Making ISO26262 Functional Safety Verification a Natural Extension of Functional Verification

Andrew Betts, Verification Engineer, Arm

Ann Keffer, Product Management Director, Functional Safety, Cadence Design Systems

accellera SYSTEMS INITIATIVE

cadence

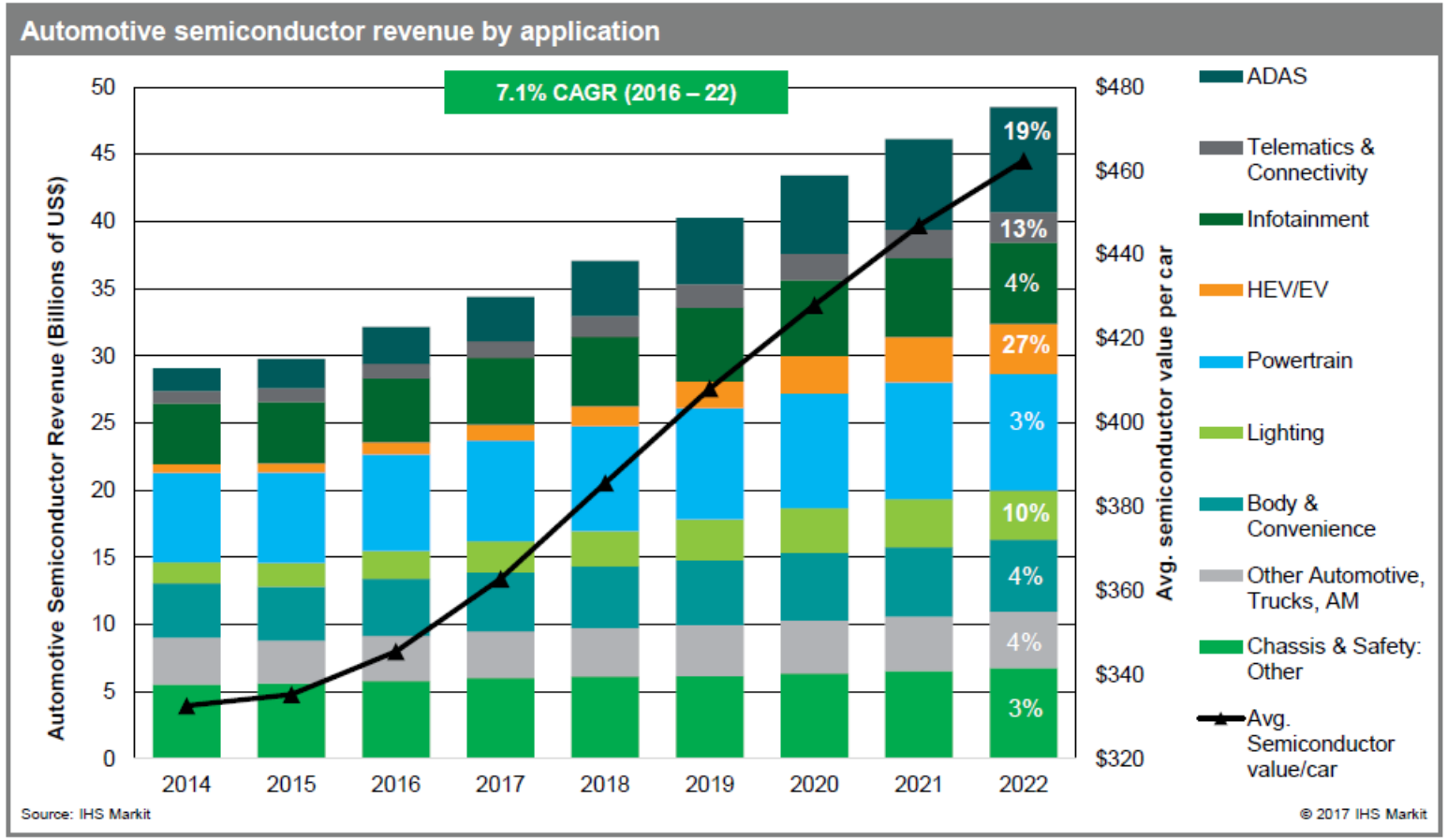
arm

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

Agenda

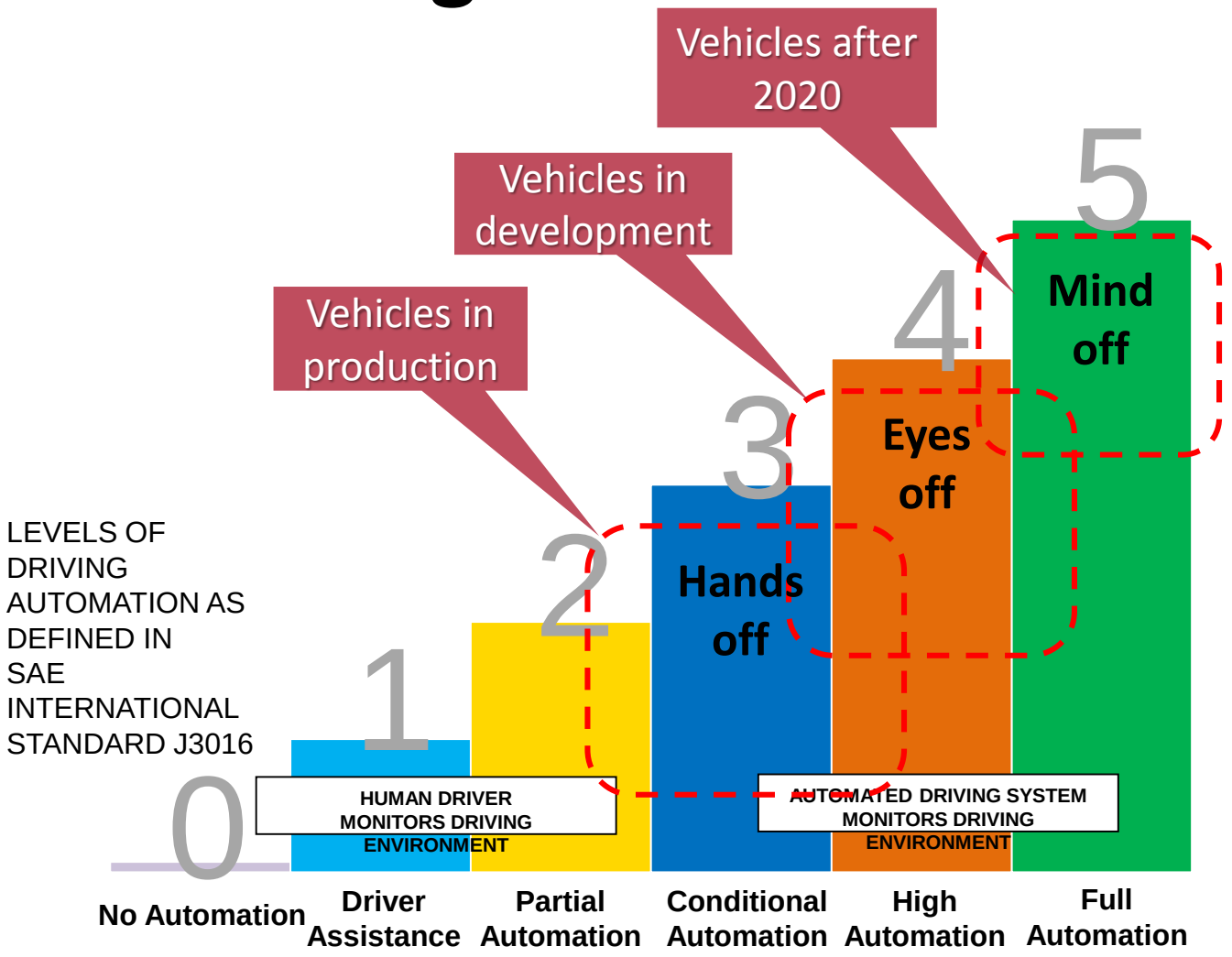
- Market landscape
- Introduction into functional safety
- Why the FMEDA is important
- Functional verification and Safety verification
- Arm® processor design description
- Arm Safety Package
- FMEDA Methodology
- Importance of safety campaigns for Arm's STL
- Validation of the FMEDA
- Arm's safety flow
- Q&A
- Demo

