Drivvize

DRIVE. VISION. REALIZE.

Training on SEooC, Safety Manual & deriving safety requirements

Topics

- What is Safety Element ?
- What is SEooC ?
- Comparison between SEooC (Out-of-Context) and Safety element developed in-context
- Process flow & activities for the development of System, HW & SW SEooC
- Pros and Cons of SEooC approach
- What is a Safety Manual ?
- Aspects considered in the development of SEooC
- Strategies for creating an effective safety manual
- How to read and analyze a Safety manual and derive requirements?
- Recommendations for successful AoU implementation

What is a Safety Element ?

- Safety element is a System, HW or SW element that is safety-relevant and it contributes to achieving the safety goal of an item.

- Examples of safety element are:
  - HW elements – Microcontrollers, Power supply, CAN transceiver, Clock etc.,
  - SW elements – CAN Stack, low level drivers, Operating System etc.,
  - System elements – Automated Lane Centering (ALC), Instrument Cluster etc.,

What is Safety Element out of Context (SEooC)?

- Safety Element out of Context (SEooC) is a **safety element** that is developed as ASIL without the context of the actual safety goal/system.

- Safety Element out of Context (SEooC) is not developed for any specific item which always requires the context of a vehicle

- Examples of SEooC safety element are:
  - SEooC HW elements – Microcontrollers, PMIC etc.,
  - SEooC SW elements – OS, Middleware stack, Standalone SW unit, CRC Library etc.,
  - SEooC System elements – Electric Parking system, an ECU etc.,

Drivvize

DRIVE. VISION. REALIZE.

Comparison between SEooC (Out-of-Context) and Safety element developed in context

Comparison between SEooC (Out-of-Context) and Safety element developed in context

Process/Work products	Safety element developed in context	SEooC (Out-of-Context)
Item Definition (functions at vehicle level, operating conditions and limitations)	Known/Pre-defined by OEM	Target Application & Use cases are assumed
Safety Goals	Provided by OEM	Assumed
Functional Safety Concept/Requirements	Provided by OEM	Assumed for System SEooC. Not mandatory for HW & SW SEooC.
Technical Safety Concept/Requirements	Derived from OEM FSR	Requirements derived from assumed FSR or assumed requirements
Development Process	Follow processes and methods as per respective ISO26262:2018 Parts: System Safety Element – Part 4 HW Safety Element – Part 5 SW Safety Element – Part 6	Follow processes and methods as per respective ISO26262:2018 Parts: System SEooC – Part 4, Part 5, Part 6, Part 7(opt) HW SEooC – Part 5, Part 7(opt) SW SEooC – Part 6

Comparison between SEooC (Out-of-Context) and Safety element developed in context

Process/Work products	Safety element developed in context	SEooC (Out-of-Context)
Safety Manual	Not provided	Safety manual created to provide set of external requirements for integrating item/system regarding how the SEooC must be used
Qualification	Assessment is performed, and safety element is qualified in the context of item.	• Assessment is performed for the SEooC scope. • The actual qualification of SEooC occurs upon the complete fulfillment of all AoUs and subsequent validation by the item itself.
Development Approach	Follows a top-down approach i.e., Item definition -> Identify hazards & safety goals -> Identify the Safety path in the system -> Identify the safety-related HW, SW and System elements -> Develop the identified safety-related elements as ASIL.	Follows a bottom-up approach i.e., Identify HW, SW or System elements that must be developed as ASIL -> Formulate assumptions on the ASIL level, Safety goals, the item or system, and the context/environment in which the safety element will be used.

Comparison between SEooC (Out-of-Context) and Safety element developed in context

Example of Safety element developed in context

Item: ALC System — Item

Hazard: When ALC feature provides steering support more than max allowed range then:

- Ego vehicle could have collision with another vehicle which is travelling in adjacent lane and result in hazard.
- If Ego vehicle is travelling in a lane which is adjacent to road edge/barrier, then it could have a collision with barrier or leave the road and result in hazard.
- Ego vehicle could lose stability & can cause hazard.

Safety Goal 02: Prevent excessive lateral adjustment that results in a lane/roadway departure while the ALC system is engaged in accordance with **ASIL D** for all levels of automation. — Safety Goal

Safe State: Disable ALC feature and provide visual notification to driver.

FTTI: 500ms

Comparison between SEooC (Out-of-Context) and Safety element developed in context

Example of Safety element developed in context

FSR: If the calculated steering angle exceeds the limit, transition to the appropriate safe state and issue a notification to the drive. (Allocation: ALC system, ASIL: ASIL B(D))

FSR

TSR_ALC_13_1_**8**	If case of a fault of the calculated steering angle which results in a value that exceeds the Maximum limit for a duration of FDTI, the System shall set the ALC Status Indicator on CAN to Error as safe state within FRTI	ASIL B(B)
TSR_ALC_13_1_**9**	If case of a fault of the calculated steering angle which results in a value that exceeds the Maximum limit for a duration of FDTI, the System shall set the Steering Angle command sent on CAN to 0 deg as safe state within FRTI	ASIL B(B)
TSR_ALC_13_1_**10**	In case fault of calculated steering disappears, the System shall restore the ALC Status Indicator on CAN as per ALC Function Request	ASIL B(B)
TSR_ALC_13_1_**11**	In case fault of calculated steering disappears, the System shall restore the Steering Angle command on CAN as per ALC Function Request	ASIL B(B)

TSR

Comparison between SEooC (Out-of-Context) and Safety element developed in context

Example of System SEooC (Out-of-Context) - Microcontrollers (MCU)

- NXP's S32K3xx series MCU's designed for multiple automotive safety-related applications such as Body Control Module (BCM), Battery management applications, TPMS, Transmission control unit etc.,

- Infineon's Aurix TC3xx MCU's designed to host Safety applications such as braking, Airbag, power steering to fail operational sensor-based systems for Autonomous driving.

- MCU supplier makes assumptions on the safety goals or safety features it supports.

- Example of generic safety goal assumptions of MCU - Prevent the MCU outputs (digital control, analog control, and communication data) from causing a hazard.

Comparison between SEooC (Out-of-Context) and Safety element developed in context

Example of SW SEooC (Out-of-Context) - SW CRC library

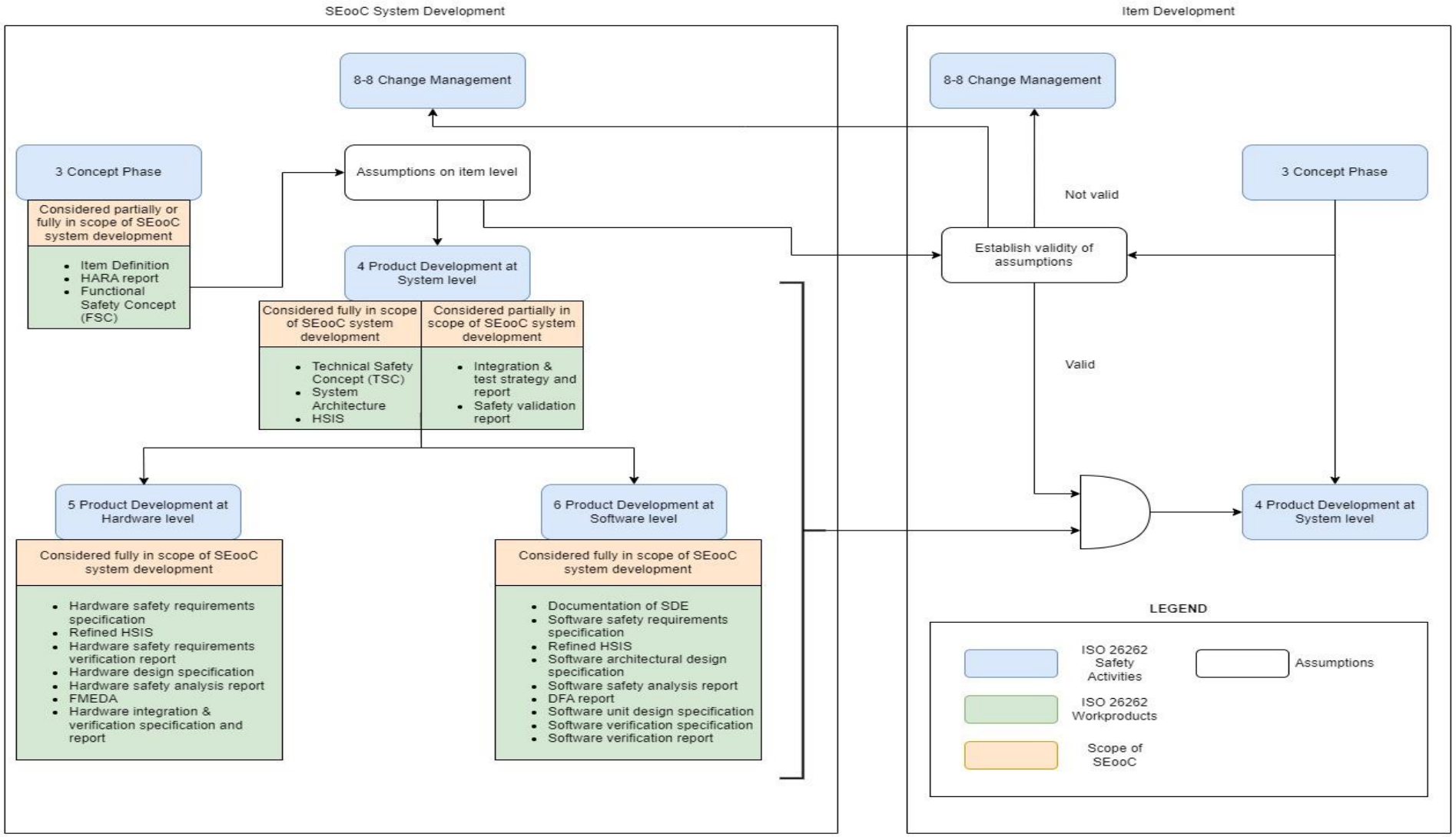
- Purpose - to provide interfaces to calculate CRC based on different CRC algorithms.
- CRC is a functionality that is used for ensuring integrity of memories and communication.
- CRC Applications - E2E (End to end protected) communication, Memory Integrity
- From the perspective of CRC developer, no difference with respect to ECU (Instrument cluster, Domain controller, Front view Camera).
- No difference with respect to application (Memory, Communication)
- SW CRC Library developer does not need to know the exact Safety goals – the only pre-requisite is to know which algorithm must be used.

