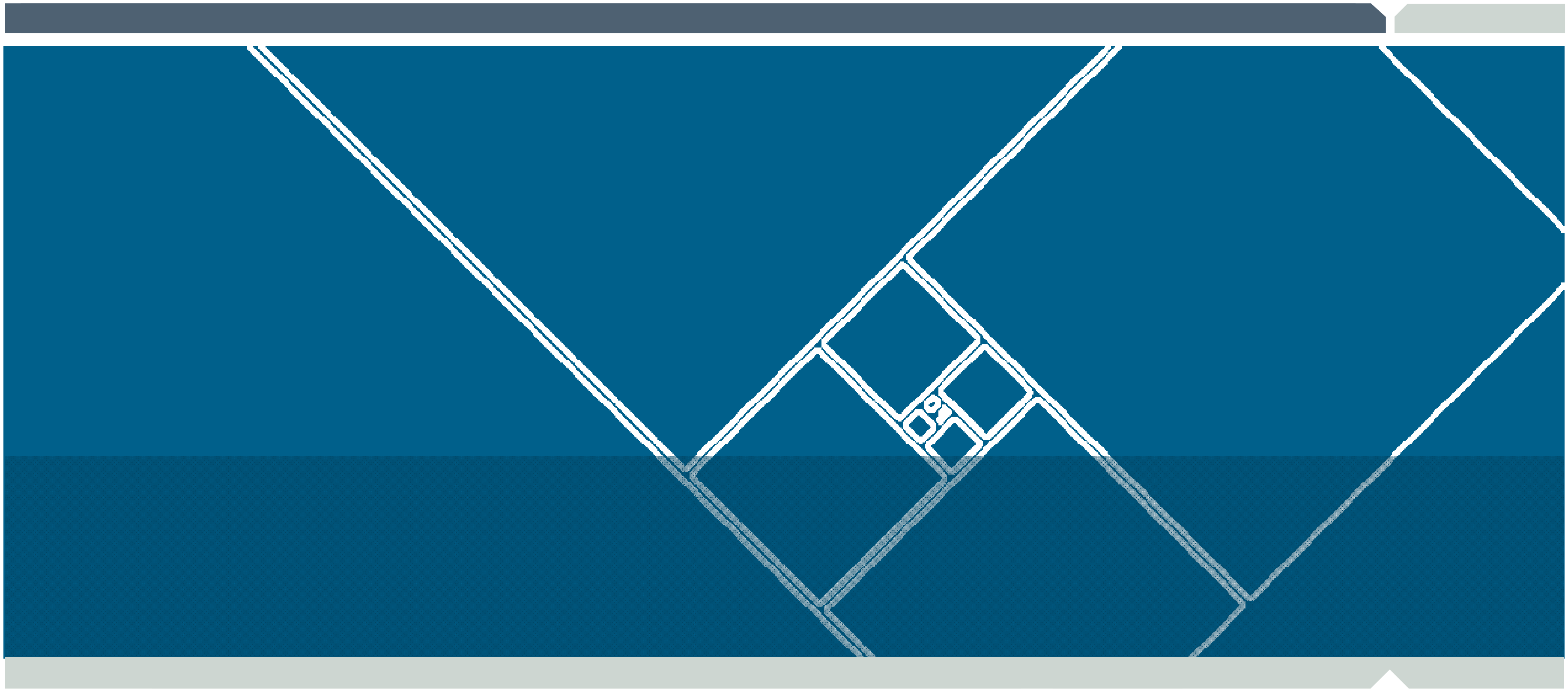
Refine the system until the remaining risk is below the highest acceptable risk

What the Standard Says for Hardware Components

Hardware safety integrity	Systematic safety integrity	Avoidance of systematic failures during the different phases of the lifecycle (relating to processes)
► Faults or failures to be analyzed in the derivation of safe failure fraction ► Faults or failures to be detected during operation	Techniques and measures to control: ► systematic failures caused by hardware and software design ► systematic failures caused by environmental stress or influences ► systematic operational failures	Recommendations to avoid mistakes: ► during specification of E/E/PES requirements ► during E/E/PES design and development ► during E/E/PES integration ►during E/E/PES operation and maintenance procedures ► during E/E/PES safety validation
► Recommended ► Highly recommended ► Mandatory measures	► Recommended ► Highly recommended ► Mandatory measures	► Recommended ► Highly recommended ► Mandatory techniques
Guidelines for assessing the maximum diagnostic coverage considered achievable through various techniques	Guidelines for assessing the effectiveness of techniques and measures to control systematic failures	Guidelines for assessing the effectiveness of techniques and measures to avoid systematic failures

Conclusion

- Applying all measures to achieve hardware safety integrity for a specific Safety Integrity Level would make a system far too expense
- The right choice of measures is required
- (Effective!) use of error detection and diagnostic capabilities to detect dangerous failures
    - Error detection measures
        - Stop errors from propagating beyond component boundary
        - Error correction (compensation)
        - Shut down (fail-silent)
    - Self test measures
        - Ensure that the device is free from dormant faults
        - Software self-test, various BIST mechanisms



ISO26262 Safety Standard (draft)

The Nine Parts of ISO26262

- ►ISO 26262 is the adaptation of IEC61508 in automotive industry
- ►ISO 26262 applies to safety related E/E systems installed in road vehicles of class M, N and O (see 70/156/EC)
- ►ISO 26262 consists of the following parts:
  - *Part 1: Glossary*
  - *Part 2: Management of functional safety*
  - *Part 3: Concept phase*
  - *Part 4: Product development: system level*
  - *Part 5: Product development: hardware level*
  - *Part 6: Product development: software level*
  - *Part 7: Production and operation*
  - *Part 8: Supporting processes*
  - *Part 9: ASIL-oriented and safety-oriented analyses (analysis techniques)*

Objective

- ISO 26262 addresses hazards caused by safety related E/E systems due to malfunctions, excluding nominal performances of active and passive safety systems
  - Provides an automotive safety lifecycle (management, development, production, operation, service, decommissioning) and supports tailoring the necessary activities during these lifecycle phases
  - Provides an automotive specific risk-based approach for determining risk classes (Automotive Safety Integrity Levels, ASILs)
  - Uses ASILs for specifying the item's necessary safety requirements for achieving an acceptable residual risk
  - Provides requirements for validation and confirmation measures to ensure a sufficient and acceptable level of safety being achieved

Quantitative Requirements ISO26262

- ISO 26262
  - Four Automotive SILs (ASIL)
  - Three key metrics
    - Probability of violation of safety goals (PVSG)
    - Single Point Fault Metric
    - Latent Fault Metric
  - Hardware redundancy in structural modeling

	ASIL B	ASIL C	ASIL D
PVSG [1/h]	$<10^{-7}$ (recom.)	$<10^{-7}$	$<10^{-8}$
SPFM	>90%	>97%	>99%
LFM	>60%	>80%	>90%

Automotive Safety Integrity Levels

- ASIL: “One of four classes to specify the item's necessary safety requirements for achieving an acceptable residual risk with D representing the highest and A the lowest class”
- Approaches to determine the ASIL
  - Focus on qualitative methods: such as risk graph or hazardous event severity matrix → see next slide

Determining Required ASIL

Classes of severity	Classes of probability of exposure regarding operational situations	Classes of controllability (by driver)		
		C1 (simple)	C2 (normal)	C3 (difficult, uncontrollable)
S1 Light and moderate injuries	E1 (very low)	QM	QM	QM
	E2 (low)	QM	QM	QM
	E3 (medium)	QM	QM	A
	E4 (high)	QM	A	B
S2 Severe and life threatening injuries (survival probable)	E1 (very low)	QM	QM	QM
	E2 (low)	QM	QM	A
	E3 (medium)	QM	A	B
	E4 (high)	A	B	C
S3 Life threatening injuries, fatal injuries	E1 (very low)	QM	QM	A
	E2 (low)	QM	A	B
	E3 (medium)	A	B	C
	E4 (high)	B	C	D