Automotive Semiconductor Growth

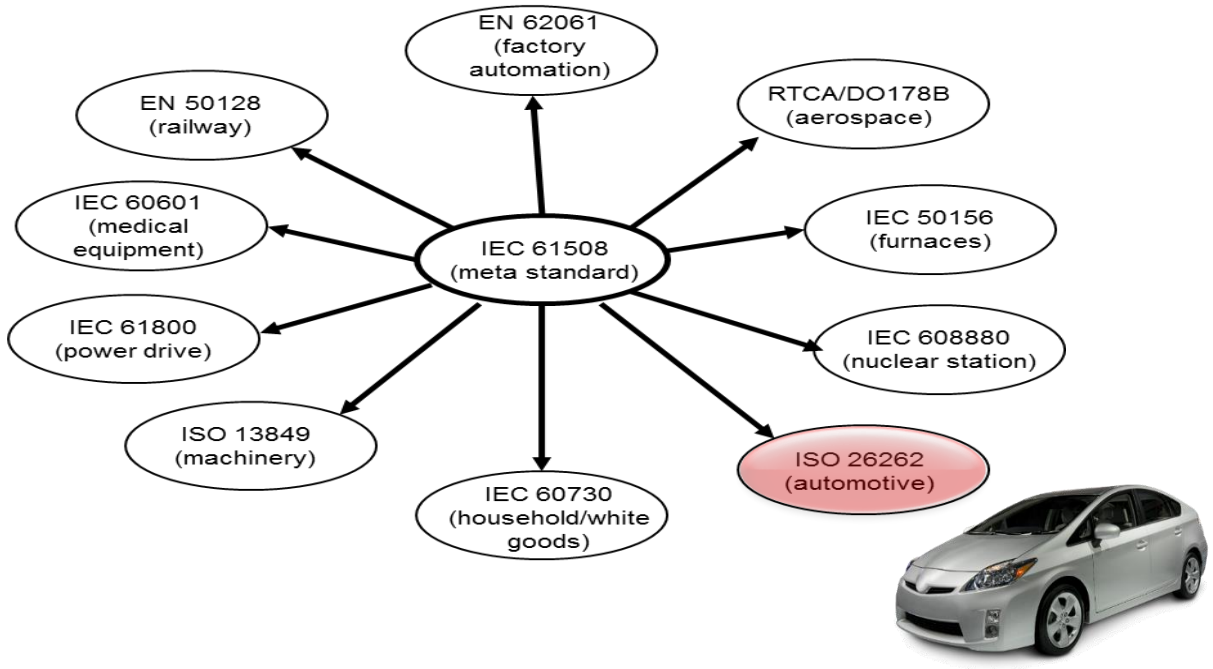


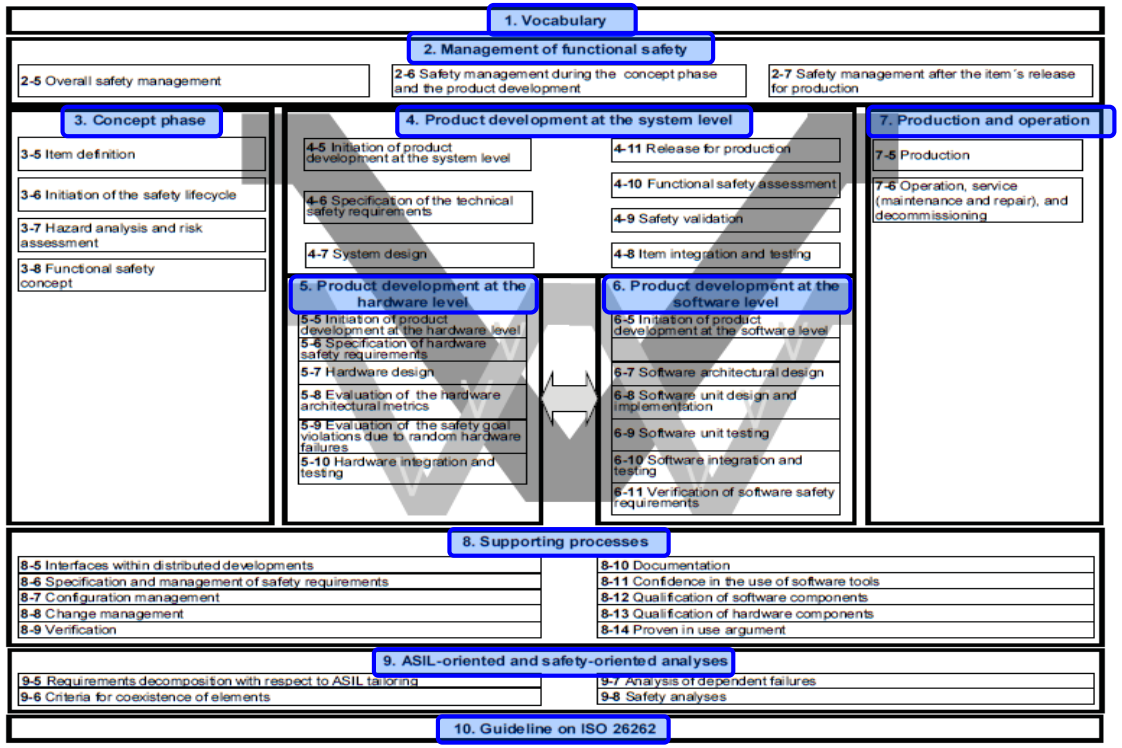
Autonomous Driving

- Amount of electronics is growing fast
- (ADAS) based on complex SoCs to enable high-performance computing
- Safety critical ADAS applications have stringent requirements on
  - Functional safety
  - Security
  - Reliability



Functional Safety Standards



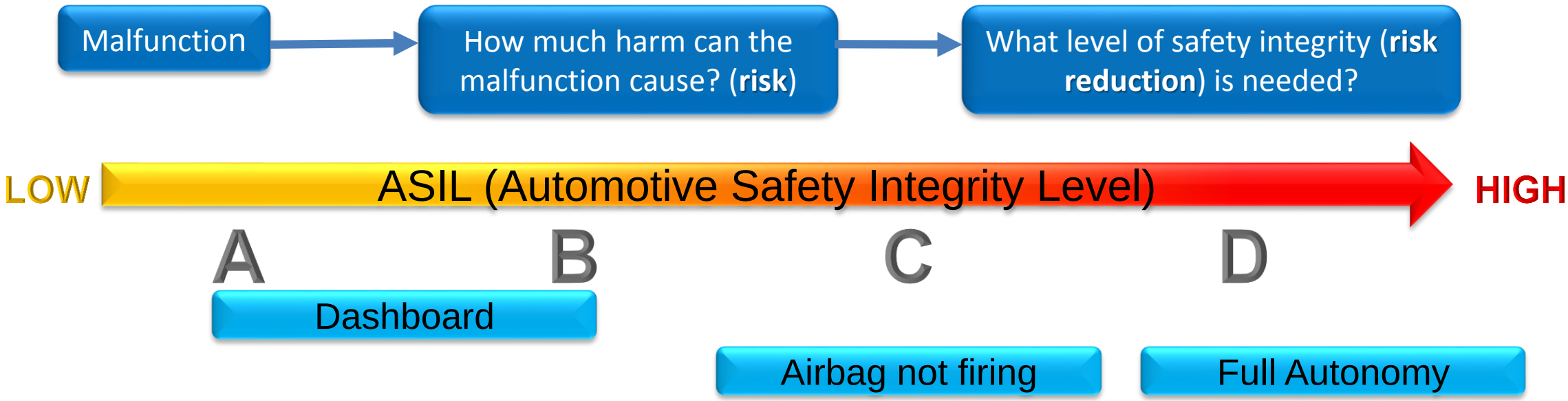


ISO 26262 defines

- Processes to follow
- Hardware/software performance to achieve
- Safety documentation to produce
- Software tools compliance process

