

Integrating Functional Safety with ARM

November, 2015

Lifeng Geng, Embedded Marketing Manager

ARM: The World's Most Scalable Architecture

- ARM ecosystem meets needs of vertical markets – from sensors to servers
 - Addressing automotive, consumer, industrial, mobile, medical, metering and beyond
 - 12bn ARM chips shipped in 2014 alone – increasingly becoming connected as part of IoT
- ARM's market share at 37% overall

Year	Market Share
2007	17%
2008	20%
2009	22%
2010	25%
2011	29%
2012	32%
2013	35%
2014	37%



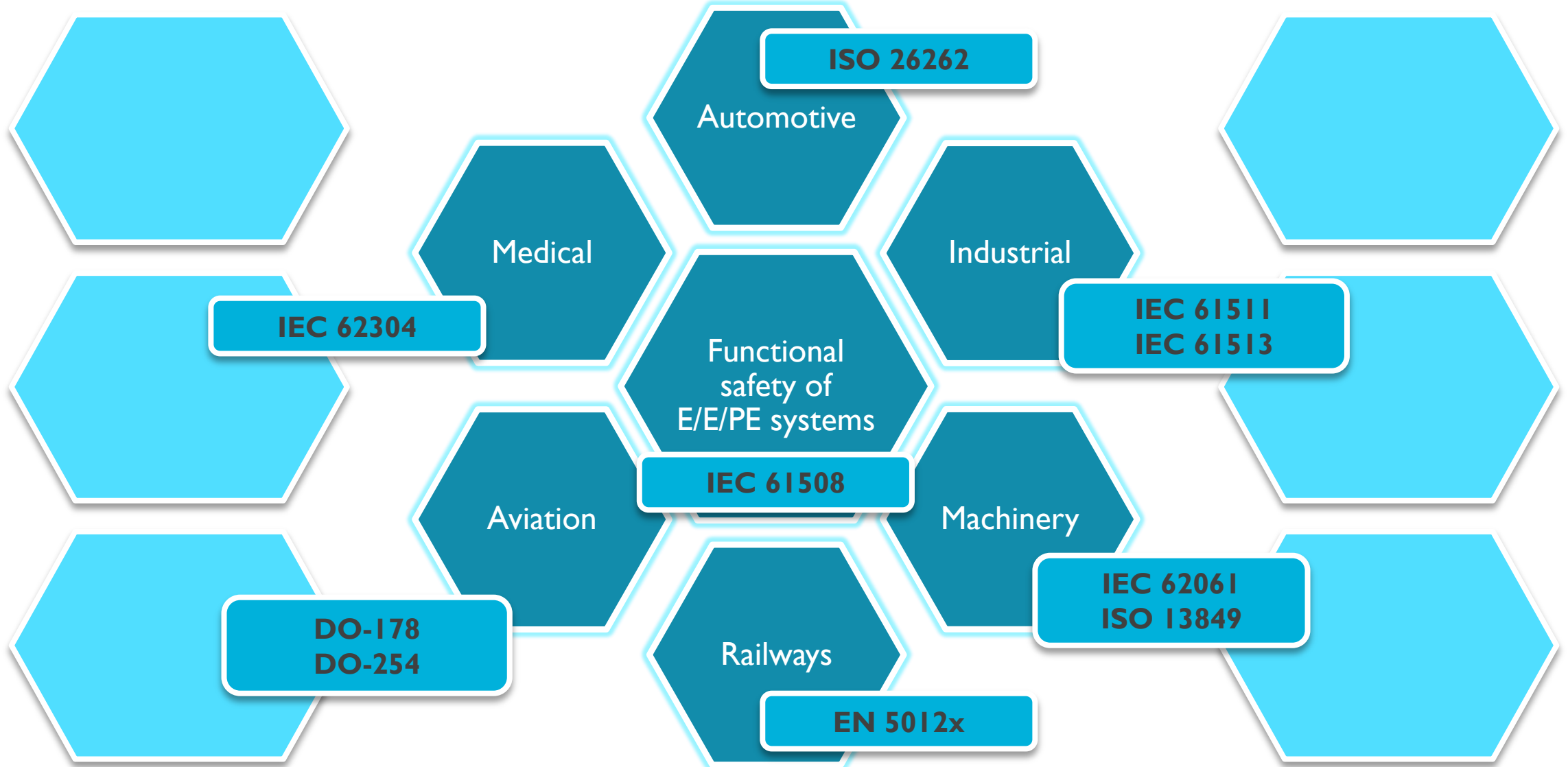
Functional Safety support is becoming essential

- Compliance with safety standards is required in many markets
- Visible reminders everywhere of the importance of electronics to automotive industry
- Also applies to other sectors: medical, factory automation, robotics, automotive, transport...