Drivvize

DRIVE. VISION. REALIZE.

Process flow and activities for the development of System, HW & SW SEooC

Process flow for development of System SEooC



Process flow for development of System SEooC

- SEooC developer defines the purpose, functionalities and external interfaces of the SEooC based on assumptions

- Examples from ALC feature
  - The system is designed for maximum road slope of 32 %
  - The system is designed to operate only when the lane markings are detected
  - The system interacts with external systems to get required vehicle information such as vehicle speed.
  - The system deactivates the function when requested by driver over steering wheel or brake pedal input.

- To identify the TSRs, assumptions on safety requirements and context (Safety goal & FSR) are made.

- HARA is performed to derive safety goals (optional)

- Examples from ALC feature
  - Avoid unintended steering from function (ASIL B)
  - The system does not activate the function at high vehicle speeds (ASIL B)
  - The system receives driver request information as ASIL from an external source (ASIL B)

Process flow for development of System SEooC

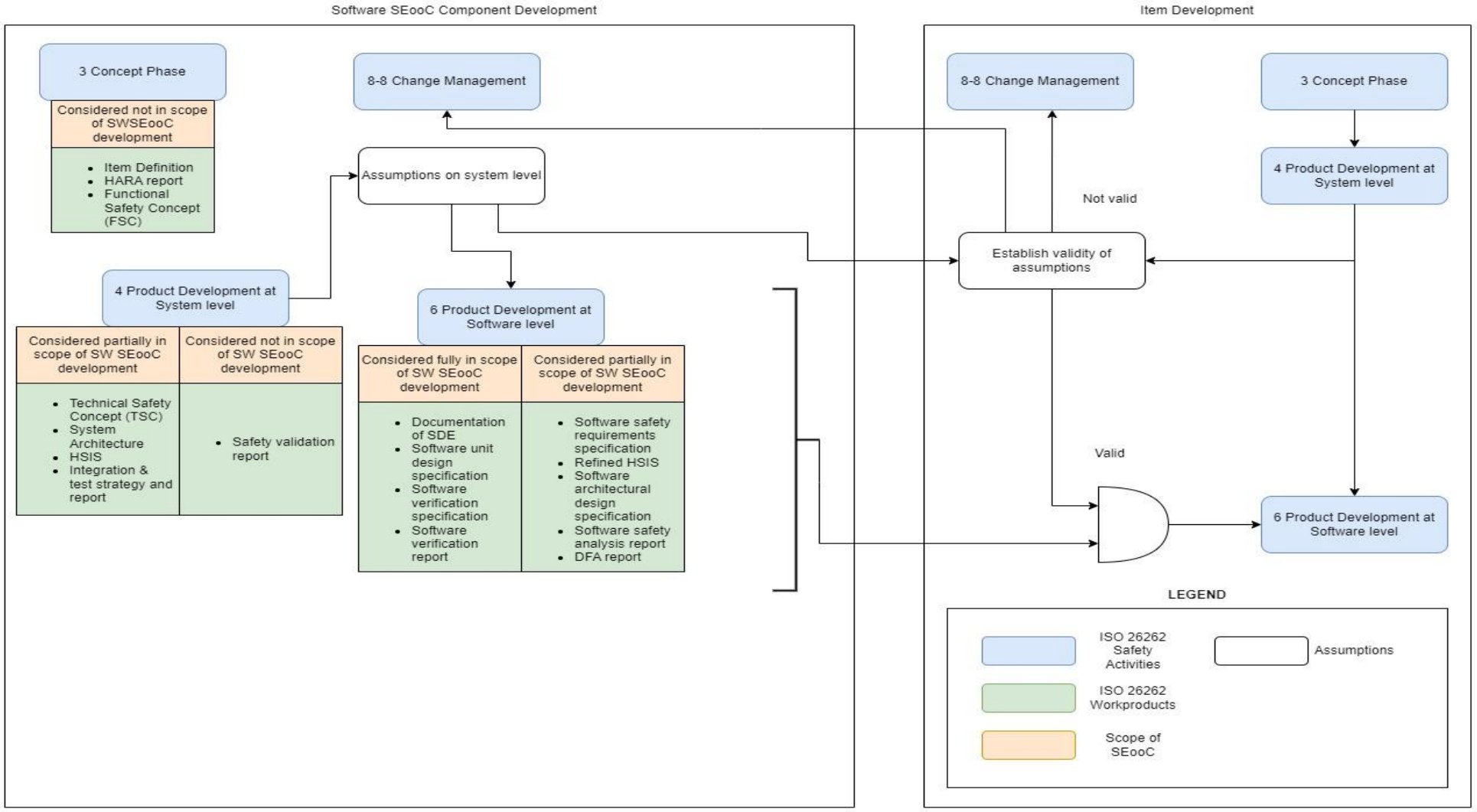
- After TSRs are derived, SEooC is developed as per ISO 26262 process and methods corresponding to its ASIL capability.

- Assumed requirements and context provided to item integrator

- During integration of the SEooC into the item, the FSRs of the item are matched with assumed FSRs from SEooC to establish their validity. Any assumption mismatch triggers an impact analysis activity.

- As per the outcome of impact analysis,
  - if the difference is acceptable w.r.t the achievement of the safety goal, then no action is taken.
  - if the difference impacts the achievement of the safety goal, then a change is necessary on either the item definition or the FSC and thereby to SEOOC itself

Process flow for development of SW SEooC



Process flow for development of SW SEooC

- The SEooC developer defines the purpose of SW component, its boundaries, target environment, its functionalities and its properties.

- Examples
  - The software component is integrated into a given software layered architecture.
  - Any potential interference caused by the software component is detected and handled by its environment.

- To derive the SSRs of SEooC, assumptions on higher level safety requirements are made.