Functional Safety Definition—ISO 26262

"Absence of unreasonable risk due to hazards caused by malfunctioning behavior of electrical and/or electronic systems" (ISO 26262)



ASIL examples for illustration purposes only

ASIL Determination Example—ISO 26262

Malfunction	ABS system failure	⟺	Safety Goal: **Prevent ABS failure**
Hazard Analysis	What unintended situations (hazards) could happen? → Loss of stability on split-μ surface		
Risk Analysis	• **How likely** is the hazard to happen? (Exposure) → oil spill, gravel, water potholes, …. • **How harmful** is the hazard? (Severity) → Car may spin out of control and crash • **How controllable** is the system if the hazard occur? (Controllability) → dashboard, driver		
ASIL Determination	What level of safety (risk reduction) does the system need? • How likely can the malfunction be? → **FIT (Failure in Time)** • How often does the system need to catch it and get to a safe situation? → **DC (Diagnostic coverage)**		

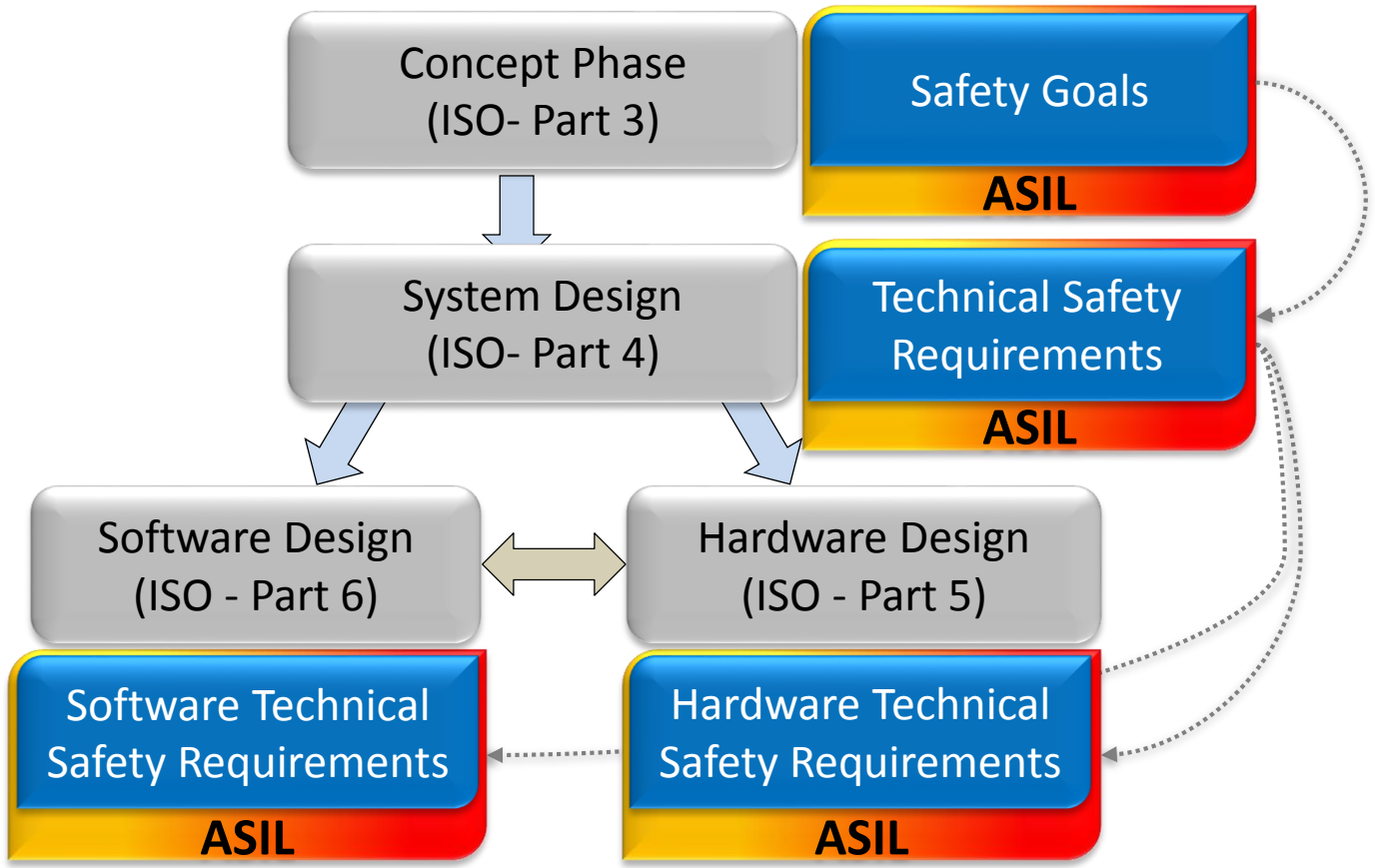
LOW — ASIL (Automotive Safety Integrity Level) → HIGH

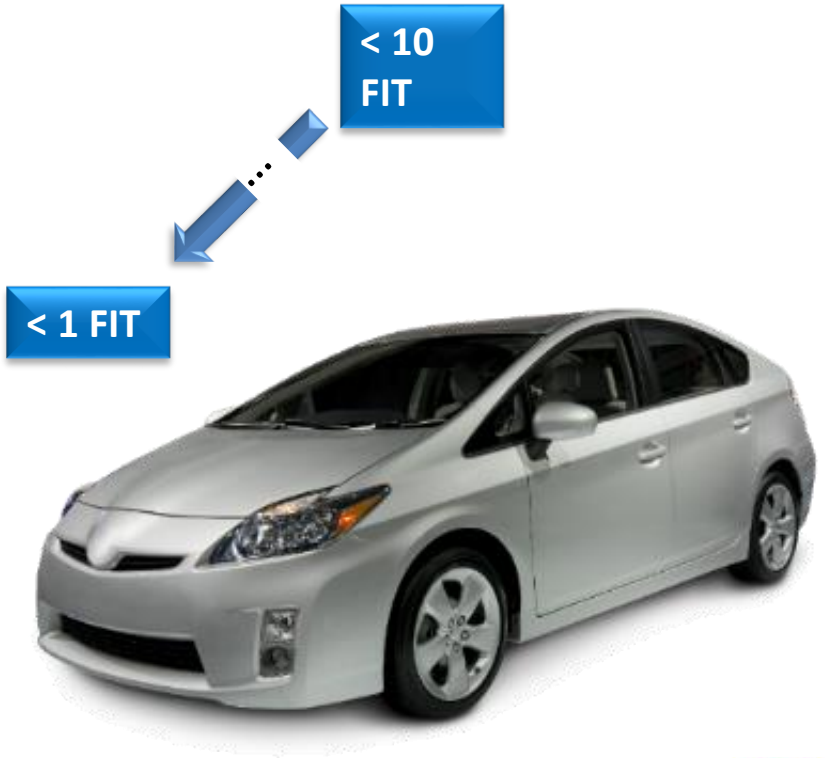
A B C D

↓ FIT (Failure In Time), ↑ Diagnostic Coverage (DC)