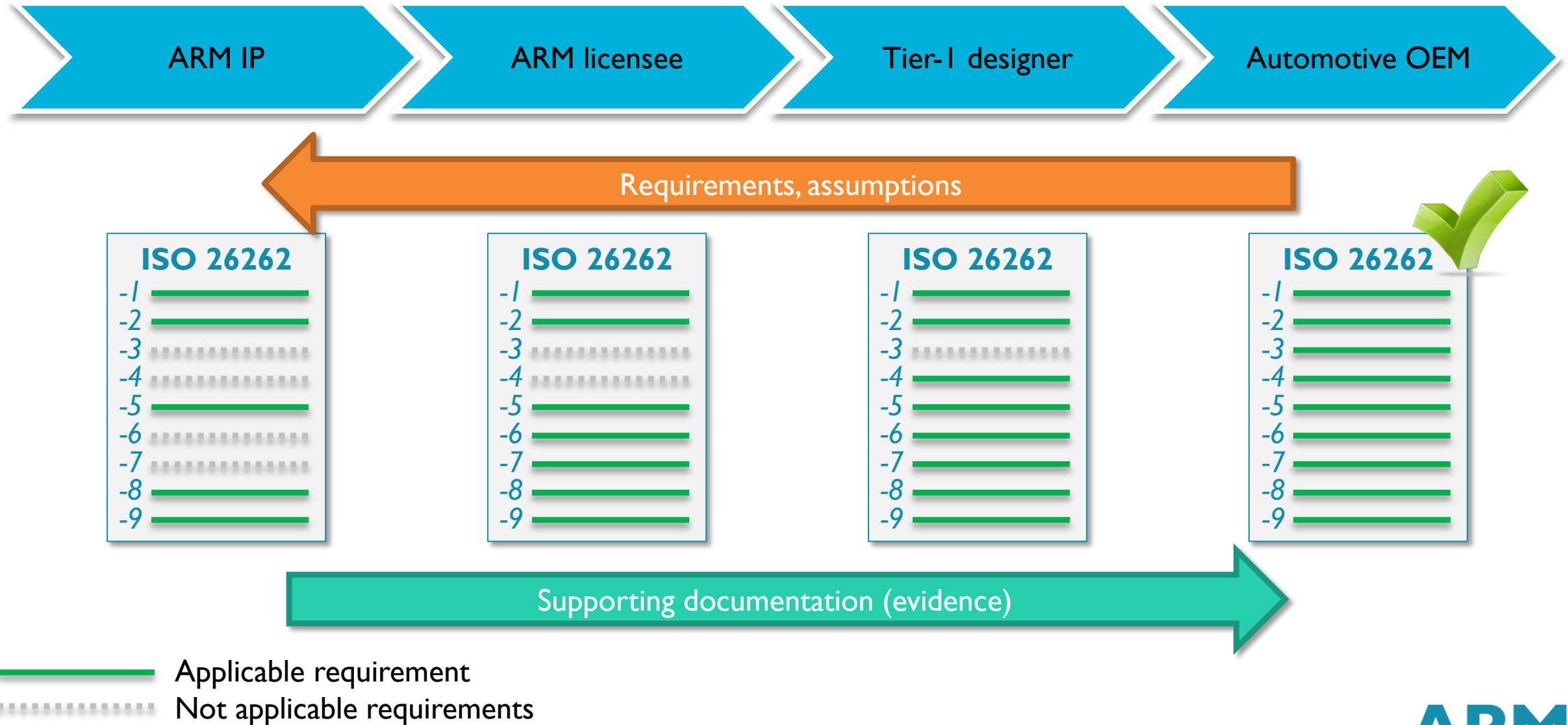


- ARM white papers provide more detail

Functional Safety – Standards



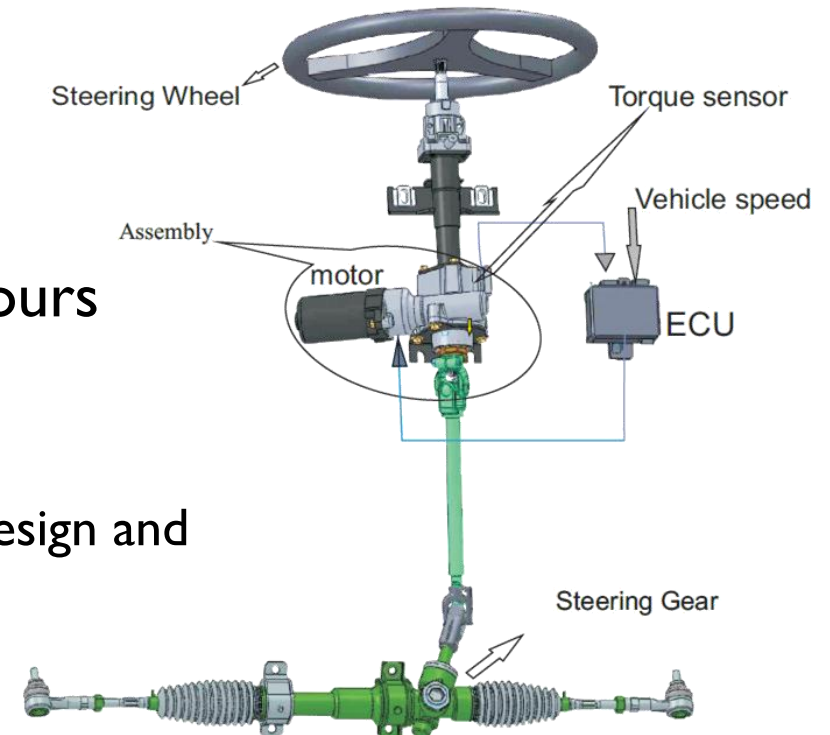
How the standard is being used in the industry?