- Examples
  - The software component detects any corruption on the input data which can violate safety goals
  - A default value is returned with a fault status for any error condition detected

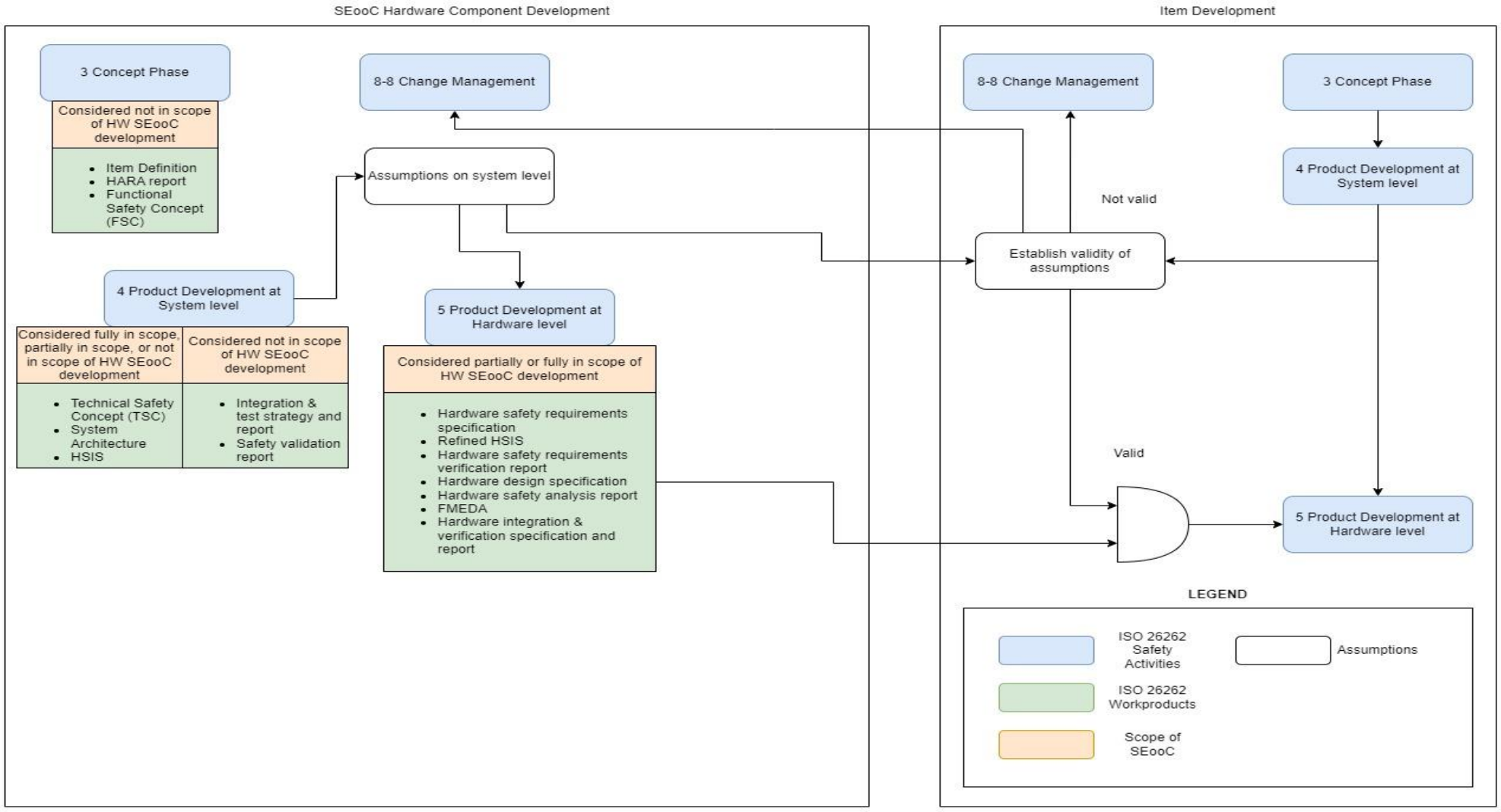
- Once necessary assumptions on the software component are explicitly stated, SEooC is developed as per ISO 26262 Part 6 process and methods corresponding to its ASIL capability. Necessary information such as compiler settings are provided to integrator.

Process flow for development of SW SEooC

- Before the SW component is integrated with other SW components in new context, the validity of all the assumptions made on this SEooC are checked w.r.t this context. If any assumptions regarding the SW component do not fit the new context, then change management activity of impact analysis is initiated.

- As per the outcome of impact analysis,
  - if the discrepancies are acceptable w.r.t the achievement of the software safety requirements, then no action is taken.
  - if the discrepancies impacts the achievement of the software safety requirements, then a change is necessary on requirements and thereby to SEOOC itself

Process flow for development of HW SEooC



Process flow for development of HW SEooC

- The development of an MCU as SEooC starts with making assumptions on TSRs and system level design external to the SEooC.

- Examples of Assumed TSR
  - A memory protection unit is present to provide the possibility of separating software tasks with different ASILs.
  - To achieve a safe state, the MCU drives all I/O outputs to a low state when reset is asserted

- Examples of Assumption on system level design external to the SEooC
  - The ALC system shall implement a safety mechanism on the power supply to the MCU to detect over voltage and under voltage failure modes.
  - The ALC system shall implement a windowed watchdog safety mechanism external to the MCU to detect either clocking or program sequence failures of the MCU.

- Based on SEooC assumptions on the TSRs and system level design external to the MCU, SEooC is developed as per ISO 26262 Part 5 process and methods corresponding to its ASIL capability.

- Based on SEooC assumptions on the TSRs and system level design external to the MCU, the safety analyses and the analysis of dependent failures internal to the MCU are performed referring to ISO 26262-9.

Process flow for development of HW SEooC

- After the end of development of SEooC, necessary information regarding the work products are provided to system integrator.

- When the MCU developed as an SEooC is considered in the context of the item HW product development phase, the validity of all SEooC assumptions are established. If there any mismatches between SEooC assumptions and system requirements, then change management activity of impact analysis is conducted.

- As per the outcome of impact analysis,
  - if the difference is acceptable w.r.t the achievement of the safety goal, then no action is taken.
  - if the difference impacts the achievement of the safety goal, then a change is necessary on either the FSC or the TSC and to SEooC itself
  - safety metrics are recalculated. If the recalculated metrics show that the design meets the system targets, then no change is necessary.

Pros and Cons of SEooC Approach

Pros	Cons
Reuse SEooC safety elements across several applications/systems	Verification of SEooC in all assumed contexts/systems is challenging
Cost of certifying the element as ASIL compliant is spent only once instead of developing as ASIL for every system separately	Correct/Complete identification of AoU's at all levels (Mechanical, System, HW, SW, Verification etc.,) is challenging and must be done meticulously
After impact analysis, if SEooC is being re-used without changes in other systems, then engineering effort spent on integrating the SEooC again can be minimized by re-using previous work products.	Mis-understanding that simply having an SEooC in the System ensures that the system is ASIL compliant

What is Safety Manual ?

- Safety Manual is a document that describes safety functionalities provided by the SEooC and how to use/integrate the SEooC correctly.

NOTE: Safety Manual is only the starting point for user to understand the safety features and functionalities provided by the SEooC. The User shall also refer to other supporting documents such as data sheets, reference manual or any other documents shared by the SEooC supplier which gives further details of the safety features.

- Links to a few safety manuals of SEooC are given below:
  - Safety manual for NxP MPC5777C – Present Here
  - Safety manual for Microchip DS40002252A – Present Here

Aspects considered in the development of SEooC

1. Scope and Target applications of the SEooC
2. Assumptions on requirements of certain hierarchical-level (item/system) where the development of an SEooC starts
3. Assumptions on design external to the SEooC
4. Correct implementation of the derived requirements for SEooC from the assumed requirements and design
5. Development of Safety strategy and concept for SEooC
6. Development of SEooC according to the ISO 26262 series of standards
7. Identification of Assumptions of use (AoU)
8. Documenting the SEooC scope, approach & assumptions in a Safety manual
9. Validation of AoU by the Integrator of SEooC during development of item

Drivvize

DRIVE. VISION. REALIZE.

Strategies for creating an effective safety manual

Strategies for creating an effective safety manual

- Generally, there are three different stakeholders for a safety manual:
  - Supplier of SEooC
  - Project team of a system who is looking for an ASIL Certified SEooC
  - Technical team who will be working on integrating the ASIL Certified SEooC with the system to implement the required Safety functionality

- If a safety manual covers the needs/expectation of each of these stakeholders, then we could consider the safety manual to be “effective”.

Strategies for creating an effective safety manual

- Describe safety features, functionalities, safety mechanisms & its diagnostic coverage, HW metrics and the ASIL compliance capability of the SEooC in the safety manual document, using which the item/system integrator can identify suitable ASIL certified SEooC and successfully achieve the Safety functionality
- **Examples**:
  - In case of Microcontroller SEooC, the safety manual describes the peripherals/modules which are safety relevant and the safety mechanisms that are implemented by supplier to detect its failure modes
- Specify any pre-requisites for the SEooC to successfully achieve its safety functionality
- **Examples**:
  - SW SEooC can be used only in AUTOSAR Architecture.
  - SW SEooC shall be executed in a microcontroller that is ASIL B compliant.
  - Specify Dependencies that the SEooC has on other elements (internal/external to SEooC) and Constraints that the SEooC imposes on the integrating item/system.
- **Examples:**
  - The power supply inputs to the Microcontroller SEooC shall be monitored for over-voltage and under-voltage failures by an externally safety-related hardware.
  - The NAND Flash memory SEooC shall be used only in an environment within the AEC Grade 1 temperature range.
  - External CRC library shall be provided to the Microprocessor SEooC

Strategies for creating an effective safety manual

- Trace the external measures coming from safety analyses (FMEA/FTA/both and DFA) activity to Safety manual

- Specify the safety measures and the verification activities that should be taken care by the integrator of the SEooC.