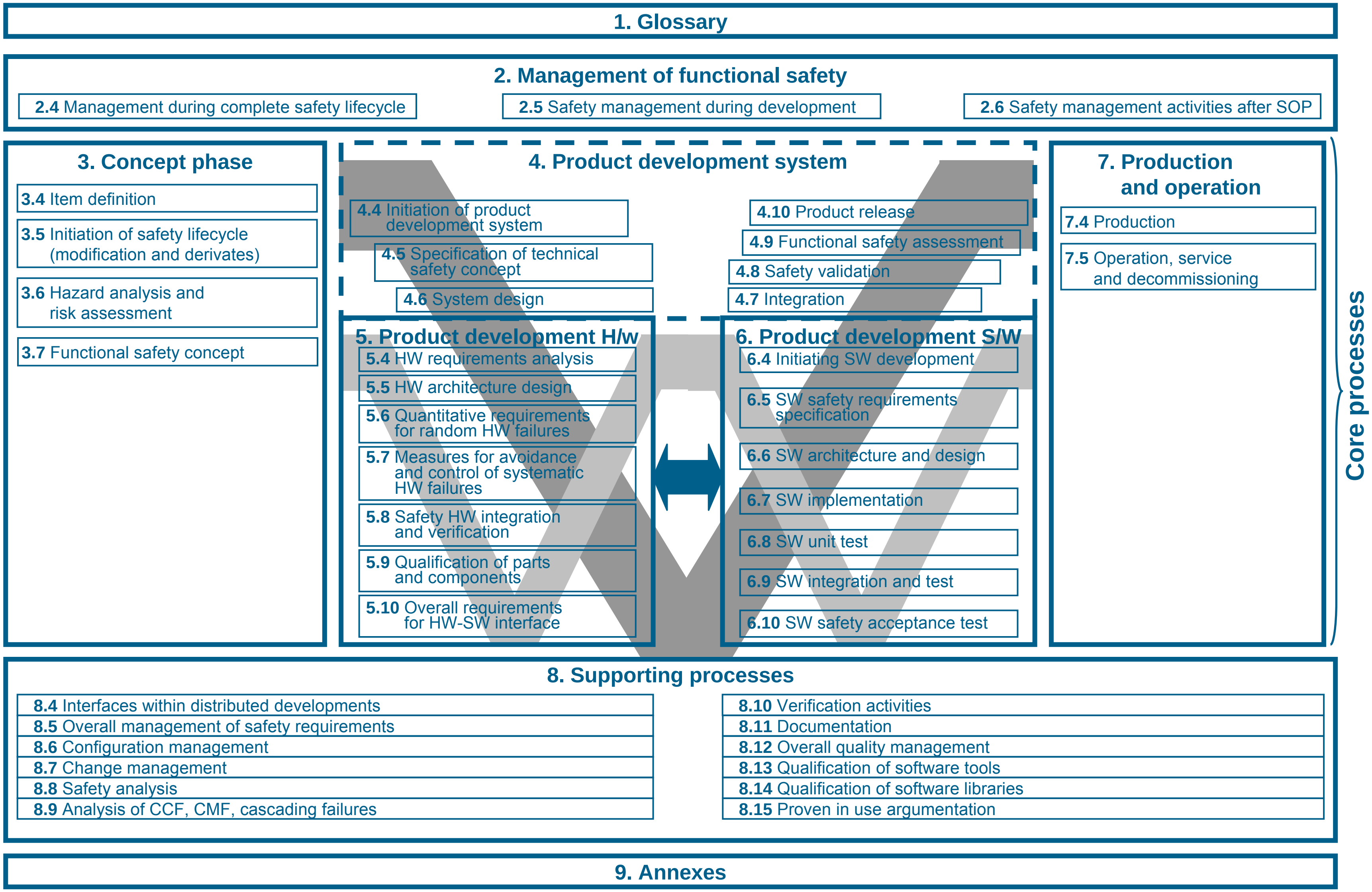
Quantitative Requirements ISO26262

- ISO 26262
  - Four Automotive SILs (ASIL)
  - Three key metrics
    - Probability of violation of safety goals (PVSG)
    - Single Point Fault Metric
    - Latent Fault Metric
  - Hardware redundancy in structural modeling

	ASIL B	ASIL C	ASIL D
PVSG [1/h]	$<10^{-7}$ (recom.)	$<10^{-7}$	$<10^{-8}$
SPFM	>90%	>97%	>99%
LFM	>60%	>80%	>90%

Key Metrics

- Probability of violation of safety goals
  - Equivalent to PFH in IEC61508
- Single Point Fault Metric
  - Quantifies how many potentially immediately dangerous faults are either safe or detected
- Latent Fault Metric
  - Quantifies how many potentially dangerous faults that not yet influence the application are either safe or detected → under discussion, consult standard!



1. Glossary

2. Management of functional safety

- 2.4 Management during complete safety lifecycle
- 2.5 Safety management during development
- 2.6 Safety management activities after SOP

3. Concept phase

- 3.4 Item definition
- 3.5 Initiation of safety lifecycle (modification and derivates)
- 3.6 Hazard analysis and risk assessment
- 3.7 Functional safety concept

4. Product development system

- 4.4 Initiation of product development system
- 4.5 Specification of technical safety concept
- 4.6 System design
- 4.10 Product release
- 4.9 Functional safety assessment
- 4.8 Safety validation
- 4.7 Integration

5. Product development H/w

- 5.4 HW requirements analysis
- 5.5 HW architecture design
- 5.6 Quantitative requirements for random HW failures
- 5.7 Measures for avoidance and control of systematic HW failures
- 5.8 Safety HW integration and verification
- 5.9 Qualification of parts and components
- 5.10 Overall requirements for HW-SW interface

6. Product development S/W

- 6.4 Initiating SW development
- 6.5 SW safety requirements specification
- 6.6 SW architecture and design
- 6.7 SW implementation
- 6.8 SW unit test
- 6.9 SW integration and test
- 6.10 SW safety acceptance test

7. Production and operation

- 7.4 Production
- 7.5 Operation, service and decommissioning

Core processes

8. Supporting processes

- 8.4 Interfaces within distributed developments
- 8.5 Overall management of safety requirements
- 8.6 Configuration management
- 8.7 Change management
- 8.8 Safety analysis
- 8.9 Analysis of CCF, CMF, cascading failures
- 8.10 Verification activities
- 8.11 Documentation
- 8.12 Overall quality management
- 8.13 Qualification of software tools
- 8.14 Qualification of software libraries
- 8.15 Proven in use argumentation

9. Annexes

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)
- Hardware redundancy in formulas (HFT)