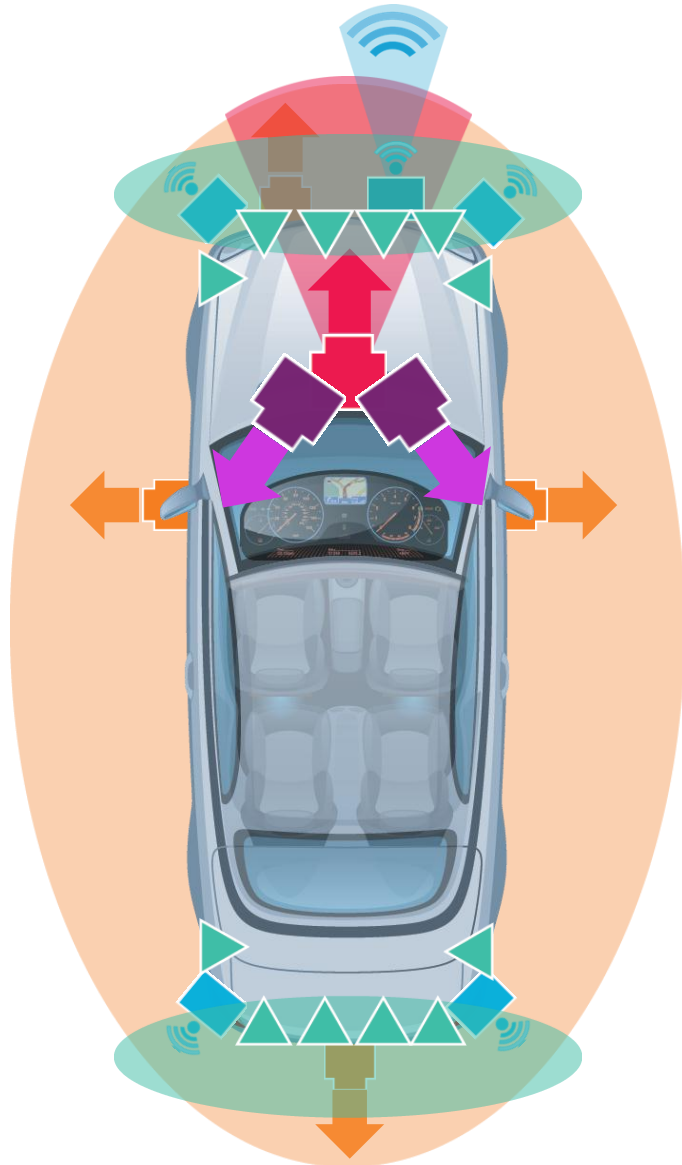
Functional Safety Example


Electric Power Steering

- An example of a control system which must demonstrate functional safety
 - Must continue to function or at least behave predictably in event of a fault
 - By predictable behaviour we mean it must shut down, fail safe, reset and restart etc.
- Functionally safe systems aim at preventing hazardous behaviour in event of a fault
- Level of risk resulting from potential malfunctioning behaviours is quantified through hazard analysis and risk assessment
 - Automotive Safety Integrity Levels range from ASIL A to ASIL D
 - The higher ASIL requirement dictates the level of robustness of design and verification processes, and often also leads to inclusion of more fault detection and control features




Another Example: ADAS Sensors and Functions




360 Surround
View

Back-up Camera
Automatic Parking
Object Detection


Front Camera
Mono or Dual


Adaptive Cruise
Automatic Braking
Smart Lighting
Object Recognition


Interior Cockpit
Camera


Drowsy Driver
Occupant Detection
Facial Recognition


Long-range
Radar

Collision Warning
Object Detection
Adaptive Cruise


Mid-range
Radar

Cross traffic warning
Object Detection

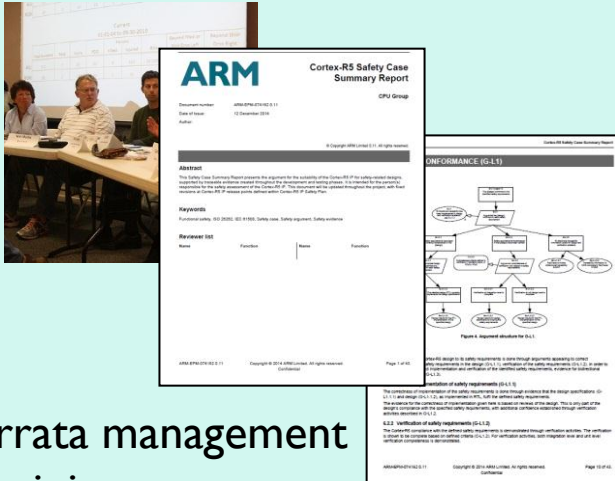

Ultrasonic
Sensors

Parking Assist
Blind spot detection

- Lots of sensors – cameras, radars, ultrasonic, and many more to come.
- Lots of opportunity for redundancy of functions
- Semi-autonomous driving can be achieved today with embedded control
- V2V and V2I will offer supplemental control from the cloud and greater redundancy
- Fail functional is needed for safety features.

Functional Safety Support for ARM IP

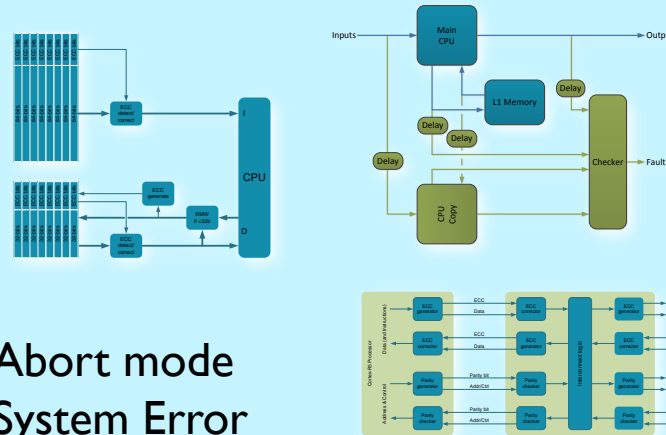
Safety management
Requirements management
Quality



Errata management
Training
Documentation

Processes

Fault detection/control features
Memory Protection
Error Correction
Dual Core Lock-Step



Abort mode
System Error
Fault containment

Design & Verification

ARM IP Product Safety Package *
Safety Manual
Failure Modes and Effects Analysis
Development Interface Report



Safety Package

* Supported IPs have separate licensable package

Functional Safety Support Levels

Standard level support

- Focus on systematic aspects
 - Design and verification description
 - FMEA Report with example quantitative analysis
- External fault detection and control mechanisms to ARM IP typically required
 - Software-based solutions
 - System-level solutions
- Example processors
 - Cortex-M0+, Cortex-M3, Cortex-M4
 - Cortex-A53, Cortex-A57, Cortex-A72

Extended level support

- Covers both systematic and random HW fault aspects
 - Robust fault detection and control mechanisms within the design
- External fault detection and control mechanisms not typically necessary
 - Dependent on overall system architecture
- Example processors
 - Cortex-R5
 - Cortex-M7

Levels of Support Explained

	Standard level	Extended level
Typical safety requirements	Up to ASIL B (ISO 26262) / SIL 2 (IEC 61508)	Up to ASIL D (ISO 26262) / SIL 3 (IEC 61508)
Target application areas	Monitoring, processing, analysis applications, e.g. ADAS, general process control	Real-time control applications, e.g. braking, EPS, industrial safety
ARM functional safety support documents	<ul style="list-style-type: none"> • Safety Manual • FMEA Report • Development Interface Report 	<ul style="list-style-type: none"> • Safety Manual • FMEA Report • Development Interface Report
FMEA format	Functional level analysis with estimated failure rate distribution	Detailed analysis with estimated failure rate distribution and diagnostic coverage
Fault detection and diagnostics within ARM IP	Limited or no diagnostic coverage achievable by hardware-only means. Additional diagnostics by system-level or software means	Typically very high diagnostic coverage achievable by hardware-only means