- **Examples:**
  - The Integrator shall implement and configure MPU to protect the safety critical memory region of SW SEooC to ensure FFI.
  - The Integrator shall implement an External Windowed Watchdog to supervise the computation and software execution within the Microcontroller SEooC
  - The Integrator shall use an external safety-related hardware to monitor and control Microcontroller SEooC reset pin to detect unintended reset cycling failure

- Clearly indicate the work products provided to the Item Integrator

Drivvize

DRIVE. VISION. REALIZE.

How to read and analyze a Safety manual and derive requirements?

Aspects to look for in the Safety Manual of a SEooC

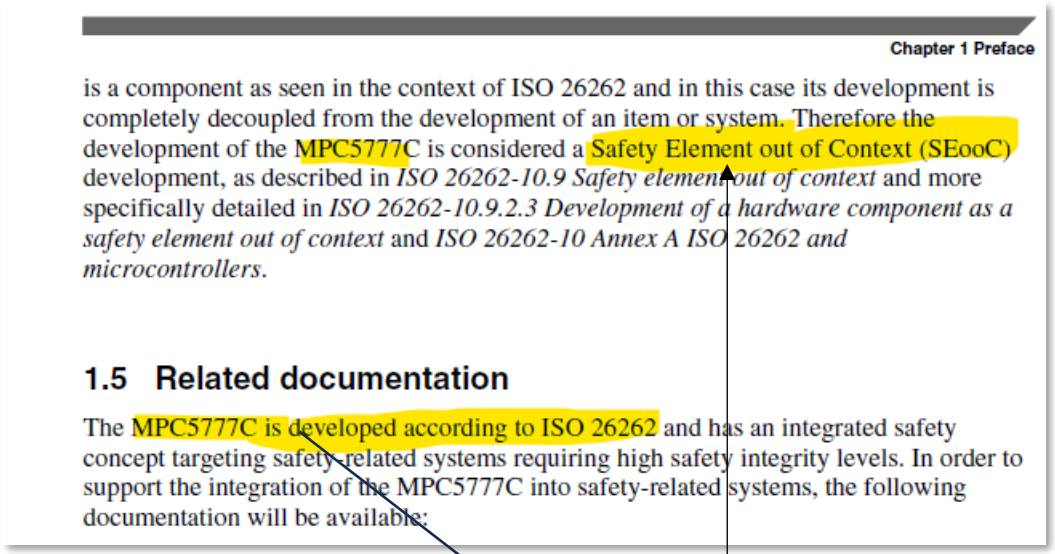
- Safety Strategy/Approach
- Assumed Targeted Applications
- Scope
- Assumed Safety goals/Safety requirements
- Assumed System design/Safety Architecture
- Maximum FHTI that can be achieved
- SEooC Safety Concept and Architecture
- Safety Mechanisms of the SEooC (Internal & External)
- AoU (Assumptions of Use) – Mechanical, System, SW, HW, Verification & Others
- In case of ASIL decomposition or FFI inside SEooC, strategy/approach for achieving Independence/FFI

Safety Strategy/Approach

Questions to ask SEooC developer:

- Is the SEooC developed in accordance with ISO26262 processes and methods ?
- Is it already ASIL Certified or in the process of certification or not certified at all?
- What is the maximum ASIL level achieved by the SEooC?

Safety Strategy/Approach Examples



Chapter 1 Preface

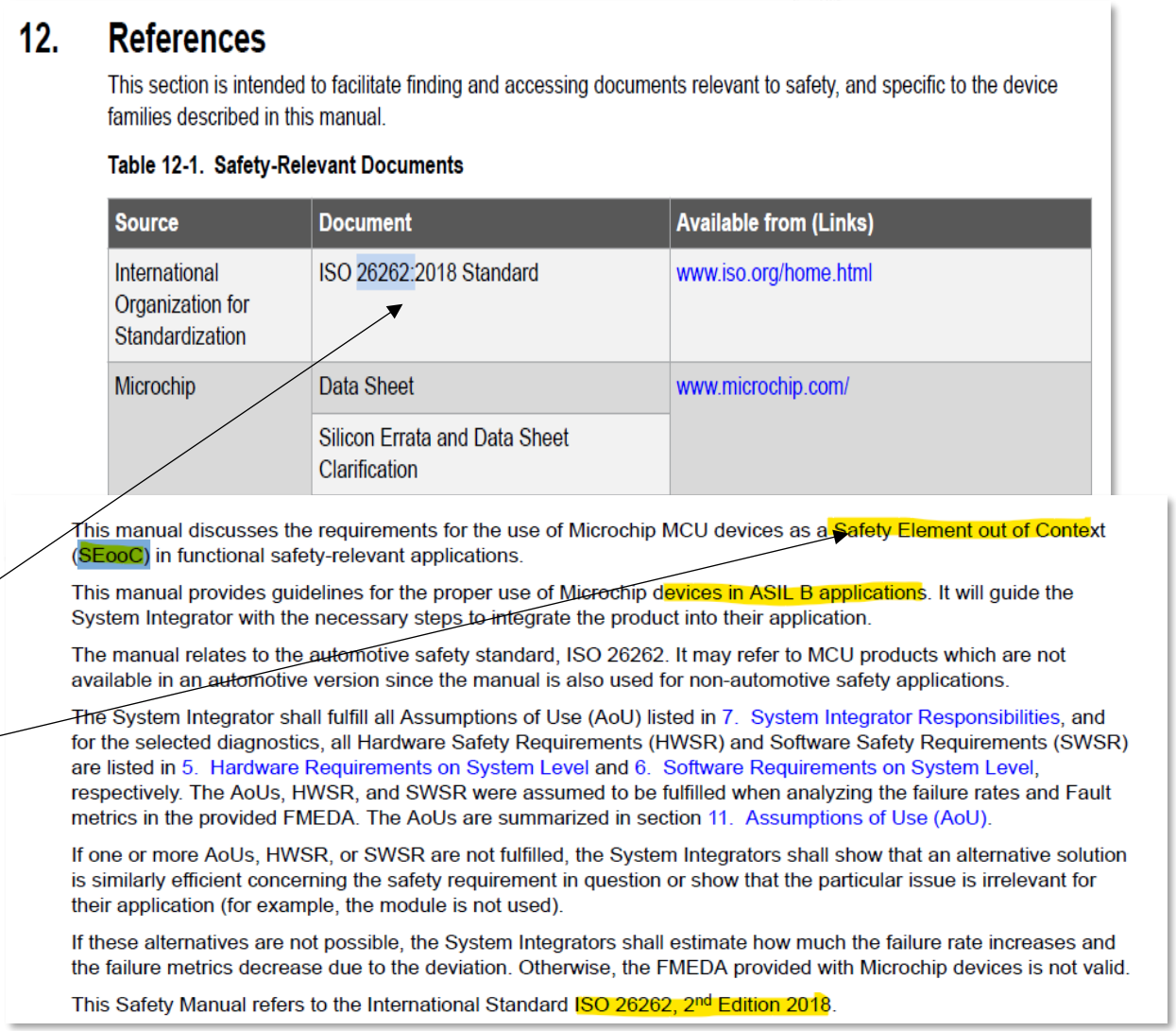
is a component as seen in the context of ISO 26262 and in this case its development is completely decoupled from the development of an item or system. Therefore the development of the MPC5777C is considered a Safety Element out of Context (SEooC) development, as described in *ISO 26262-10.9 Safety element out of context* and more specifically detailed in *ISO 26262-10.9.2.3 Development of a hardware component as a safety element out of context* and *ISO 26262-10 Annex A ISO 26262 and microcontrollers*.

1.5 Related documentation

The MPC5777C is developed according to ISO 26262 and has an integrated safety concept targeting safety-related systems requiring high safety integrity levels. In order to support the integration of the MPC5777C into safety-related systems, the following documentation will be available:

Examples of statements that clarify that the element was developed as SEooC, which ASIL level is supported and which edition of ISO2626 was used for development

Image from: MPC5557



12. References

This section is intended to facilitate finding and accessing documents relevant to safety, and specific to the device families described in this manual.