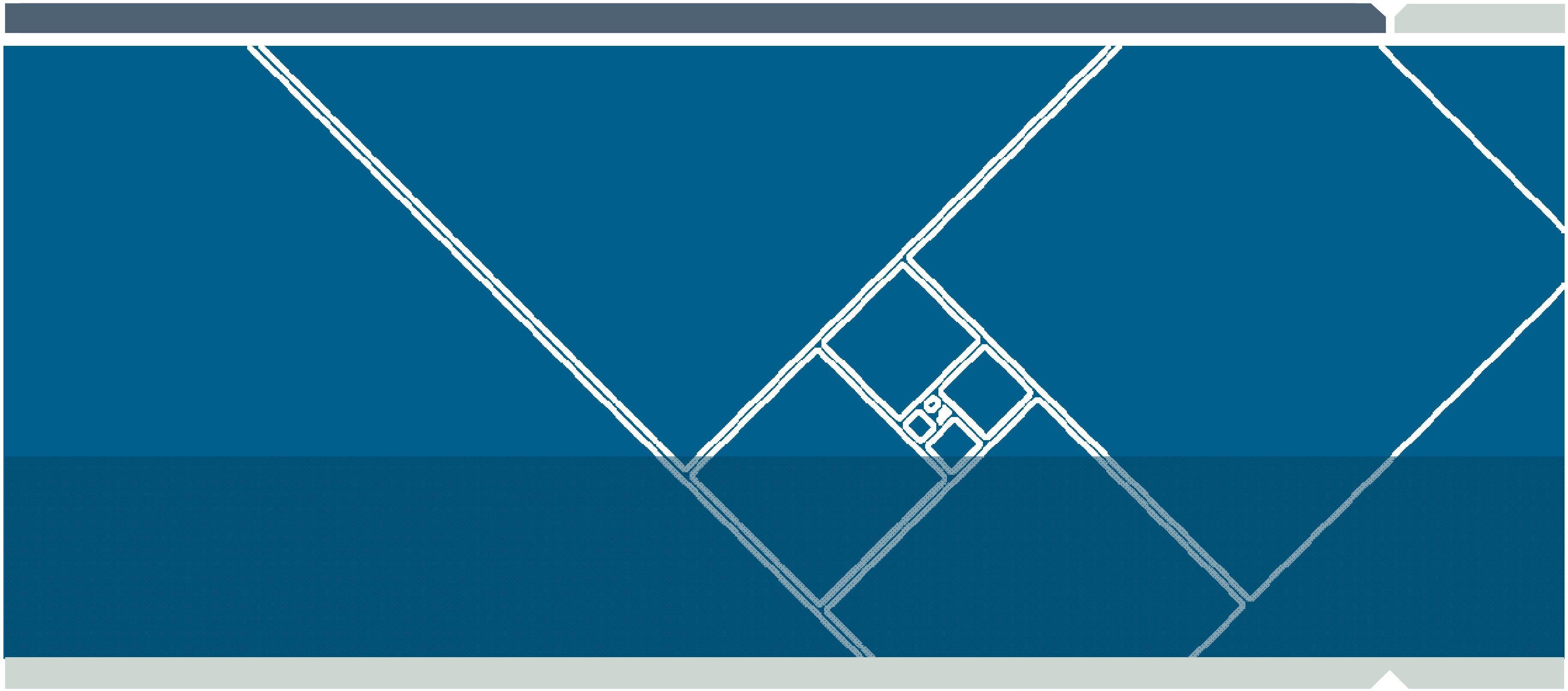
	SIL 1	SIL 2	SIL 3
PFH [1/h]	$<10^{-5}$	$<10^{-6}$	$<10^{-7}$
SFF (HFT=0)	>=60%	>=90%	>=99%
SFF (HFT=1)	-	>=60%	>=90%

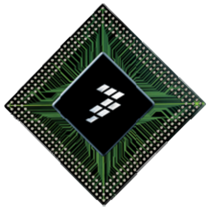
Note: Table adopted for typical automotive application

► ISO 26262

- Four Automotive SILs (ASIL)
- Three key metrics
  - Probability of violation of safety goals (PVSG)
  - Single Point Fault Metric
  - Latent Fault Metric
- Hardware redundancy in structural modeling

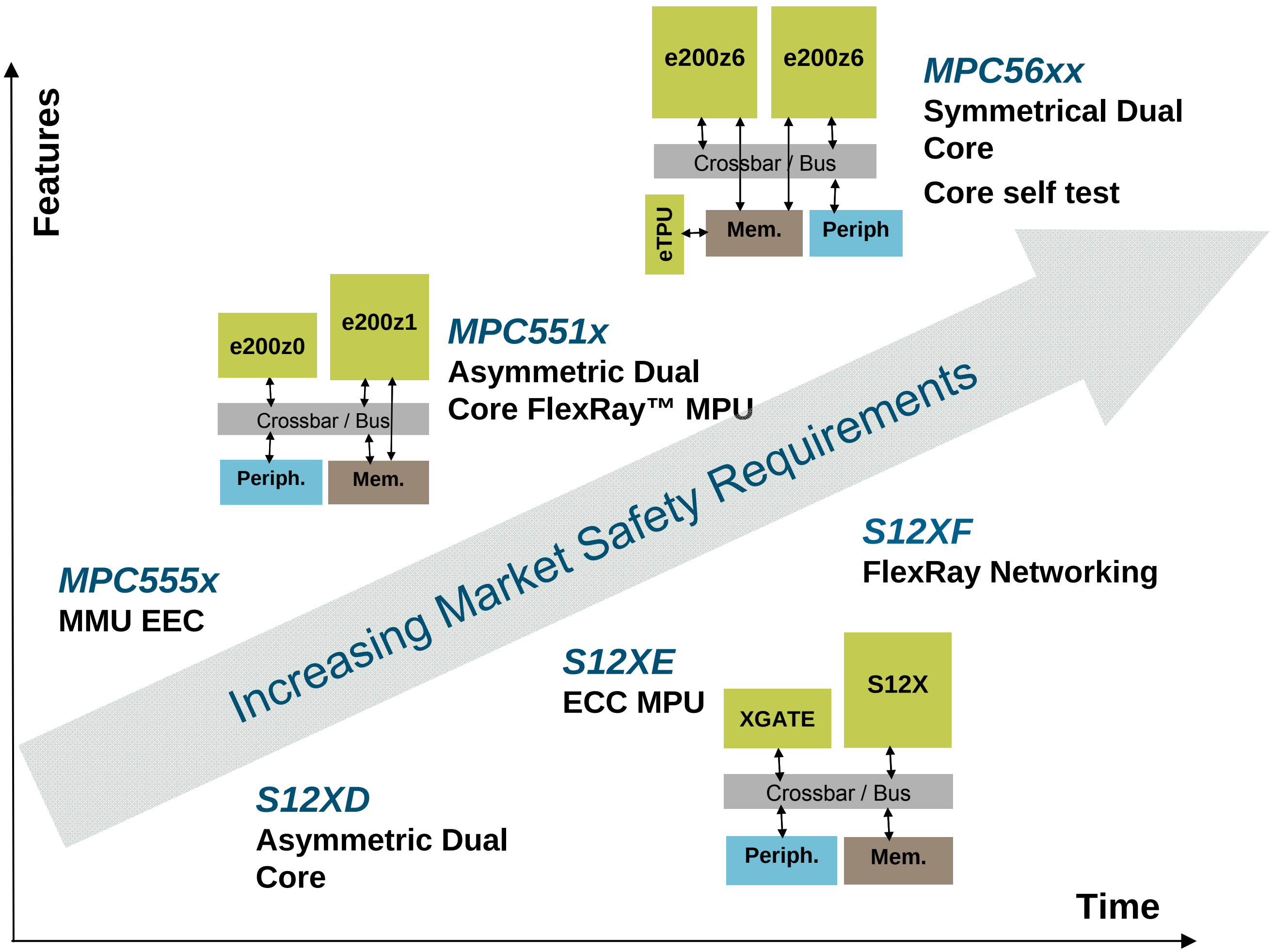
	ASIL B	ASIL C	ASIL D
PVSG [1/h]	$<10^{-7}$ (recom.)	$<10^{-7}$	$<10^{-8}$
SPFM	>90%	>97%	>99%
LFM	>60%	>80%	>90%