Design and Safety Flow

ISO 26262





Failures Relevant to Functional Safety

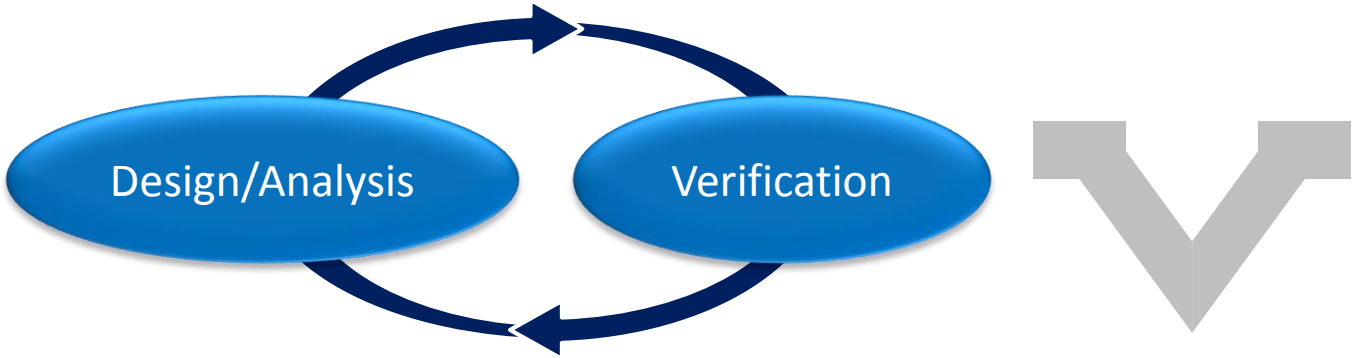
ISO26262—Functional Safety Principles

Systematic Failures
(e.g., software bug)

- Addressed by processes (planning, traceability, documentation, specs)
- Strictness of processes are dependent on the ASIL level

Random Failures
(e.g., component malfunction, noise injection)

- Considers permanent failure and transient effects
- Includes **safety mechanisms** design and integration to handle faults
- Demonstrated by calculations of Reliability/verification of failure rates
- Failure rates and diagnostic coverage requirement depend on ASIL

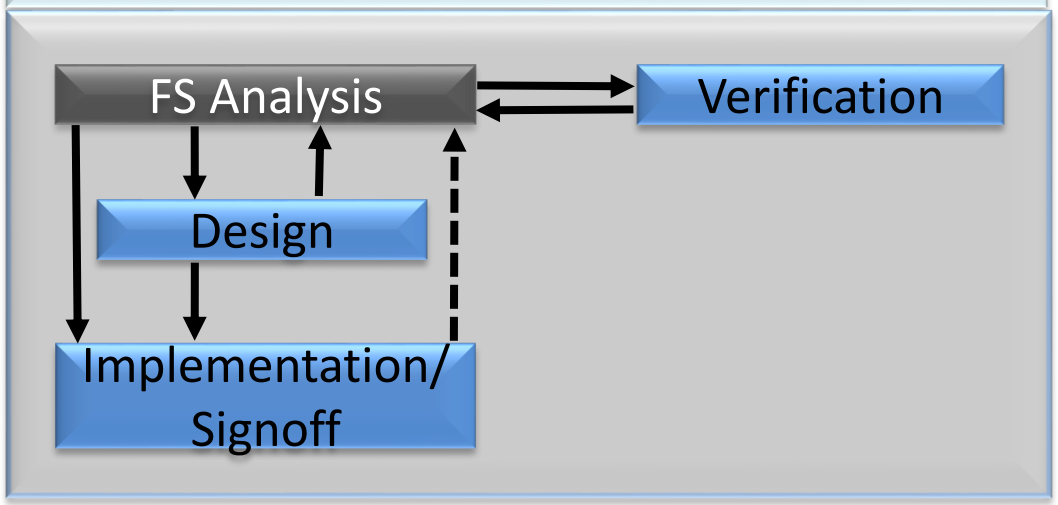


What is Functional Safety Analysis?

- Define Failure Modes (FM)
- Determine Safety Mechanisms (SM)
- Validate Single Point Failure Metric (SPFM) and Latent Failure Metric (LFM)

How to improve your HW metric (to achieve the target ASIL):

- Better component
- Better/Additional Safety Mechanism



Defining Functional Safety

Structured Approach to Measure the ASIL HW Metrics

Safety is the freedom from unacceptable risk of physical injury or damage due to unplanned or undesired events

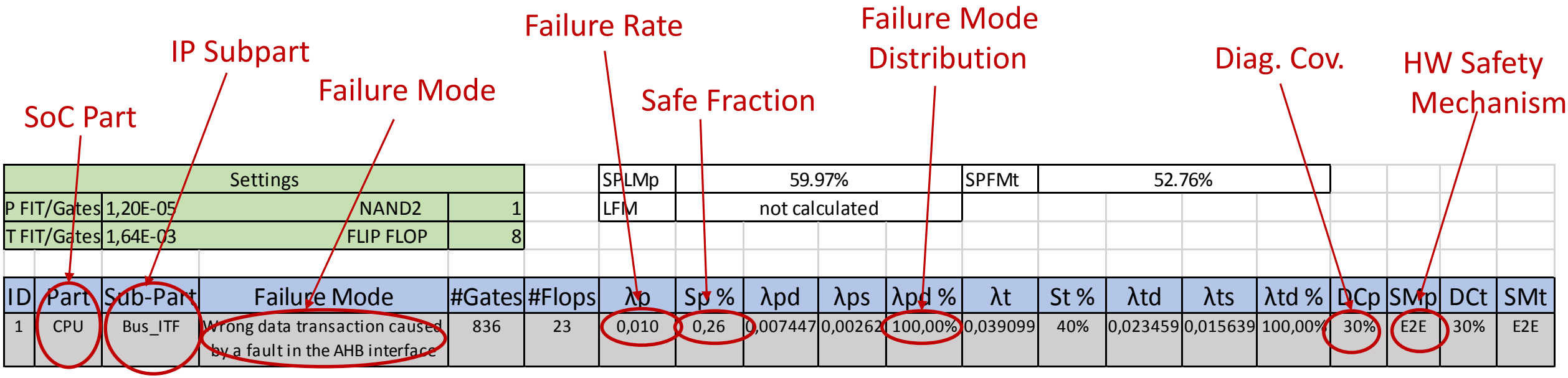
ASIL	Failure Rate	Single Point Failure Mode (SPFM)	Latent Failure Mode (LFM)
A	< 1000 FIT	Not relevant	Not Relevant
B	< 100 FIT	≥ 90%	≥ 60%
C	< 100 FIT	≥ 97%	≥ 80%
D	< 10 FIT	≥ 99%	≥ 90%

Importance of Failure Mode Effects and Diagnositc Analysis (FMEDA)

- The FMEDA is safety analysis required by ISO26262
  - Other analysis are FMEA, FTA, DFA,....