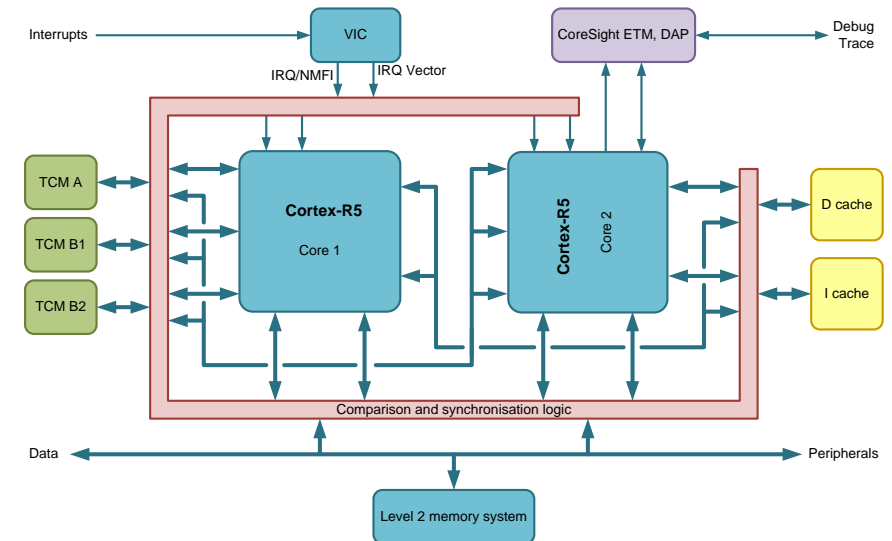
Fault Detection and Control Features

- Processor specific features
 - Typically redundant elements in the design
 - Not required for normal operation
 - Provide additional fault detection capability
 - Estimate of diagnostic coverage possible
- Examples
 - ECCs
 - Lock-step
- Architecture defined features
 - Applicable to all processors implementing the architecture
 - Generic in nature, with potentially lower fault detection capability
 - Estimation of diagnostic coverage difficult
- Examples
 - Exception handling
 - Memory protection and management

Example: Cortex[®]-R5 Fault Detection and Control

- Processor specific
 - TCM ECC
 - Cache ECC and parity
 - TCM external error
 - Bus protection
 - Dual core lock-step with delay

- ARMv7-R architecture based
 - Memory protection unit (MPU)
 - Exception model

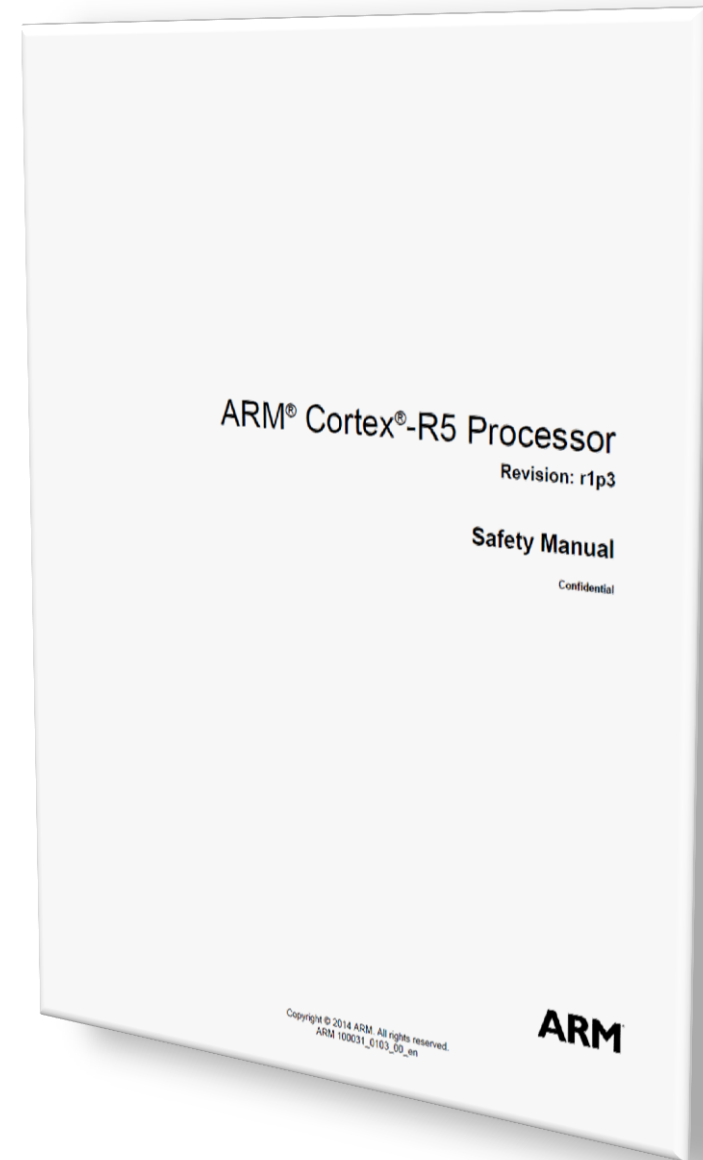


Safety Documentation Package Contents

- Essential documentation to support designing SoC / MCU products for safety-related markets
- Three key documents within the Safety Documentation Package:
 - Safety Manual
 - Overall description of functional safety activities within ARM
 - Product specific aspects of functional safety
 - Overview of safety architecture
 - Description of fault detection and control mechanisms
 - Summary of safety analysis results
 - Failure Modes and Effects Analysis Report
 - Block and sub-block level partitioning
 - Estimated failure rate distributions
 - Sample quantitative analysis
 - Development Interface Report
 - Identification of safety-lifecycle aspects applicable to ARM and IP integrator

Cortex[®]-R5 Safety Manual

- Contents at top level
 - Introduction
 - Cortex-R5 safety lifecycle
 - Cortex-R5 safety architecture
 - Cortex-R5 configuration options
 - Cortex-R5 fault detection and control mechanisms
 - Cortex-R5 assumptions of use
 - Cortex-R5 safety analysis results
 - Appendix – ECC tables
 - Appendix – Measures for systematic fault avoidance
 - Appendix – Lock-step initialization sequence
- Total contents about 150 pages