MCU Safety Continuum

Integrated Safety Features



MPC56xx
Symmetrical Dual Core

Core self test

MPC551x
Asymmetric Dual Core FlexRay™ MPU

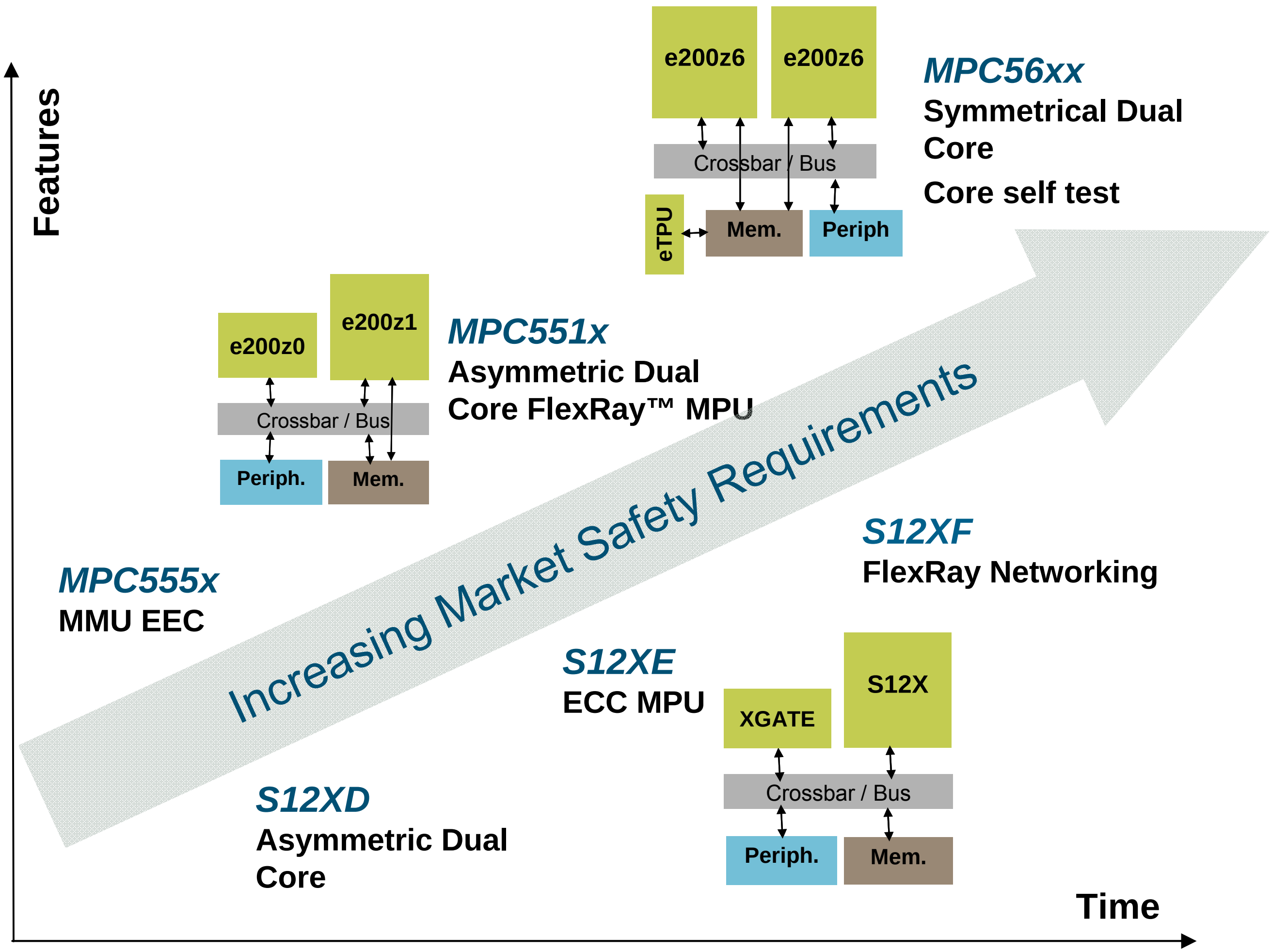
MPC555x
MMU EEC

S12XF
FlexRay Networking

S12XE
ECC MPU

S12XD
Asymmetric Dual Core

Integrated Safety Features



MPC56xx
Symmetrical Dual Core
Core self test

MPC551x
Asymmetric Dual Core FlexRay™ MPU

MPC555x
MMU EEC

S12XF
FlexRay Networking

S12XE
ECC MPU

S12XD
Asymmetric Dual Core

Integrated safety

- Fail safe MCUs
- Fail operational MCUs

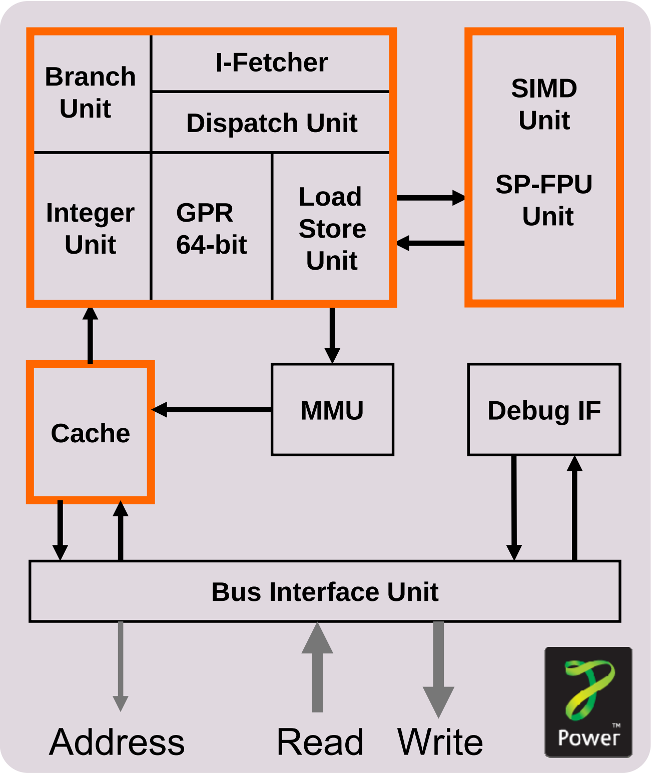
Safety Properties

- Transient fault detection
- Early detection of permanent faults
- Detection of systematic software faults

System Properties

- Cost benefit
- Low complexity
- High availability

Processor Core — Performance



Example: Freescale e200 core family built on Power Architecture® technology

►Example: Increased pipeline depth

- Typically 7-stages+ pipeline architectures allows more instructions per clock cycle
- Most instructions provide single cycle execution
- Integer and floating point multiply and multiply-accumulate in three clocks, fully pipelined

►Example: Dual instruction issue

- Two execution units allow parallel processing of instructions

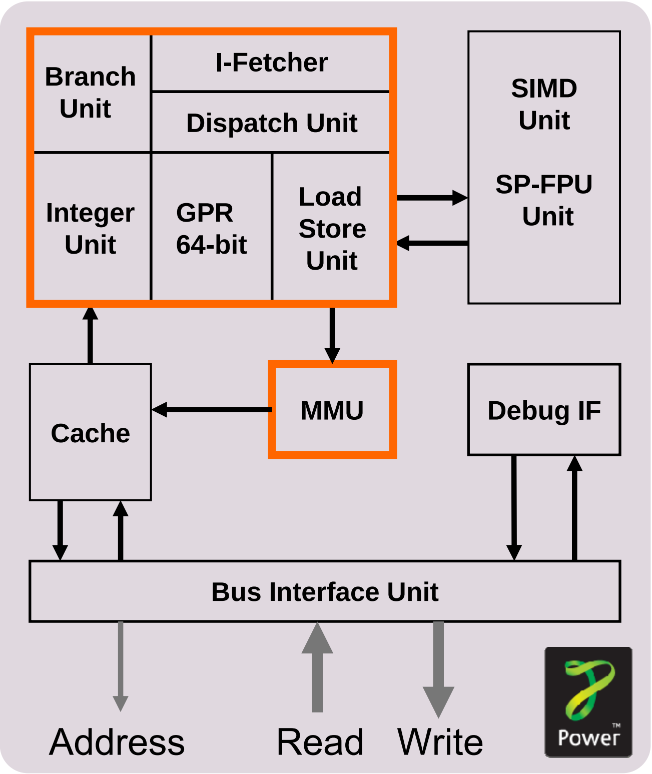
►Example: Instruction and data cache

- I-cache to speed up executable instruction fetch
- D-cache to speed up data fetch and store
- TLB to improve the speed of virtual address translation