- FMEDA is needed to:
  - Verify the number of Safety Mechanisms and their claimed diagnostic coverage properties are enough to reach the required ASIL level calculating the architectural safety metrics: SPFM, LFM
  - Validates the Safety Architecture (collection of safety mechanisms) and calculates the performance of the system (SPFM, LFM)

FMEDA – Capture and Analyze Safety Goals

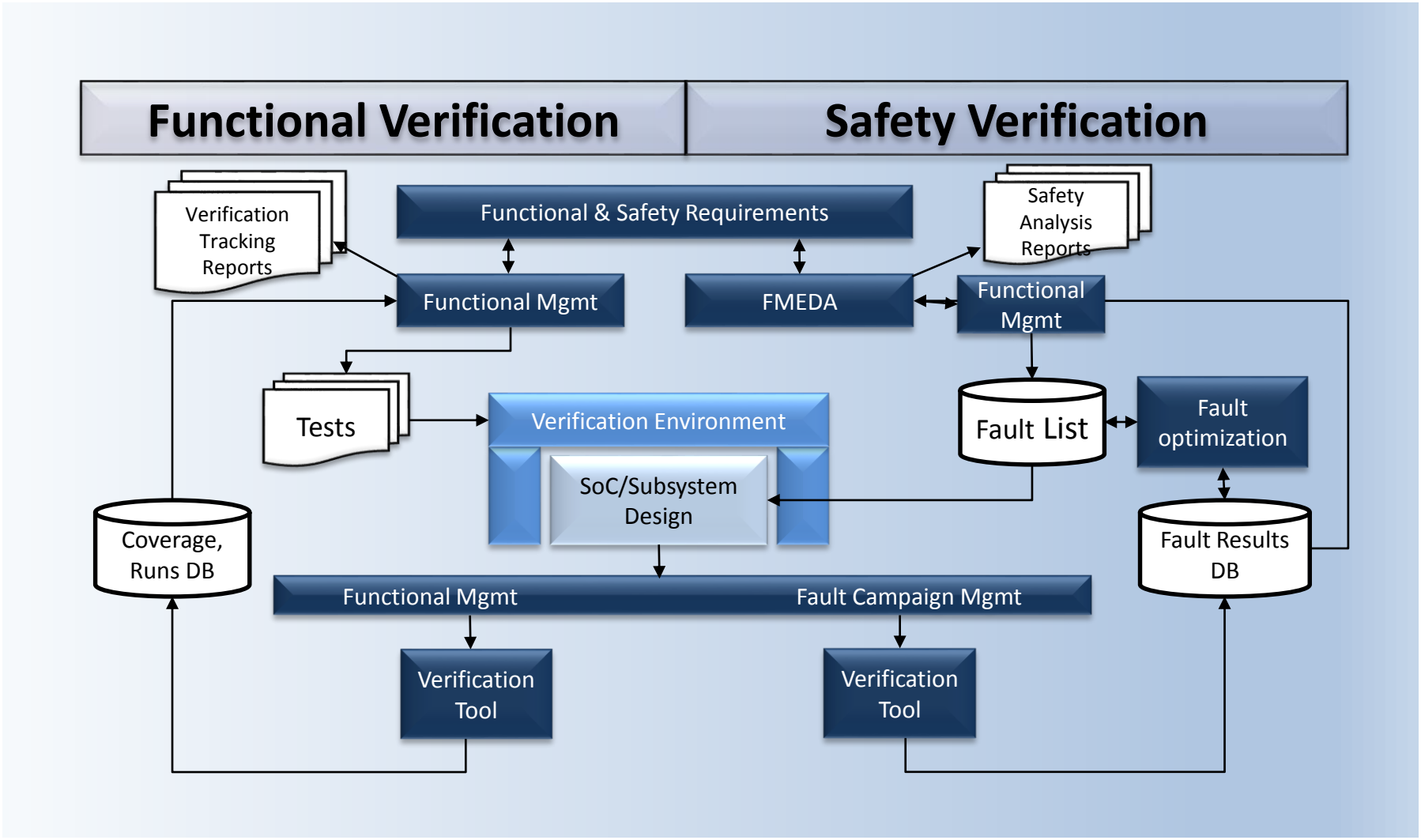


Settings			
P FIT/Gates	1,20E-05	NAND2	1
T FIT/Gates	1,64E-03	FLIP FLOP	8

SPLMp	59.97%	SPFMt	52.76%
LFM	not calculated		

ID	Part	Sub-Part	Failure Mode	#Gates	#Flops	λp	Sp %	λpd	λps	λpd %	λt	St %	λtd	λts	λtd %	DCp	SMp	DCt	SMt
1	CPU	Bus_ITF	Wrong data transaction caused by a fault in the AHB interface	836	23	0,010	0,26	0,007447	0,00262	100,00%	0,039099	40%	0,023459	0,015639	100,00%	30%	E2E	30%	E2E

Safety Verification Flow



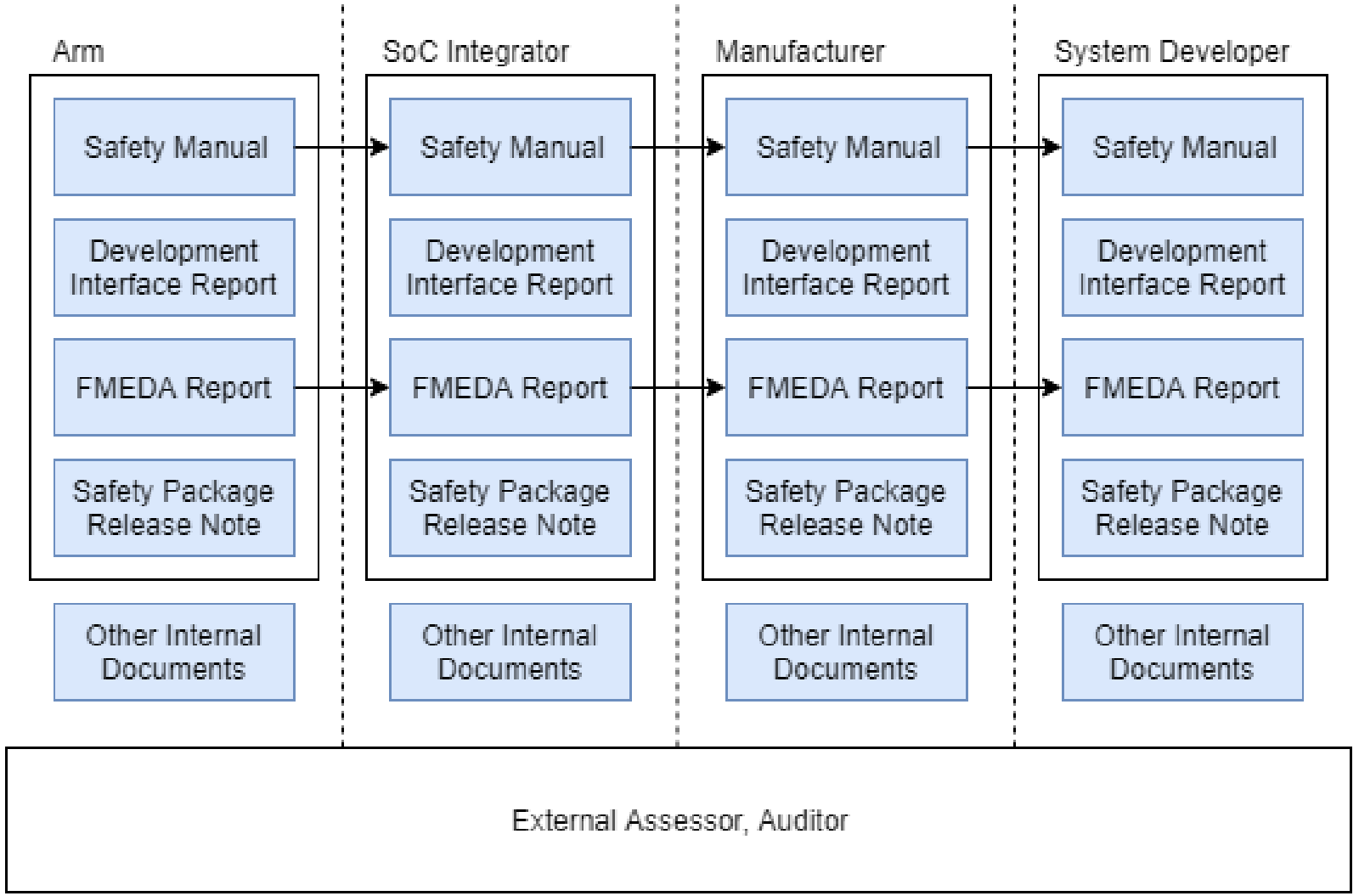
Arm Functional Safety Design Description