Cortex[®]-R5 Safety Manual

	Preface	
	About this book	7
	Feedback	10
Chapter 1	Introduction	
	1.1 Role of ARM IP in safety context	1-12
	1.2 Intended use of this document	1-14
Chapter 2	Cortex-R5 Processor Safety Lifecycle	
	2.1 About the Cortex-R5 processor safety lifecycle	2-17
	2.2 Overall functional safety management	2-18
	2.3 Project specific functional safety management	2-19
	2.4 Functional safety audits	2-23
	2.5 Functional safety assessments	2-24
Chapter 3	Cortex-R5 Processor Safety Architecture Overview	
	3.1 About the Cortex-R5 processor safety architecture	3-26
	3.2 Single core configuration	3-30
	3.3 Dual core lock-step configuration	3-31
Chapter 4	Cortex-R5 RTL Configuration Options	
	4.1 About Cortex-R5 configuration options	4-33
	4.2 RTL configuration for internal TCM ECC	4-34
	4.3 RTL configuration for cache ECC	4-36
	4.4 RTL configuration for cache parity	4-37
	4.5 RTL configuration for TCM external error	4-38
	4.6 RTL configuration for L2 AMBA bus diagnostics	4-39
	4.7 RTL configuration for Memory Protection Unit	4-40
	4.8 RTL configuration for exceptions	4-41
	4.9 RTL configuration for lock-step	4-42
	4.10 RTL configuration for split/lock	4-43
Chapter 5	Cortex-R5 Processor Fault Detection and Control Mechanisms	
	5.1 About Cortex-R5 fault detection and control mechanisms	5-45
	5.2 Internal TCM ECC	5-46
	5.3 Cache ECC	5-57
	5.4 Cache parity	5-69
	5.5 TCM external error	5-77
	5.6 L2 AMBA bus diagnostics	5-82
	5.7 Memory Protection Unit	5-95
	5.8 Exceptions	5-100
	5.9 Lock-step	5-109
	5.10 Split/lock	5-115
Chapter 6	Cortex-R5 Processor Assumptions of Use	
	6.1 About the assumptions of use	6-120
	6.2 Assumptions of use for the system integrator	6-121
	6.3 Assumptions of use for the system developer	6-123
Chapter 7	Cortex-R5 Processor Safety Analysis Results	
	7.1 About safety analysis results	7-126
	7.2 Failure modes and effects analysis	7-127
	7.3 Sample core implementation results	7-130
	7.4 Dependent failures	7-133
	7.5 Systematic faults	7-134
	7.6 Security considerations	7-135
Appendix A	ECC Encoding Tables	
	A.1 Introduction	Appx-A-137
	A.2 64-bit ECC scheme	Appx-A-138
	A.3 32-bit ECC scheme	Appx-A-140
	A.4 3-bit ECC scheme	Appx-A-141
Appendix B	Measures for Systematic Fault Avoidance	
	B.1 Systematic fault avoidance measures	Appx-B-143
Appendix C	Suggested Initialization Code for Lock-Step Operation	
	C.1 Suggested initialization code for lock-step operation	Appx-C-149
Appendix D	Revisions	
	D.1 Revisions	Appx-D-152

Cortex[®]-R5 Safety Manual

- Safety lifecycle description
- Overall and product specific safety management
 - Lifecycle aspects
 - V&V activities
 - Supporting processes
- Functional safety audits and assessments
 - Description of planned and completed activities
 - Summary of findings

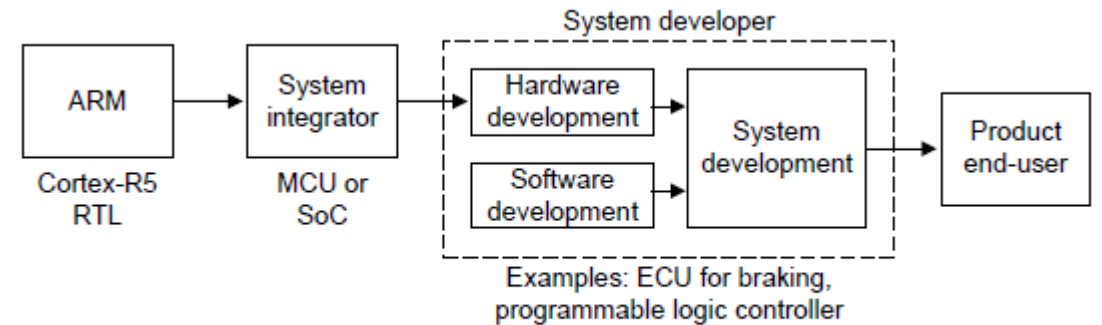


Figure 1-1 Allocation of roles and responsibilities

Cortex[®]-R5 FMEA Report

- General structure
 - Description of contents
 - ARM IP partitioning for safety analysis
 - Summary results of safety-related metrics
 - Example quantitative FMEA analysis

- Designed for usability
 - Standard Excel workbook
 - Fully modifiable / customizable by licensee
 - No macros required

- Complemented with an application note
 - Detailed description of analysis method