►Example: SIMD unit and FPU

- Provides DSP capabilities
- Executes an operation on two separate sets of data

Processor Core — Safety



Example: Freescale e200 core family built on Power Architecture® technology

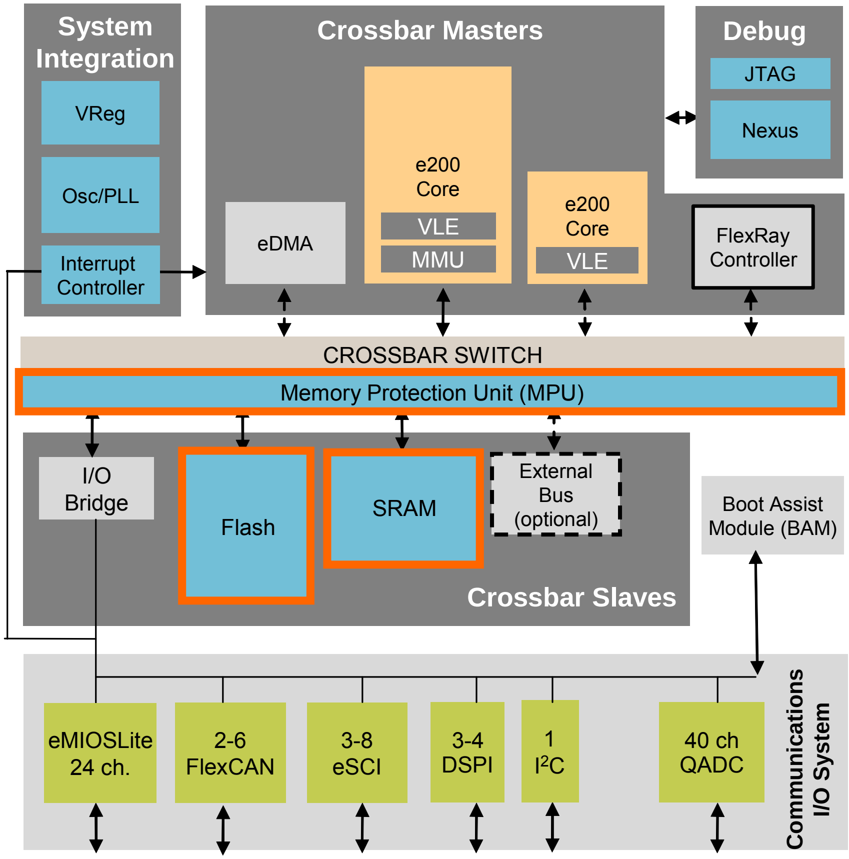
►Example: Memory management unit (MMU)

- Optimization of self test coverage by using different virtual adresses without relocating customer application data and code
- MMU can be used to protect accesses due to occurence of faults in the core (exception generation)

►Example: Multiple input shift register (MISR)

- Method for verifying all intermediate results of a set of architected registers at the end of an instruction stream
- Introduction of MISR improves observability of the core resulting in:
  - Increased self test coverage
  - Faster detection of dormant faults

Memories and Crossbar — Safety



Example: Typical 32-bit MPC55/56xx processor

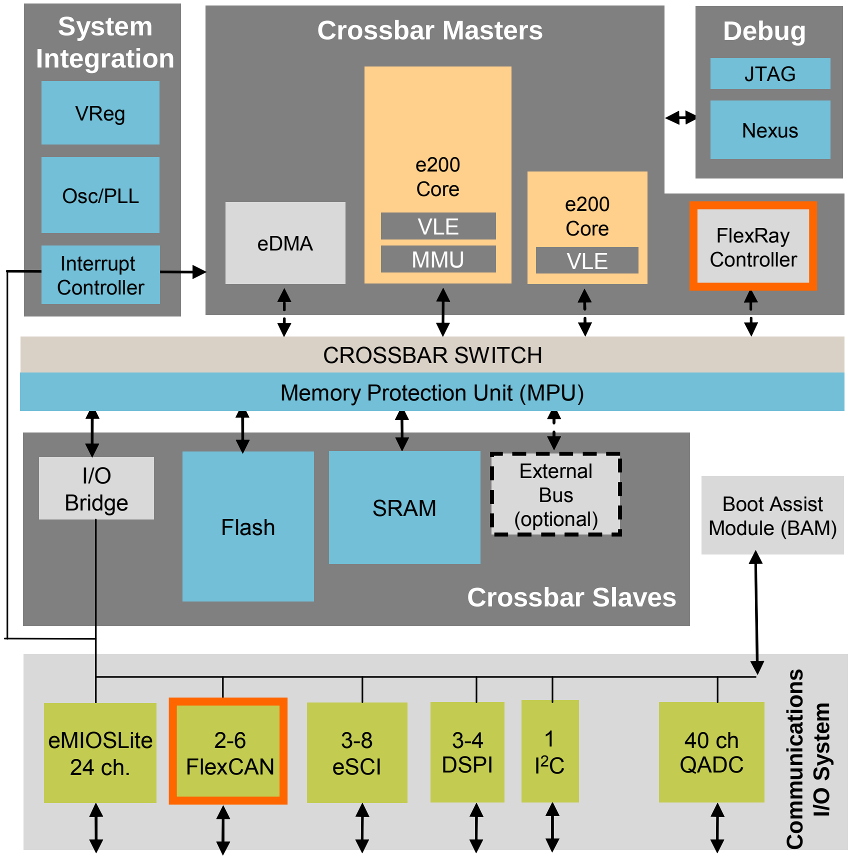
►Example: Memory protection unit

- Monitors all system bus transactions and evaluates the appropriateness of each transfer
- Pre-programmed region descriptors define memory spaces and associated access rights
- Unmapped references are terminated with a protection error response

►Example: Error-correcting code

- Used to detect failures of flash/SRAM stored data
- Typical solution for correcting bitflips caused by soft error rate (SER) impact
- ECC module (64 data bits + 8 ECC bits) can:
  - Correct all single bit errors
  - Detect all dual bit faults
  - Detect several faults affecting >2 bits