- Functional Safety is critical to many Arm products, especially those targeting automotive segment

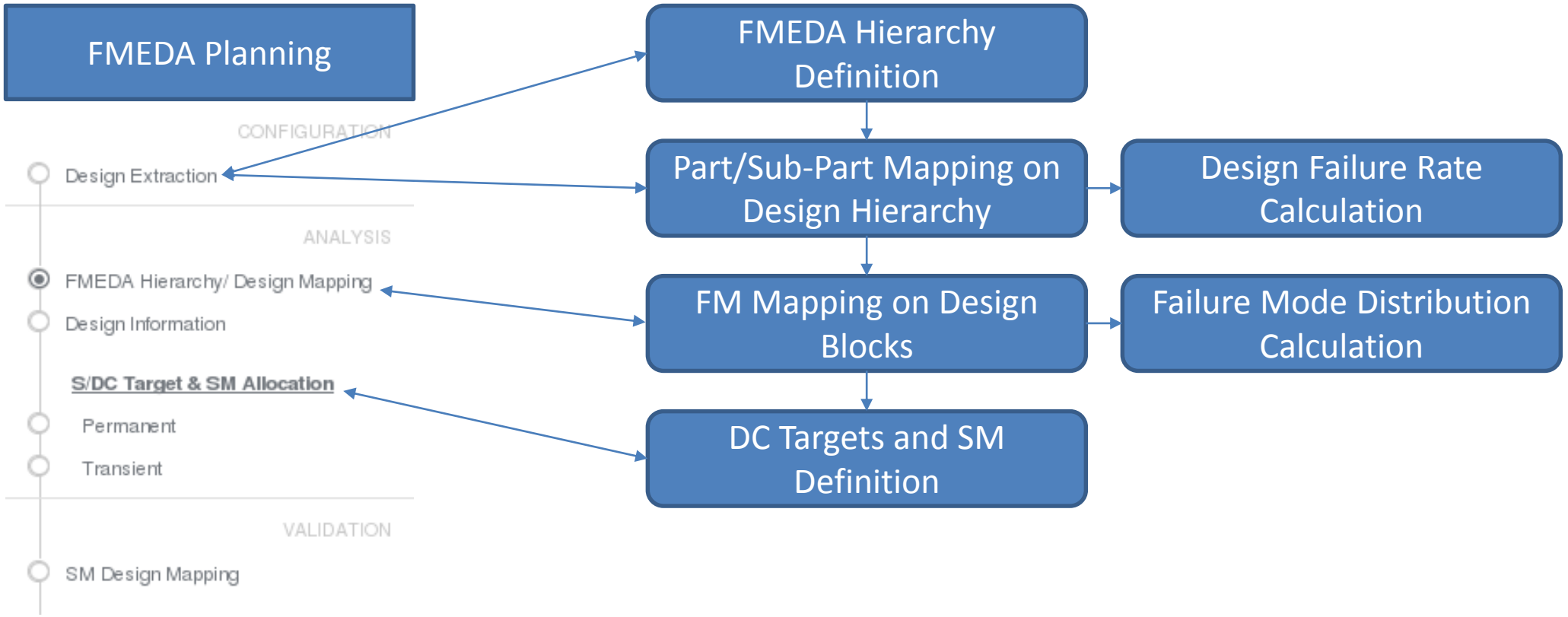
- Arm has many safety packages already:
  - Arm Cortex®-A53, A57, A72, A75, A76, A35, A32, A34, A55, R5F, R52, M3, M4, M0+, M7, M33, M23

- Arm's safety analysis is for a Safety Element out of Context (SEooC)