Table 12-1. Safety-Relevant Documents

Source	Document	Available from (Links)
International Organization for Standardization	ISO 26262:2018 Standard	www.iso.org/home.html
Microchip	Data Sheet	www.microchip.com/
	Silicon Errata and Data Sheet Clarification	

This manual discusses the requirements for the use of Microchip MCU devices as a Safety Element out of Context (SEooC) in functional safety-relevant applications.

This manual provides guidelines for the proper use of Microchip devices in ASIL B applications. It will guide the System Integrator with the necessary steps to integrate the product into their application.

The manual relates to the automotive safety standard, ISO 26262. It may refer to MCU products which are not available in an automotive version since the manual is also used for non-automotive safety applications.

The System Integrator shall fulfill all Assumptions of Use (AoU) listed in 7. System Integrator Responsibilities, and for the selected diagnostics, all Hardware Safety Requirements (HWSR) and Software Safety Requirements (SWSR) are listed in 5. Hardware Requirements on System Level and 6. Software Requirements on System Level, respectively. The AoUs, HWSR, and SWSR were assumed to be fulfilled when analyzing the failure rates and Fault metrics in the provided FMEDA. The AoUs are summarized in section 11. Assumptions of Use (AoU).

If one or more AoUs, HWSR, or SWSR are not fulfilled, the System Integrators shall show that an alternative solution is similarly efficient concerning the safety requirement in question or show that the particular issue is irrelevant for their application (for example, the module is not used).

If these alternatives are not possible, the System Integrators shall estimate how much the failure rate increases and the failure metrics decrease due to the deviation. Otherwise, the FMEDA provided with Microchip devices is not valid.

This Safety Manual refers to the International Standard ISO 26262, 2nd Edition 2018.

Image from: DS40002252A

Safety Strategy/Approach Examples

11. Assumptions of Use (AoU)

The Microchip MCUs have been designed according to the Microchip New Product Development standard processes and can be considered as SEooC, since they are general purpose microcontrollers and have not been designed explicitly for any system.

Microchip has made certain AoUs to meet technical and functional safety requirements at the system level. The AoUs are described in sections 11.2-11.6. The details of some of these AoUs can be found in other sections in this manual, where they are addressed as part of a specific topic being discussed.

It is the responsibility of the System Integrator to address all the AoUs listed in this manual.

Examples of approach for development process. In such cases it must be clarified if internal process is certified or the Element will be certified or not

If there is no clear statement from the Safety manual about the status of certification then it must be confirmed if the element is already certified, or certification is ongoing.

Failure Rates and FMEDA

Per ISO 26262:2018, FMEDA targets are summarized in the table below. The corresponding FMEDA needs to be completed to assess the quantitative coverage for the microcontroller.

Table 8-1. Hardware Architectural Metrics

Hardware Architectural Metrics	Microcontroller Target	System Target ASIL B
Single Point Fault Metrics	**(Note 2)**	≥ 90%
Latent Fault Metrics	**(Note 2)**	≥ 60%
Probabilistic Metric from Random Hardware Failure (PMHF) **(Note 1)**	$< 10^{-8}\ h^{-1}$ corresponding to 10 FIT	$< 10^{-7}\ h^{-1}$ corresponding to 100 FIT

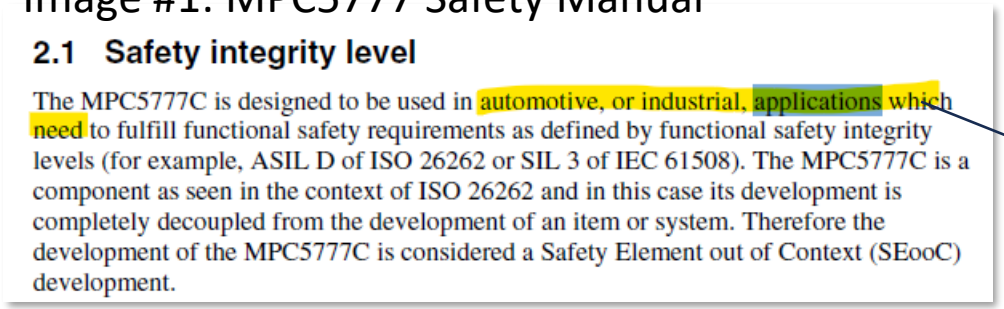
Images from: DS40002252A

Assumed Targeted Applications and Examples

Questions to ask SEooC developer:

- Do the assumed target applications of the SEooC match the application of the system using the SEooC ?

Image #1: MPC5777 Safety Manual

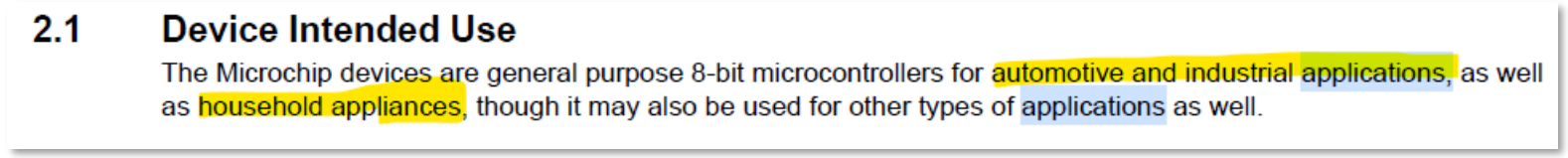


2.1 Safety integrity level

The MPC5777C is designed to be used in automotive, or industrial, applications which need to fulfill functional safety requirements as defined by functional safety integrity levels (for example, ASIL D of ISO 26262 or SIL 3 of IEC 61508). The MPC5777C is a component as seen in the context of ISO 26262 and in this case its development is completely decoupled from the development of an item or system. Therefore the development of the MPC5777C is considered a Safety Element out of Context (SEooC) development.

Examples of targeted applications specified in different Safety manuals such as ADAS systems, Body Control Module, Battery management etc.,

Image #2: Sample-MCU-Safety-Manual-DS40002252A



2.1 Device Intended Use

The Microchip devices are general purpose 8-bit microcontrollers for automotive and industrial applications, as well as household appliances, though it may also be used for other types of applications as well.

Image #1 from: MPC5557

Image #2 from: DS40002252A

Scope and its examples

Questions to ask SEooC developer:

- What is provided as ASIL within the scope of SEooC?

- What is considered as **not** safety-relevant?

Examples:

- In case of MCU, if Ethernet peripheral is not used by the system, then it will be classified as not safety relevant.
- In case of SW BSW Stack SEooC, the components that are not required as ASIL will be classified as not safety relevant.
- In case of Computer vision sensor, following measurements are provided with ASIL B integrity and lane detection is provided with QM integrity:
  - Hazard Detection
  - Object measurements