Communication — Safety



Example: Typical 32-bit MPC55/56xx processor

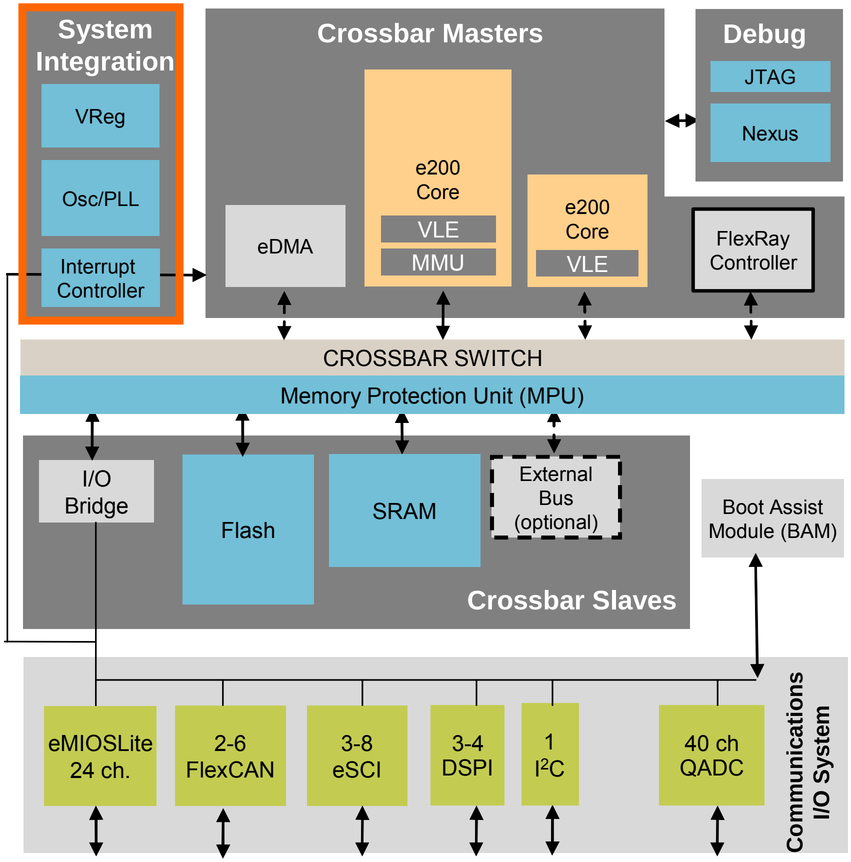
▸Example: FlexRay™ networking

- FlexRay master controller directly linked to the crossbar
- Replicated transmission of safety relevant data by single/dual channel FlexRay support with 2.5, 5 and 10 MBit/s data rates
- Message buffer stored and protected in dedicated memory partition located in system memory

▸Example: Safety port

- Controller area network (CAN)-type interface supporting high bandwidth for fast MCU-MCU communication
- Bit rate up to 7.5 Mbit/s
- 32 message buffers of 0 to eight bytes data length

Power Supply and Clock — Safety



Example: Typical 32-bit MPC55/56xx processor

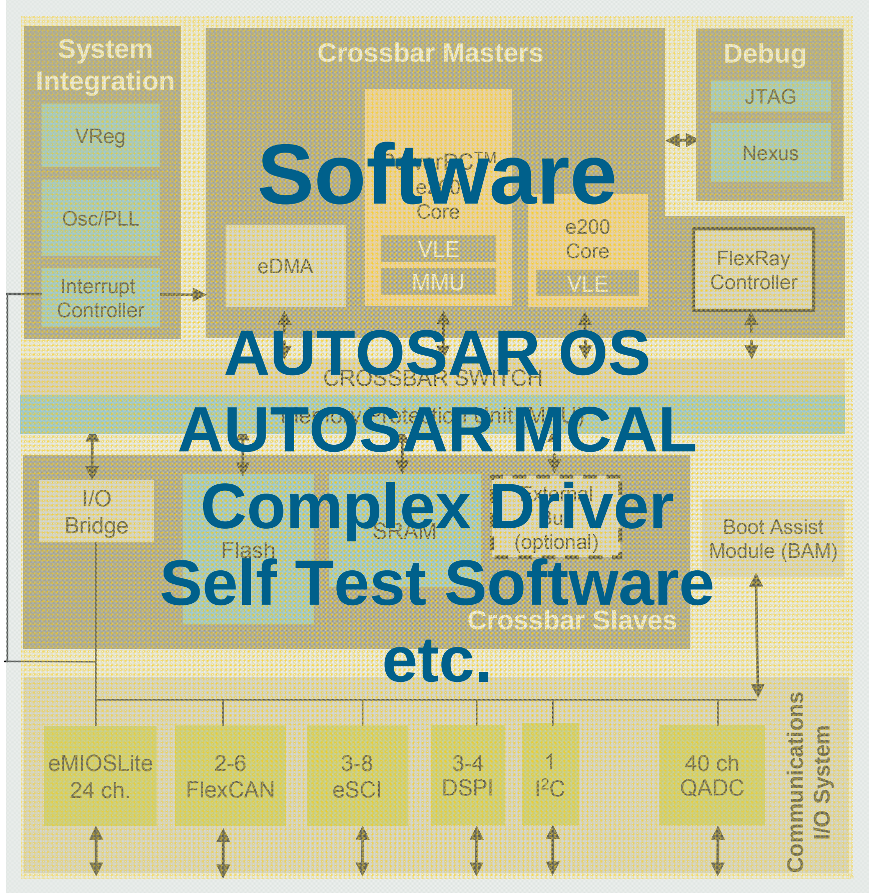
►Example: Power supply

- Monitoring of internal and external voltages — internal and external power supply
- Over- and undervoltage detection
- Testing capability of monitoring circuitry — e.g., for detection of dormant faults

►Example: Clock and monitoring

- Clock monitoring for system and periphery clock:
  - Loss of crystal or PLL clock
  - PLL frequency higher/lower than reference
- Redundant clock generation with internal RC oscillator
- Glitch filtering with on-chip PLL

Software – Safety



Software

AUTOSAR OS

AUTOSAR MCAL

Complex Driver

Self Test Software

etc.

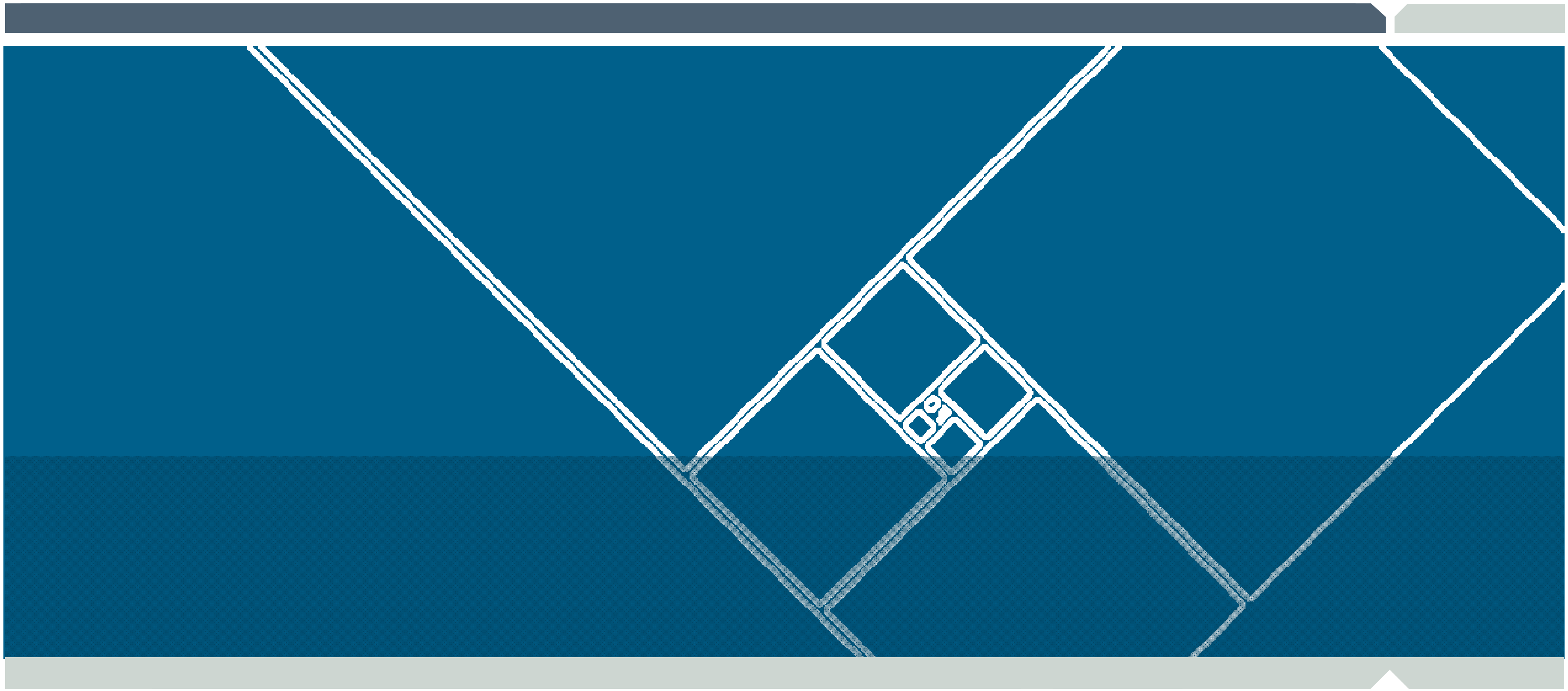
Example: Typical 32-bit MPC55/56xx processor