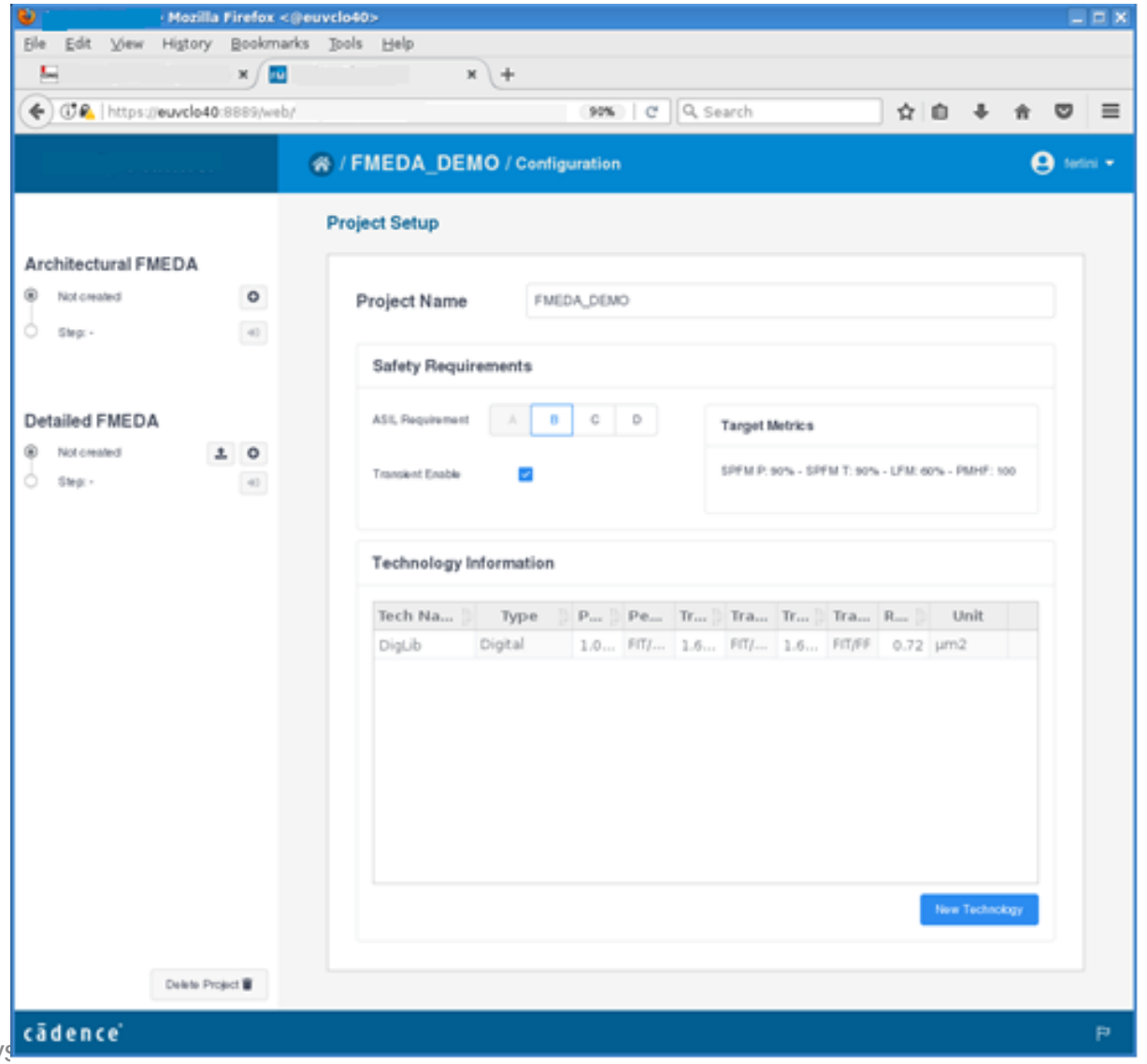
Arm Safety Documentation Package



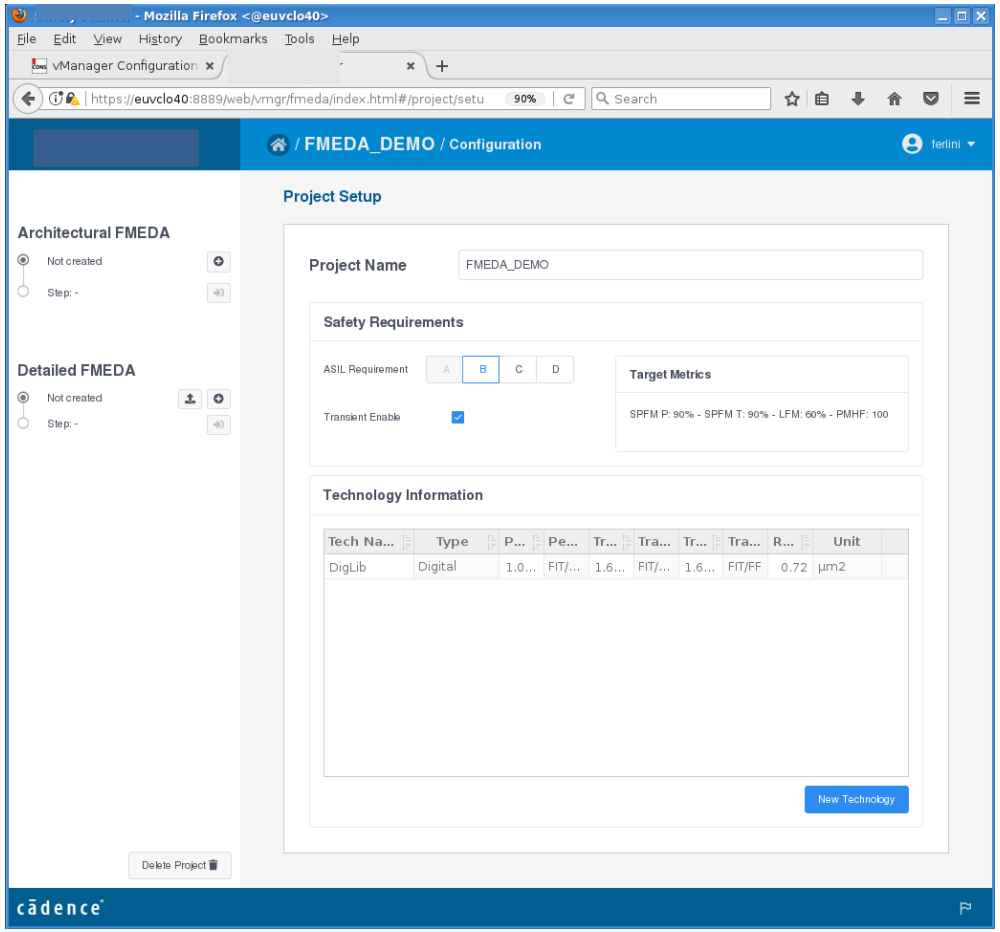
FMEDA Methodology



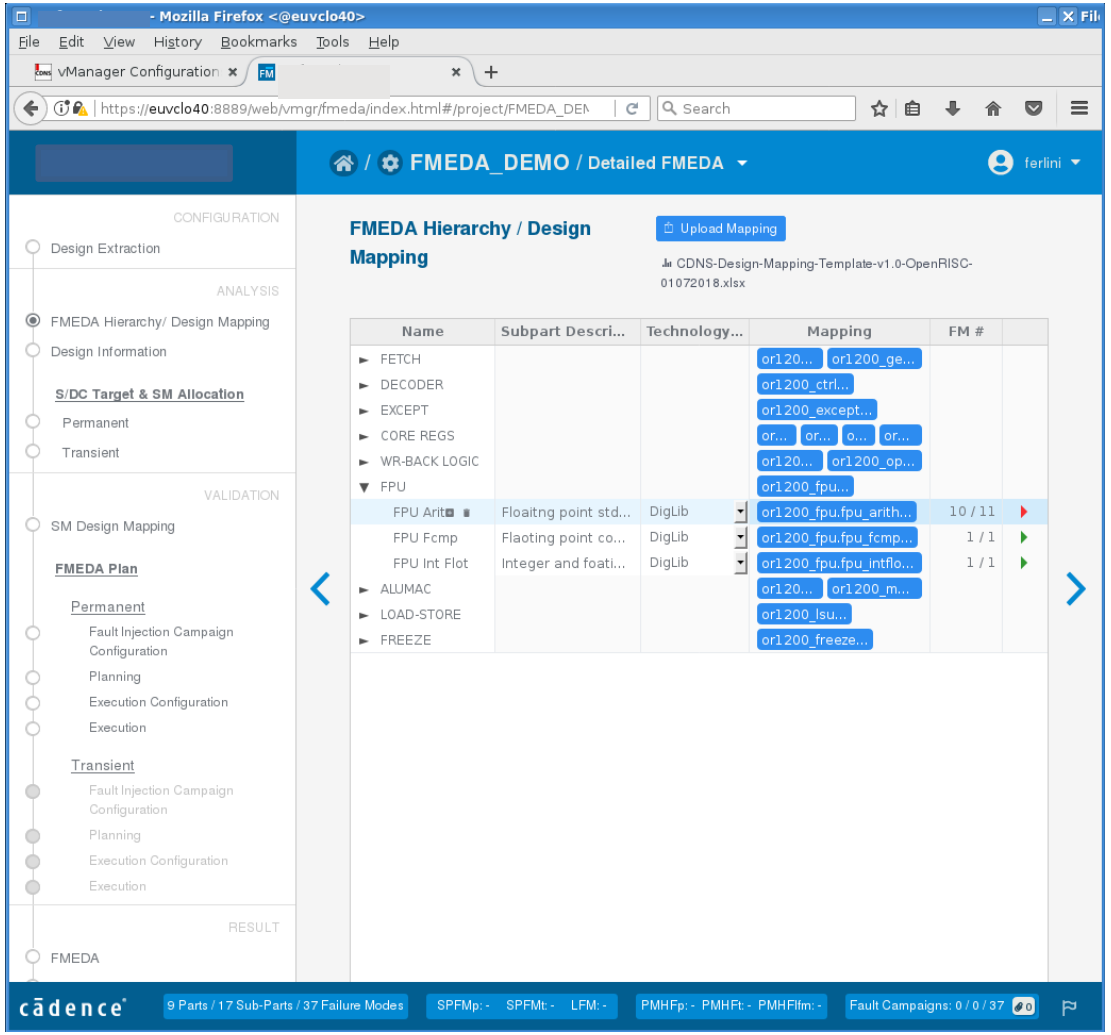
FMEDA Analytics



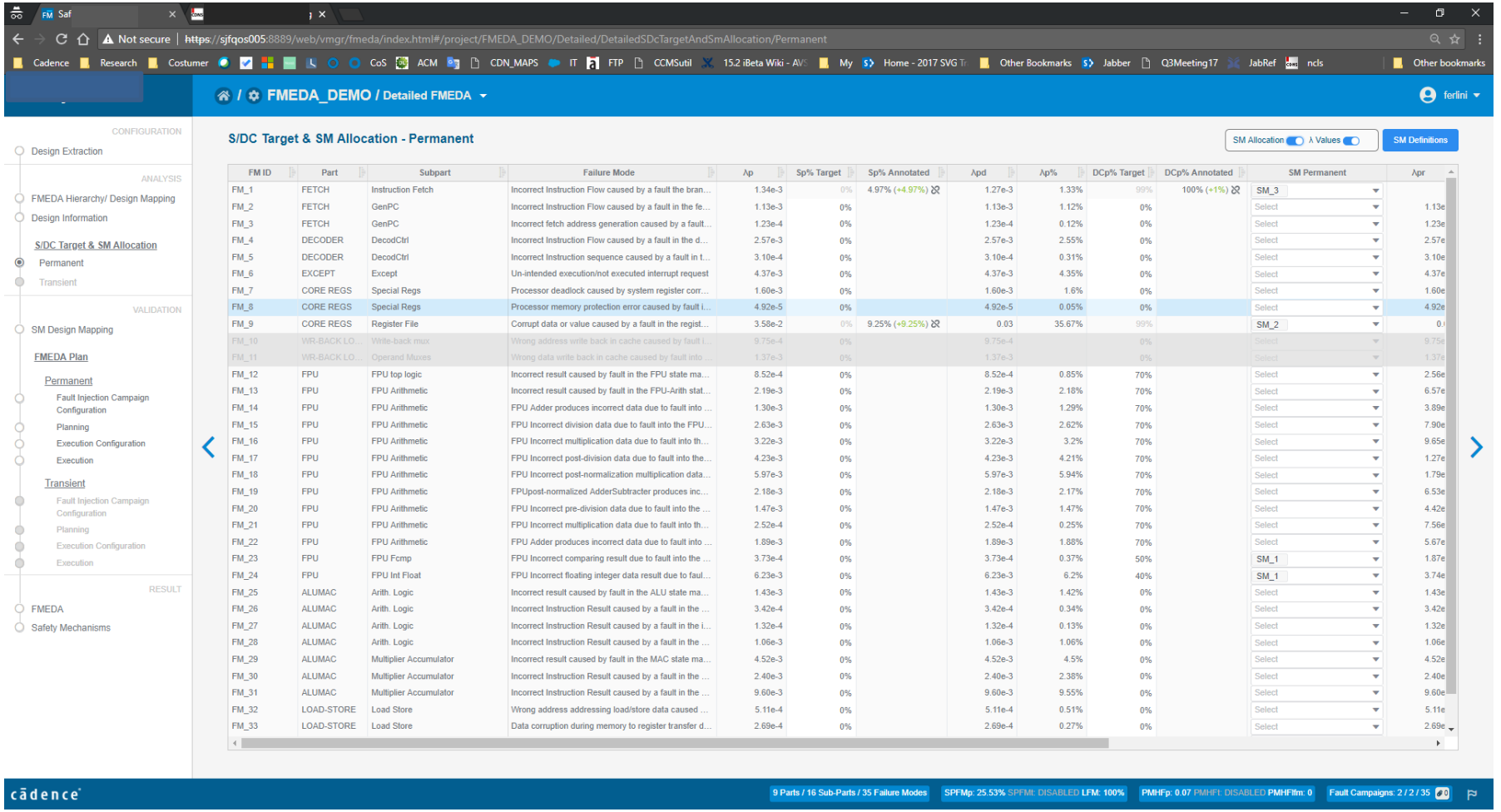
FMEDA Methodology



Design Mapping



S/CD Target and SM Allocation



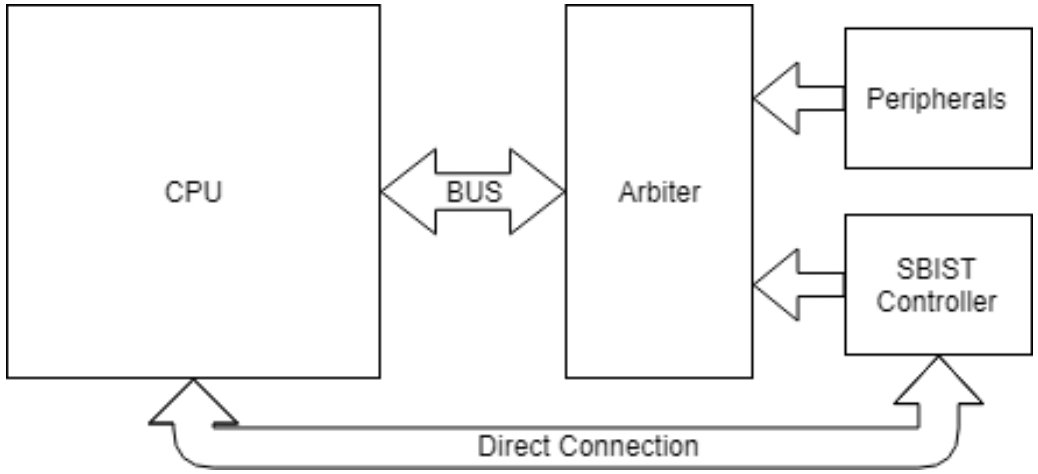
STL and Safety Mechanisms

The Importance of Fault Injection Campaigns on the STL