Scope Examples

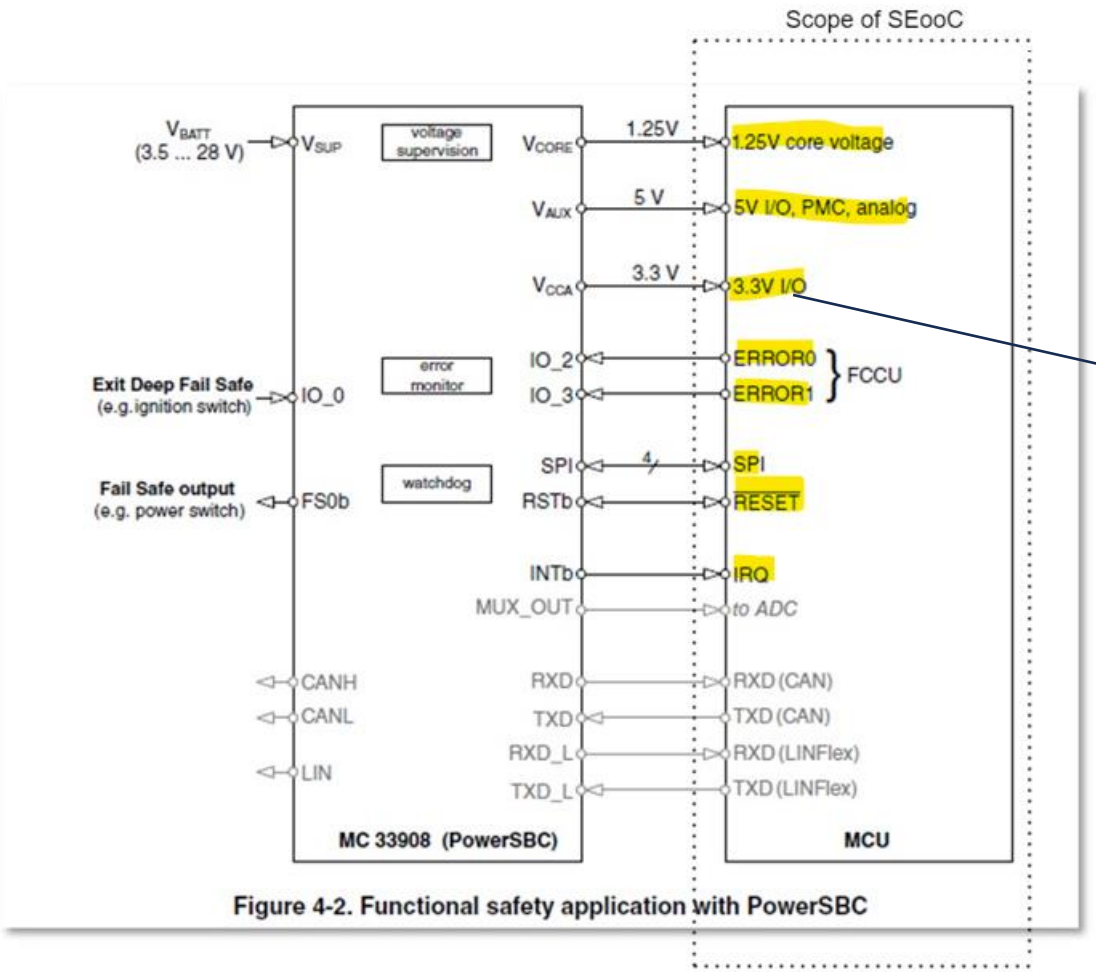


Figure 4-2. Functional safety application with PowerSBC

In case of MCU, examples of the peripherals within the micro which are considered as safety-relevant

Image from: MPC5557

Assumed Safety Goals/Safety Requirements

- The assumed Safety goals/Requirements help us to understand if the SEooC's safety development is aligned with safety scope of our system.

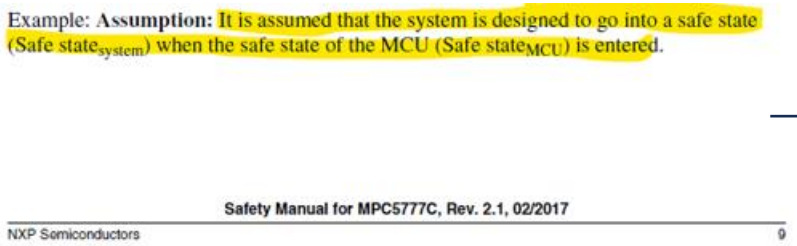
- Some suppliers provide their Systems/Technical Safety requirements as a separate document.

Assumed Safety Goals/Safety Requirements Examples

Safety Goal:

Prevent the microcontroller outputs (digital control, analog control, and communication data) from causing a hazard

FSR:


Example: **Assumption:** It is assumed that the system is designed to go into a safe state (Safe state$_{system}$) when the safe state of the MCU (Safe state$_{MCU}$) is entered.

Safety Manual for MPC5777C, Rev. 2.1, 02/2017

NXP Semiconductors 9

Functional safety requirement

Example of MCU SEooC assumption mismatch with requirements of the system

Assumption: *[SM_042] It is assumed that the SEooC monitors when the power supply voltage is above and below the specified operational range and the monitoring is enabled by default*.

Assumption applicable for the system external to SEooC**:** *[SM_053] For each safety-related power supply input to MCU SEooC, an off-chip voltage monitor detects both undervoltage and overvoltage conditions*.

Since both the safety mechanisms from the SEooC supplier as assumptions address same failure modes, the assumption [SM_053] was discussed with SEooC developer and was rejected by system integrator. The system integrator then configured the error reaction of supply voltage monitoring mechanism within MCU SEooC to not trigger a MCU reset.

Image from: MPC5557

Assumed System Design/Safety Architecture

Questions to ask SEooC developer:

- Is the assumed System Architecture aligned with our System's architecture (i.e., assumed external ICs in the System, Hardware connections are matching)?
- Have all the assumptions regarding System design been verified?
- Are there any assumptions provided to the integrator to verify certain assumptions at the Product level/System level ?

Assumed System Design/Safety Architecture Examples

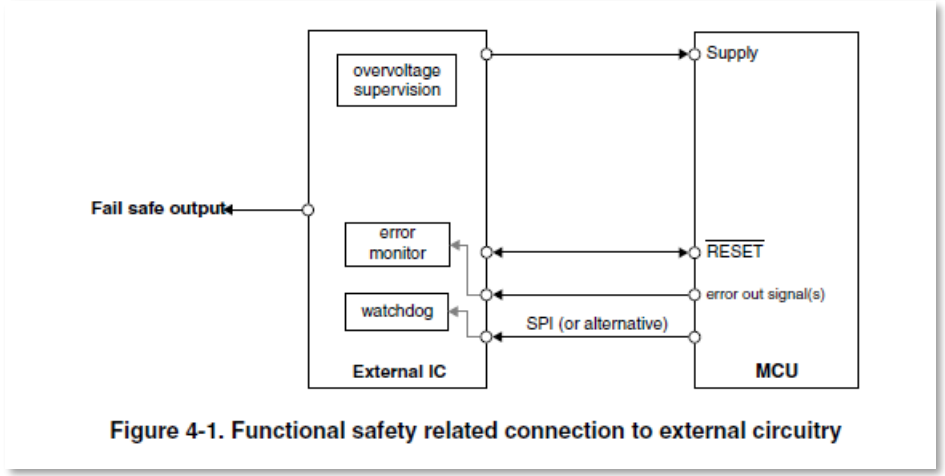
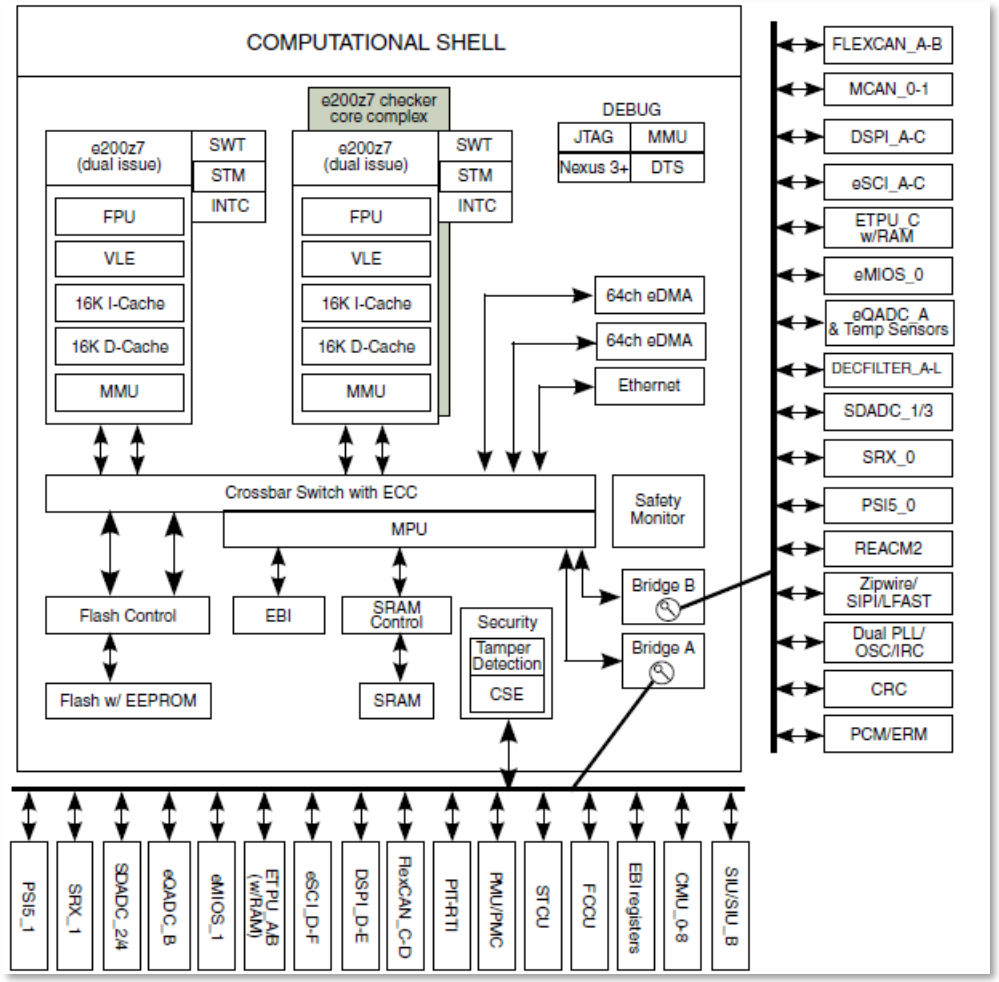