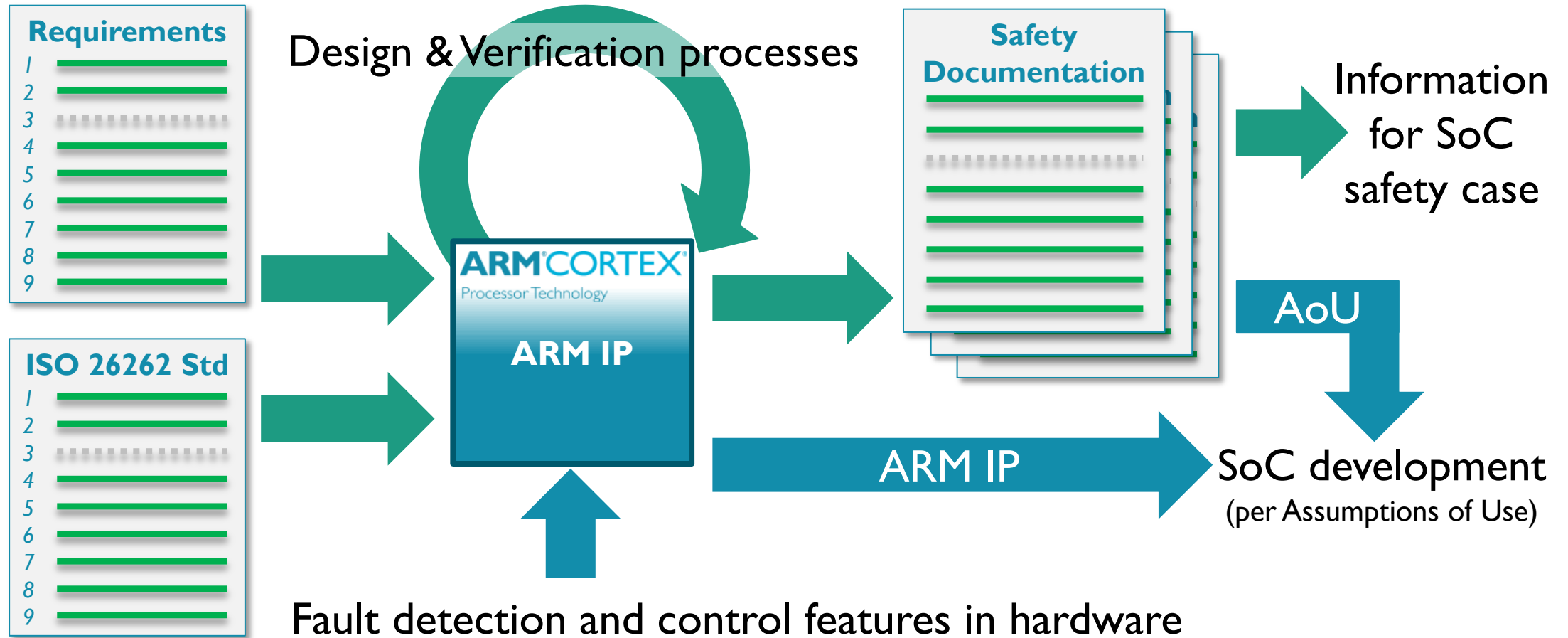
Component level	Block level	Sub-block level (end point)	Safety Relevant	Detailed fault
CPU	Cortex R5 CPU0 (Master) Processor Core	Core Prefetch Unit	1	Permanent fault causing wrong instruction load opcode feeding the processor
			1	Transient fault causing wrong instruction load opcode feeding the processor
CPU	Cortex R5 CPU0 (Master) Processor Core	Core Prefetch Unit	1	Permanent fault in the taken/not-taken decision
			1	Transient fault in the taken/not-taken decision

Symbol
$\lambda (\lambda_{SR} + \lambda_{NSR})$
λ_{SR}
λ_{NSR}
λ_S
Residual and single point faults $\lambda_{RF} + \lambda_{SPF}$
Multiple point faults λ_{MPF}
Latent multiple point faults λ_{MPFL}
Architecturally safe faults (by F_{safe}) λ_{S-ARCH}
Transient faults
Overall failure rate: $\lambda (\lambda_{SR} + \lambda_{NSR})$

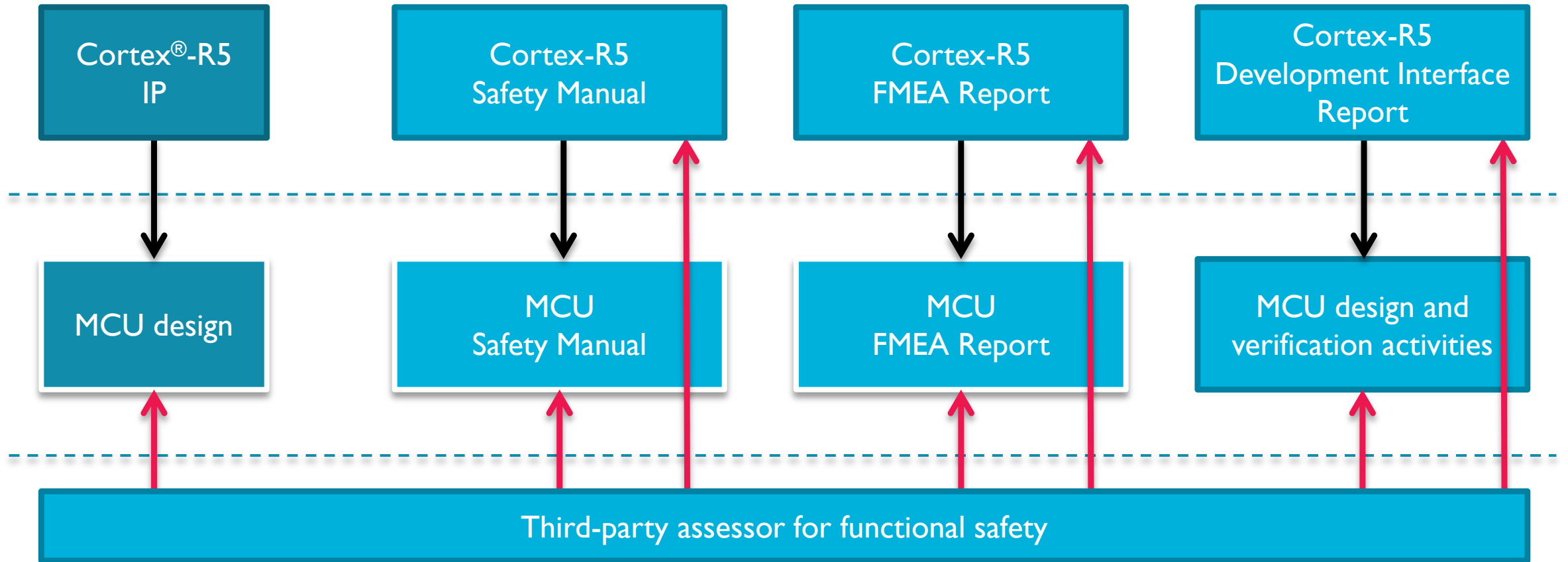
Development Interface Report – Contents

	Preface	
	<i>About this book</i>	7
	<i>Feedback</i>	10
Chapter 1	Introduction	
	1.1 <i>About this book</i>	1-12
	1.2 <i>Purpose</i>	1-13
	1.3 <i>Role of ARM IP in a safety context</i>	1-14
	1.4 <i>Assignment of roles and responsibilities</i>	1-15
	1.5 <i>Intended use of the Development Interface Report</i>	1-16
Chapter 2	Development Interface for the ARM® Cortex®-R5 Processor	
	2.1 <i>Communication</i>	2-18
	2.2 <i>Safety management activities</i>	2-19
	2.3 <i>Technical aspects</i>	2-20
	2.4 <i>Supporting documentation and deliverables</i>	2-21
Appendix A	Summary of the Processes and Activities	
	A.1 <i>Summary of the processes and activities</i>	Appx-A-23
Appendix B	Information related to work products defined in ISO 26262:2011	
	B.1 <i>Information related to work products defined in ISO 26262:2011</i>	Appx-B-27
Appendix C	Revisions	
	C.1 <i>Revisions</i>	Appx-C-38