▶ Example: Core self test — basic

- Coverage: instruction-set based, all addressing modes
- Integration: mostly interruptible, low integration effort
- Safety: not fault graded, determined behavior in fault-free case
- For PPC instruction set

▶ Example: Core self test – advanced

- Coverage: stuck-at fault model, based on physics of failure
- Integration: partly interruptible, can be adjusted to application/OS specifics
- Safety: detailed test coverage provided, fault graded, determined behavior in fault-free and faulty case
- For selected PPC devices



Basic Core Self-Test

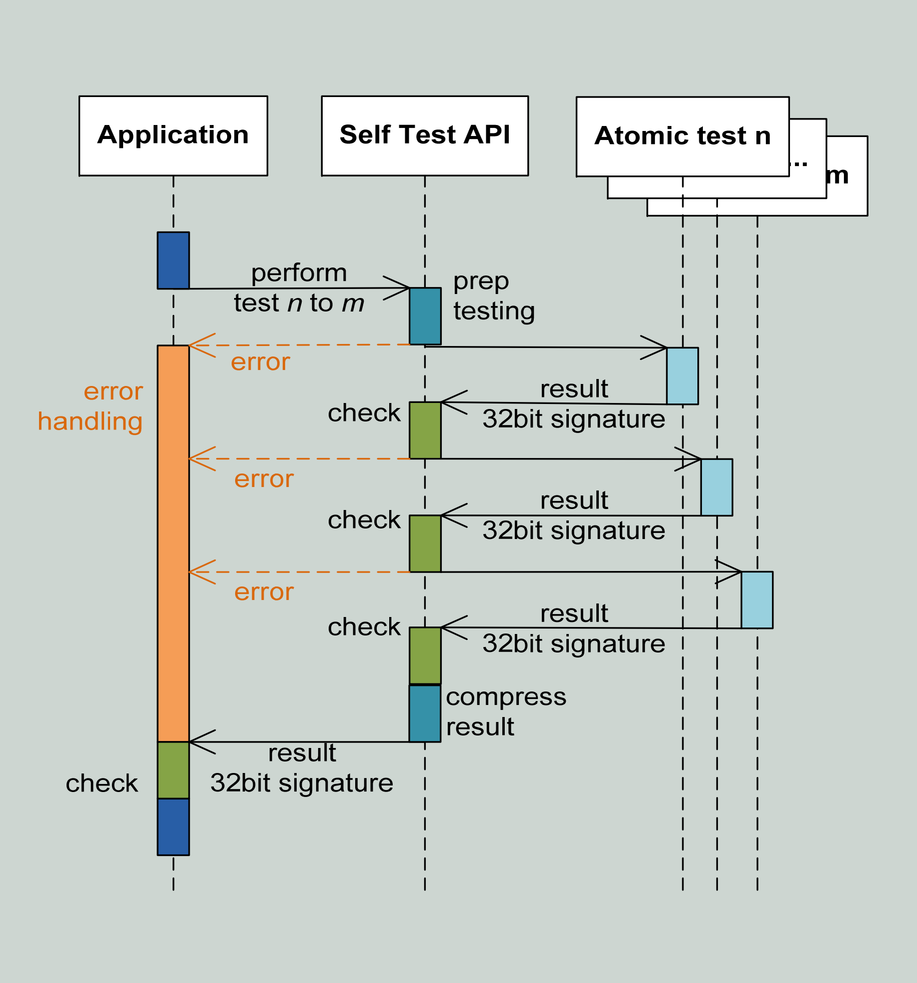
CST with Instruction Coverage Metric

		Instruction sets		
		BookE instructions	VLE instructions	SPE instructions
Instruction coverage*		~83% to ~98%	~86% to ~98%	Estimated 85% to 99%
Code size (bytes)		< 10k	< 5k	In development
Execution time (clock cycles)		< 6000	< 5000	In development
Supported PPC Cores	Z6	Supported	Supported	In development
	Z3	Supported	Supported	In development
	Z1	Supported	Supported	Not applicable
	Z0	Not applicable	Supported	Not applicable

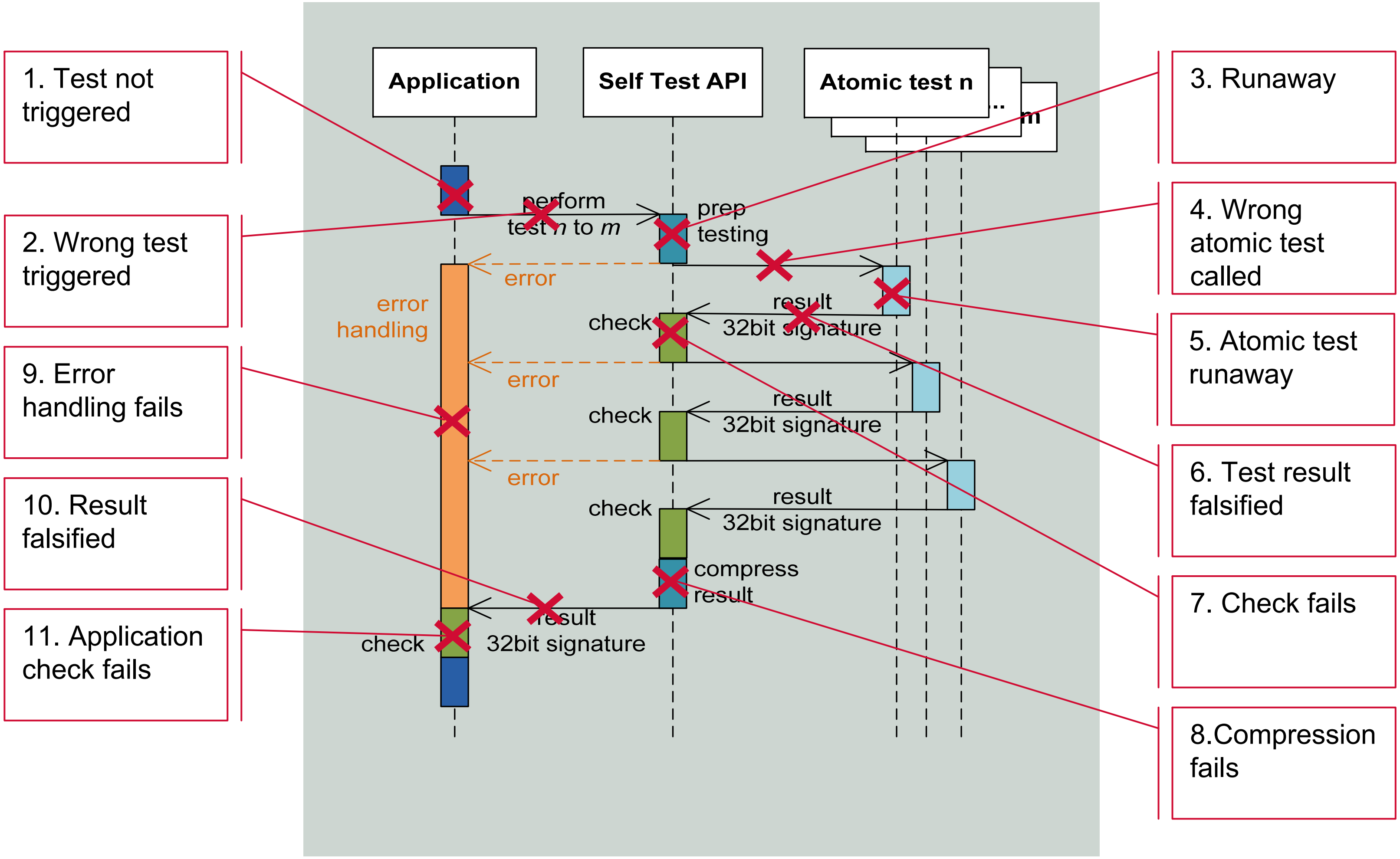
* Variability caused by whether instructions or operations (performed by instructions) are considered, and whether MMU and cache configuration instructions/operations are taken into account or not