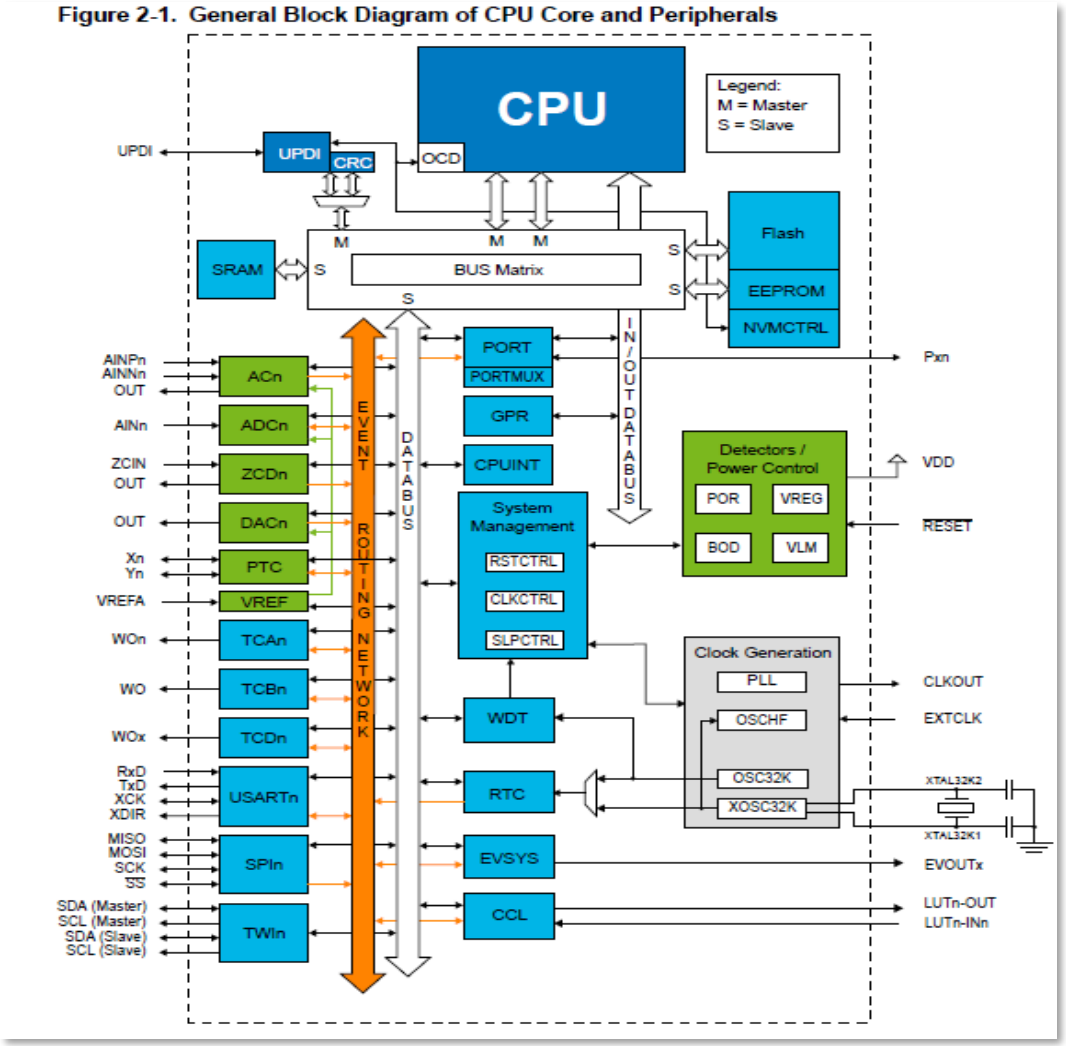
Figure 4-1. Functional safety related connection to external circuitry



MPC5777C Block Diagram

Images from: MPC5557

Assumed System Design/Safety Architecture Examples



DS40002252A Block Diagram

Image from: DS40002252A

Maximum FHTI that can be achieved

Questions to ask SEooC developer:

- What is the maximum FHTI that is achieved by safety mechanisms implemented by MCU SEooC ?

- What does FHTI mean in the context of the element? Which is the fault (start point) and which is the safe state (end point)?

NOTE: In some SEooC safety manual, the supplier would have referred to FHTI as Fault Indication time or FTTI.

Examples of Maximum FHTI that can be achieved

2.3 Mission Profile

The key aspects of the mission profile are summarized in the following table.

Targets	Item	Value	Comments
Device targets	Device lifetime	15 years	15 years correspond to 10,000 hours of active operation of the MCU. It might be necessary to adjust the product lifetime per specific operating profile of the application.
Product targets	Safety function target	ASIL B	This is the target of the safety functions for these devices
	Lifetime	15 years	
	Total active operation	10,000 hours	
	Maximum time of operation without a start-up reset	10 hours	
	FTTI budget	Application dependent	The FTTI budget allocated to the device is use-case specific. FTTI down to 5 ms can be supported for Single Point Faults. The required FTTI period depends on the application; it is up to the application to determine the required FTTI period.
	Temperature Profile	As per the data sheet	

Image from: DS40002252A

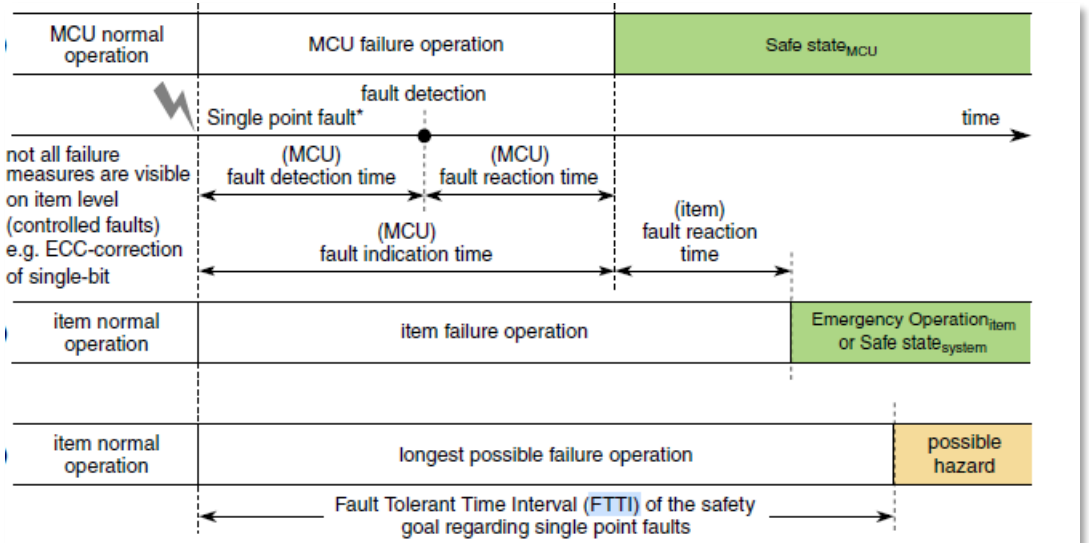


Figure 2-6. Fault tolerant time interval for single point faults

Image from: MPC5557

SEooC Safety Concept & Architecture and its Example

Questions to ask SEooC developer:

- What is the information that is available in the safety manual regarding the safety concept of this SEooC and its Architecture ?