Safety Documentation Information Flow

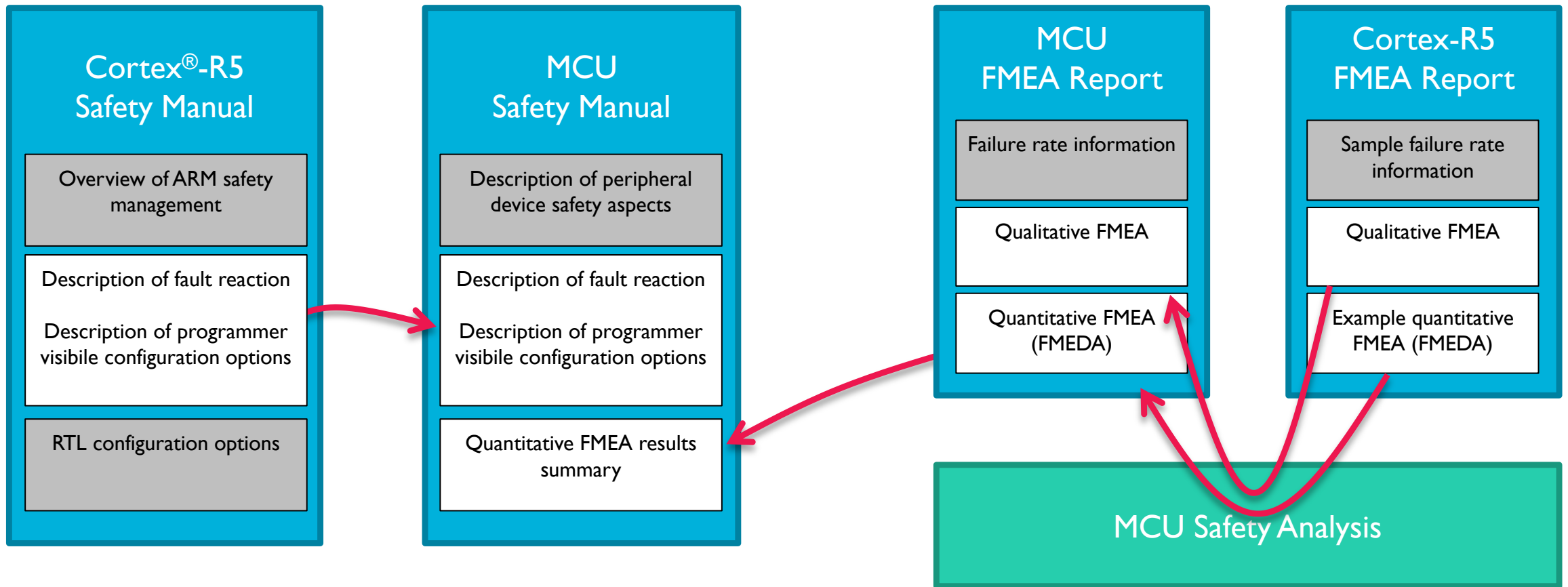


Use Case Example: Support Third-party Assessment of an MCU



- Information or data flow
- Access for audit and assessment purposes





Use Case Example: MCU / SoC Safety Documentation



Case: Ecosystem Partner Support from Yogitech

- Yogitech is a provider of services and solutions to silicon vendors and system integrators to help them meet their functional safety challenges
- Currently supports a number of ARM processor designs
 - Hardware solutions
 - Software solutions

YOGITECH offer for ARM cores

- **fRCPU (available for Cortex-M3)** 
 - optimized tightly coupled fault supervisor for low-cost safety concepts, implementing ASILD “asymmetric redundancy” (ISO 26262-5 D.2.3.6).
- **fRSmartComp (available for Cortex-R4F, R5)** 
 - enhanced dual-core lock-step for fail operational safety concepts, included in ISO 26262 as “2-way voting” (ISO 19451 PAS).
- **fRSTL (available for Cortex-M0, M0+, M3, M4 – in development for A15, A9, A7 – in roadmap for A53, A57)** 
 - Application Independent Software Test Library. Each Test Segment targets a specific function of the CPU. It provides pass/fail information and self-checking signatures (CRC). It may be interrupted at any time by the application SW.
- **fRSmartMultiCore (in roadmap for Cortex-A)** 
 - Complete solution including fRSTL for Cortex-A, fRSmartWatchdog (a SW layer comparing fRSTL results and handling application redundancy) and fRSVC_multicore (a safety verification component that provides customers with the safety analysis and safety verification artifacts to combine fRSTL and fRSmartWatchdog with application redundancy to reach up to ASIL C for both permanent and transient faults).

ARM Compiler 5 Support for Functional Safety

- Compiler Safety Package for software development in safety markets
 - Industrial control, automotive, medical, transportation, military and others