Basic Operating Principle

- Application
  - Triggers test execution
  - Selects subset of tests to perform
  - Checks actual versus expected result
- Self test API
  - Saves application context
  - Prepares core and device for testing
  - Calls atomic tests
  - Checks results
  - Restores application context
  - Compresses atomic test results into one 32-bit signature
- Atomic test
  - Short piece of assembly code
  - Optimizes to activate and propagate faults in different core modules



Potential Issues beyond the Self-Test Software



1. Test not triggered

2. Wrong test triggered

9. Error handling fails

10. Result falsified

11. Application check fails

3. Runaway

4. Wrong atomic test called

5. Atomic test runaway

6. Test result falsified

7. Check fails

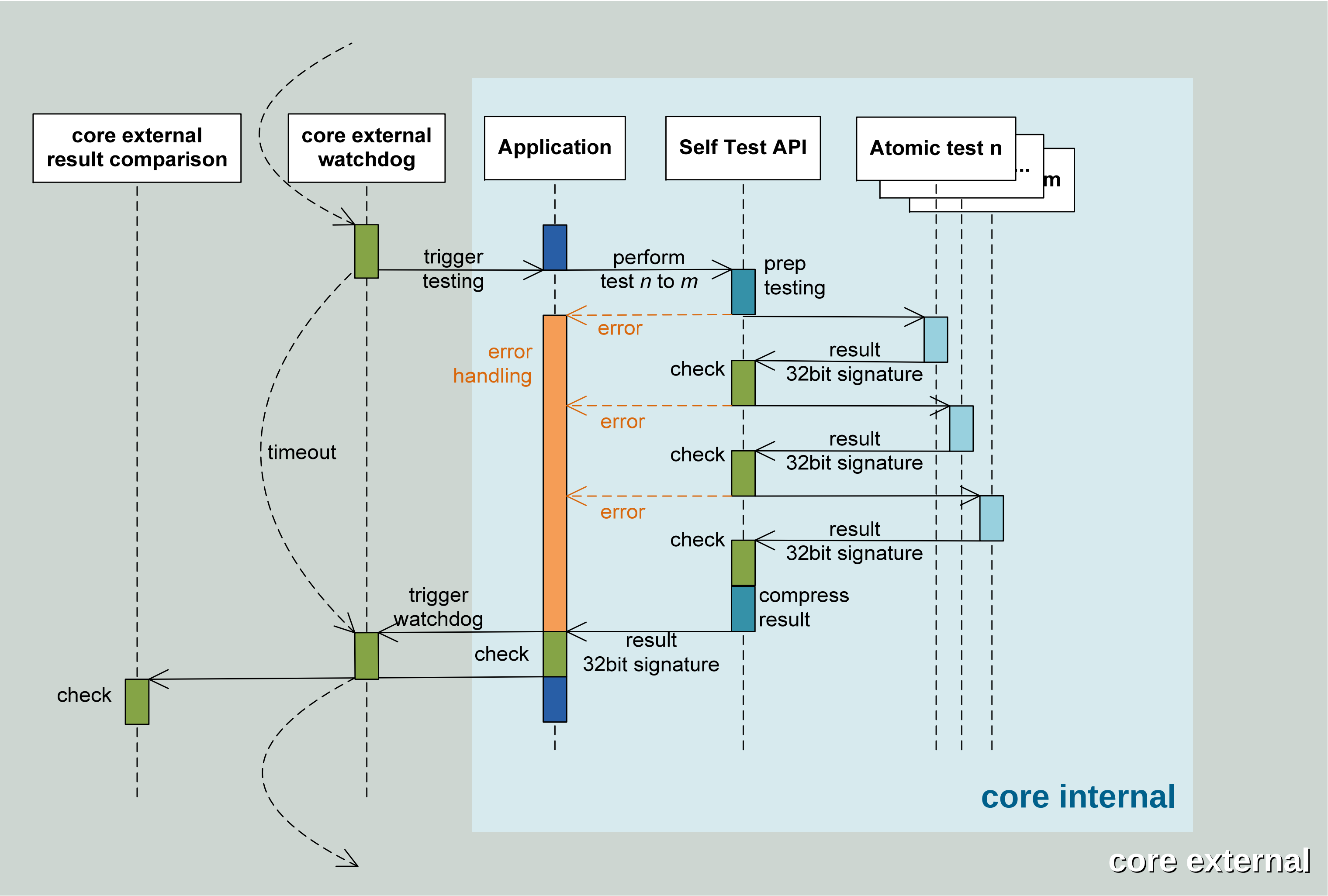
8.Compression fails

Mitigation Measures

		Can be caught by		
		Basic Watchdog	Intelligent watchdog	Application check and signature
1	Test not triggered	✓		
2	Wrong test triggered			✓
3	Runaway			✓
4	Wrong atomic test called			✓
5	Atomic Test Runaway	✓		
6	Test result falsified			✓
7	Check fails			✓
8	Compression fails			✓
9	Error handling fails		✓	
10	Result falsified			✓
11	Application check fails		✓	

- Watchdog and redundant result check
  - External to core
  - May be device internal, however (coprocessor, ETPU, etc.)
- Application check
  - Unique result for each atomic test

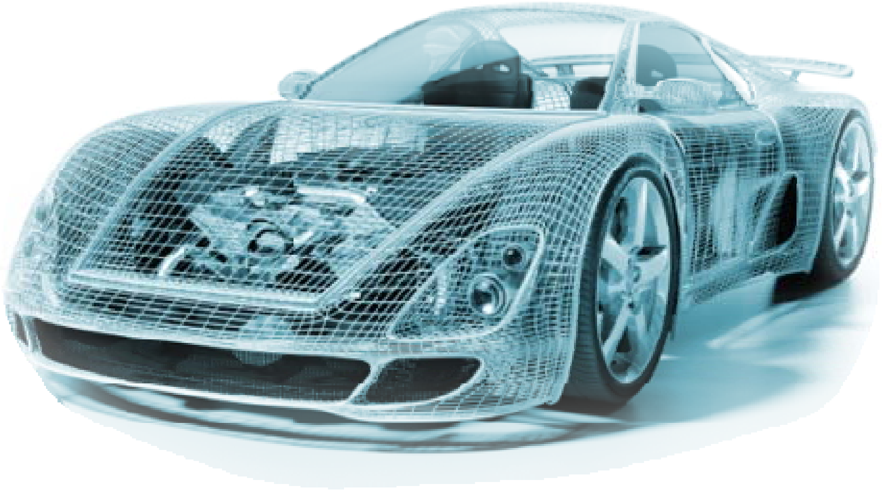
Overall Operating Principle



Summary

Summary

- Safety standards are becoming key for the design of new controller solutions and influence the architecture of virtually all building blocks
- Freescale sees safety, and in particular, functional safety as a key paradigm of next generation electronic vehicle systems
- Freescale is continuously expanding the product controller, analog and sensor portfolio to address the needs of these systems in line with IEC61508 and ISO26262



Q&A

- Thank you for attending this presentation. We’ll now take a few moments for the audience’s questions and then we’ll begin the question and answer session.

freescale™
semiconductor