MPC5777C Safety concept with elements (SBC) external to the SEooC

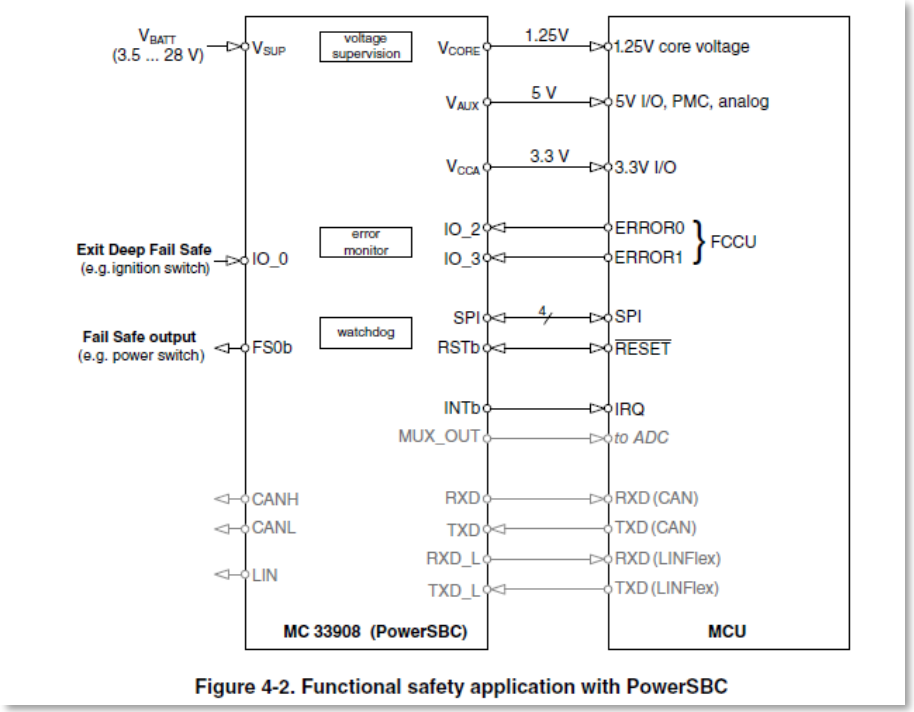


Figure 4-2. Functional safety application with PowerSBC

Image from: MPC5557

Safety Mechanisms of the SEooC (Internal & External)

Questions to ask SEooC developer:

- What are the Safety mechanisms that are implemented within the SEooC to detect its internal faults?
- Are there any Safety mechanisms that are to be implemented by other safety-related elements of the system that are interfacing with our present SEooC?
- What is the diagnostic coverage provided by HW related Safety mechanisms provided in SEooC ?
- Do the safety mechanisms listed in safety manual provide coverage for only faults on primary function, or do they also provide coverage for faults of safety mechanism ?
- Is the safety mechanism that is proposed in the safety manual based on HW or SW or both ?
- What is the test interval of the safety mechanism ? Is it Continuous/Periodic/On-Demand/Once per drive cycle ?

Safety Mechanisms of the SEooC (Internal & External)

Questions to ask SEooC developer:

- What is the estimated “FDTI + FRTI” time (in microseconds) for each safety mechanism ?
- Is this a safety mechanism, that is already implemented & available from SEooC supplier, but needs to be purchased by system integrator?
- How does each safety mechanism within the SEooC report its error ? Is it through interrupt or polling ?
- What is the safe state recommended by SEooC supplier when the safety mechanisms to be implemented within the SEooC and outside SEooC detects faults ?

Example Safety Mechanisms of the SEooC (Internal & External)

Hardware Safety mechanisms from SEooC

4.1.1 Assumed functions by separate circuitry

This section describes external components used in a system in conjunction with the MPC5777C for safety-related systems.

It should be noted that failure modes of external services are only partially considered in the FMEDA of the MPC5777C (for example, clock(s), power supply), and must be fully analyzed in the system FMEDA by the safety system developer.

4.1.1.1 High impedance outputs

If the MPC5777C is considered to be in a Safe $state_{MCU}$ (for example, unpowered and outputs tristated), the system containing the MPC5777C may not be compliant with the Safe $state_{system}$. A possible system level safety measure to achieve Safe $state_{system}$ may be to place pull-up or pull-down resistors on I/O when the high-impedance state is not considered safe.

Assumption: [SM_038] If a high-impedance state on an output pin is not safe, pull-up or pull-down resistors shall be added to safety-related outputs. The need for this will be application dependent for the unpowered or reset (tristated I/O) MPC5777C.[end]

Rationale: In order to bring the safety-related outputs to such a level, that a Safe $state_{system}$ is achieved.

Software Safety mechanisms from SEooC

5.2.7.1 Runtime checks

Parts of the MPC5777C configuration registers do not provide the functional safety integrity IEC 61508 series and ISO 26262 require for high functional safety integrity targets. This relates to systematic faults (for example, application software incorrectly overwriting registers), as well as random hardware faults (bit flipping in registers).

Assumption: [SM_170] The CRC calculation shall be executed at least once per FTTI to verify the content of the safety-relevant configuration registers. [end]

Images from: MPC5557

AoU (Assumptions of Use)

Questions to ask SEooC developer:

- What kind of assumptions of use are provided in the Safety manual?
  - System/Product Assumptions
  - Restrictions/Constraints
  - HW Integration
  - Software Implementation
  - Software Configuration
  - Mechanical setup of sensors,
  - Verification etc.,

- Is the Assumption clearly understandable (i.e., is it clear on what must be done in our System to fulfill the assumption)?

- If not, we must discuss it with others (within Systems team or HW team or SW team or SEooC Supplier)

- If we have an alternate method to meet an AoU, then we must discuss it with SEooC supplier and get their feedback.

Recommendations for identifying requirements from AoU

- Understand the nature of the assumption to determine whether to bring it into as a safety requirement (TSR/HSR/SSR). Think about how we will verify the assumption.

  - Is it a generic assumption that must be taken care by a company's internal process?
  - Is it a general assumption that is fulfilled by a product level decision?
  - Is it an assumption that is/ must be fulfilled by Customer requirements?
  - Is it a single implementable requirement by HW or SW that can be verified or is it a general requirement that needs a broader verification?

- Bring in an AoU as a safety requirement (TSR/HSR/SSR) only if it can be implemented (i.e., in design) by Systems, SW, HW or Manufacturing.

- Process AoU's need not be added as a safety requirement. They can be considered in the Safety plan as part of defining the Safety strategy.

Recommendations for successful AoU implementation

- Discuss with SEooC supplier on each AoU and safety mechanism and document System Integrator's feedback & supplier's feedback [i.e., whether we agree, disagree (with rationale) or an AoU/safety mechanism is not applicable (with rationale)]. This will help the safety engineers who are maintaining the safety requirement documents.

- Document also the alternate methods that was discussed and agreed with SEooC supplier.

- Provide traceability from AoU's and safety mechanisms using their unique ID provided by the SEooC supplier to the safety requirements.

Safety manual analysis example

Sl	Safety Mechanism ID	Peripheral Gr	Safety Mechan Categor	Safety Mechanism Description	Extern Intern	Safety Mechanism T	SW Trigger/ Configure/ Cre	Safety mechanism timing behavior w.r.t test interva	Estimated Fault Detectio time in µs	Error Reactio (Interrupt/Polli	Safe State
4	SM3.RESET_MON	Reset & Boot	Diagnostic	Monitors and controls the chip's reset input/output. Detects when the chip is in reset. Initiates a chip reset Off-chip safety-related hardware monitors and controls chip's reset pin to detect scenarios like reset cycling.	External	Both	Create	Continuous	The detection time will be system dependent. Refer to the datasheet of off-chip hardware.	Interrupt	a. PMIC Switches off the power supply to MCU (when no. of resets is >= 3)
5	SM4.AIPS_SWCHECK	Interconnects	Diagnostic	The application software ensures that a transaction is not lost or corrupted within AIPS. This safety mechanism requires the software to read registers from safety relevant peripherals mapped on AIPS peripheral space and check their values for correctness to detect faults in the path from core to these registers. The application software needs to periodically check the contents of configuration registers (more than 10 registers) of safety relevant peripherals mapped on AIPS peripheral space to detect faults in the peripheral bridge path. This readback test needs to be run every FHTI.	Internal	SW	Create	Periodic	Based on FTTI of the system Based on MPFDTI of the system	Polling	Destructive reset of MCU
6	SM4.eDMA_SW_CHECK	Memories and mem	Diagnostic	The eDMA controller is checked by system-level checks when transferring safety-related data using the eDMA module. The eDMA controller performs complex data transfers with minimal intervention from the host processor. The eDMA can be used to transfer data from memory/peripherals to memory/peripherals. This mechanism includes the following checks by application software: • CRC check when transferring safety-related data using the eDMA module • checks to ensure the correctness of the eDMA data transfer • checks to ensure that the errors reported by the eDMA module are resolved	Internal	Both	Create	Periodic	Based on FTTI of the system	Polling	Destructive reset of MCU
7	SM4.FLASH.PGM_ERS_CONTENT	Flash Controller	Diagnostic	After performing flash-memory programming operations, software checks that the programmed contents have been updated as expected. This check ensures that the Program and Erase State Machine (PESM) in Flash is able to update the flash blocks with	Internal	Both	Create	Once Per Drive cycle	NA	Polling	MCU shall not Boot-up and execute safety critical applic ation

Drivvize

DRIVE. VISION. REALIZE.

Thank You