Qualification Kit

- Development process docs
- Safety manual
- Defect report
- Test report



Extended Maintenance

- Five year commitment
- Technical support
- Critical defect fixes



Functional Safety Certified

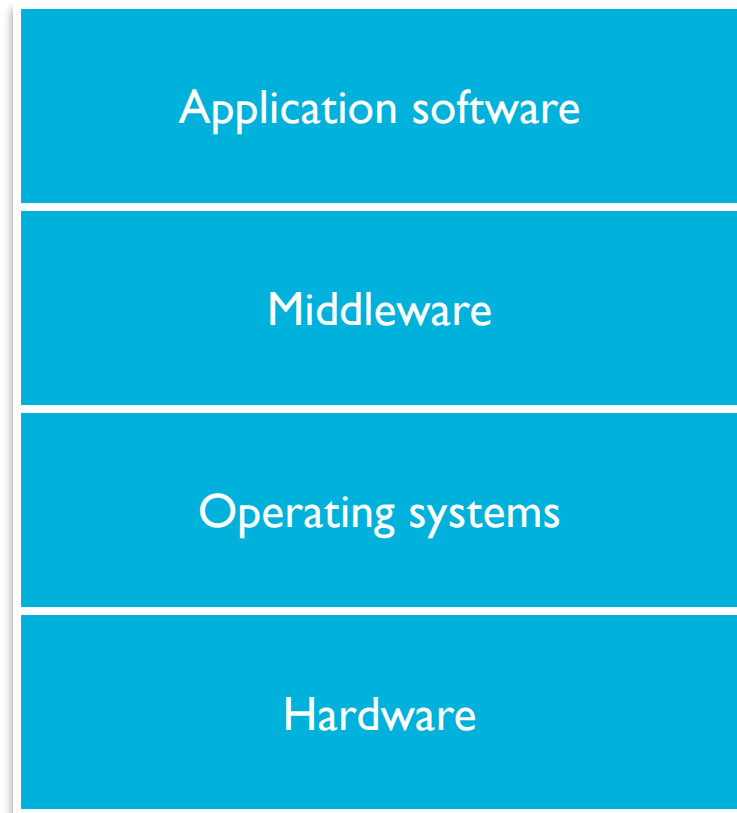
- TÜV SÜD certification
- ISO 26262 (ASILD)
- IEC 61508 (SIL3)



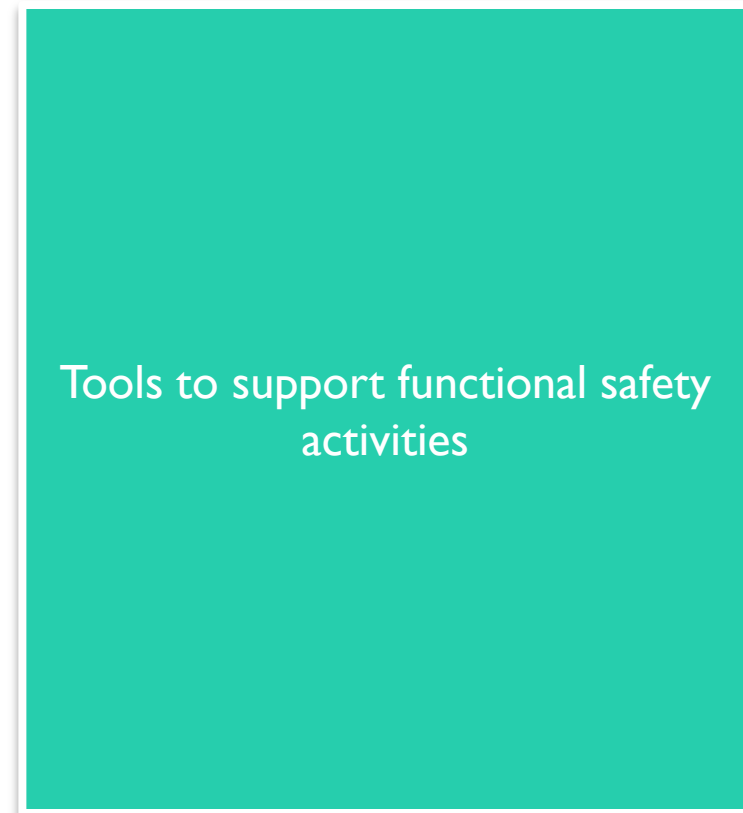
- Access to the Safety Package provided with DS-5 Ultimate and Keil® MDK Pro
 - Valid DS-5 or MDK support and maintenance entitlement enables extended maintenance
 - Compiler installation is an add-on to the standard product installation

Importance of ARM Ecosystem for Functional Safety

- Functional safety support required for all aspects of designs

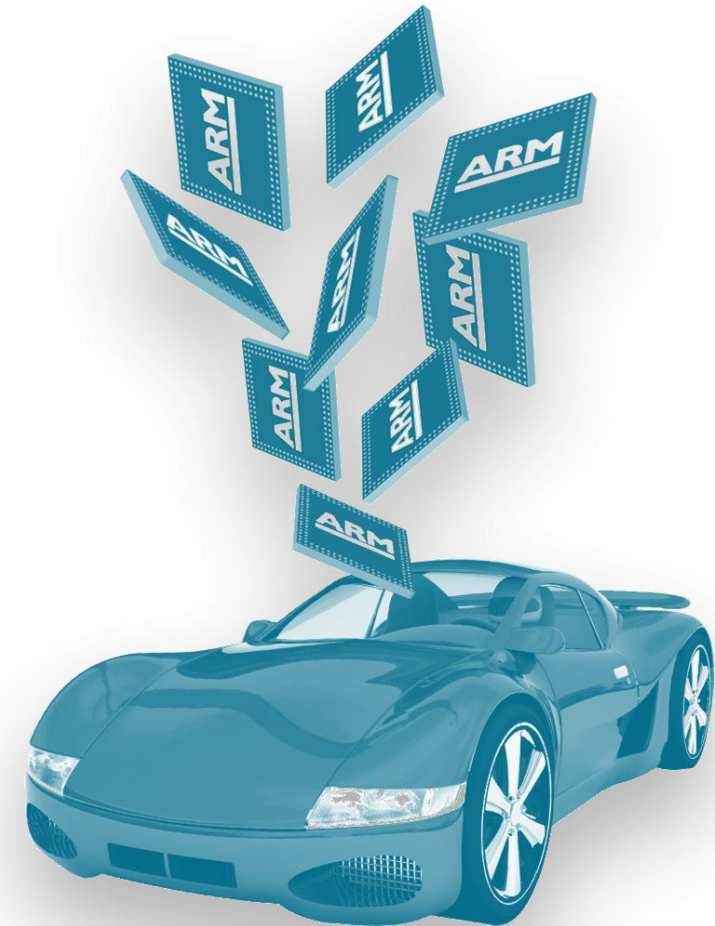


- ...including design, verification and analysis tools



Conclusions

- ARM is actively working on functional safety support
 - Goal is to enable semiconductor manufacturers to develop SoC and MCU designs for safety applications
 - This requires collaboration throughout the ecosystem
 - Actively participating in ISO 26262 standardization activities
- We want to understand your needs
 - What is the best way for ARM to support your safety-related designs?
 - Your expectations for semiconductor suppliers' safety documentation and support?



THANK YOU!

For further information, please contact

Lauri Ora

lauri.ora@arm.com

+44 (0) 7741 272 100