- Software Test Library (STL)

- It is intended to be run on a safety critical processor, and attempt to detect any errors
  - This can allow the system to fail safely

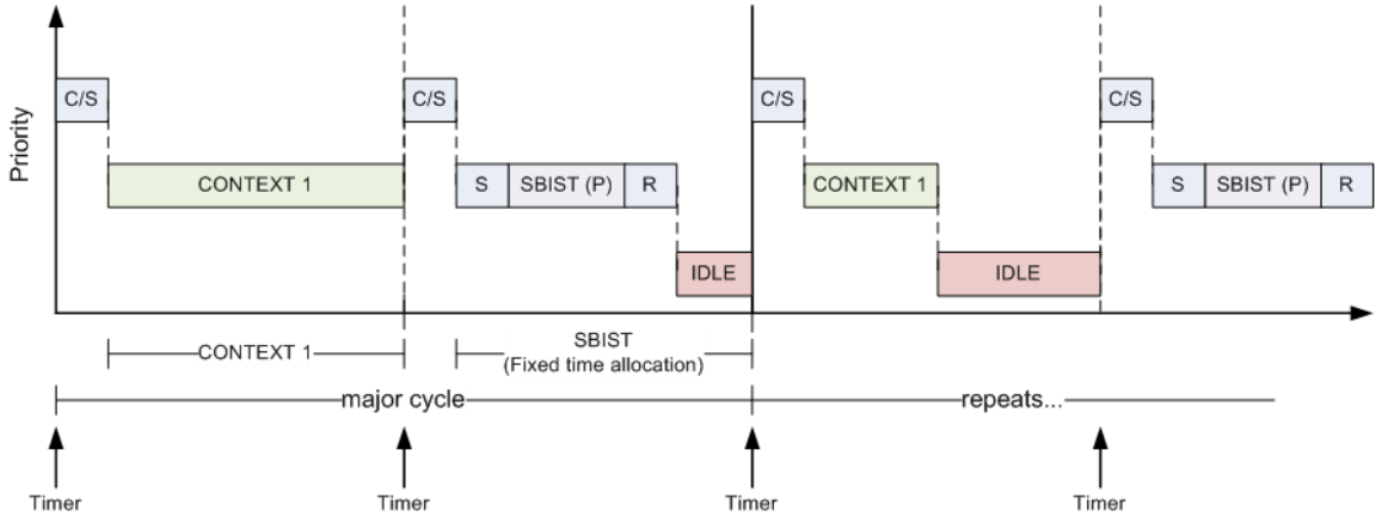
- The design of the processor is created with the STL in mind
  - Optimisations enable software to test certain parts and be deterministic



STL and Safety Mechanisms

The Importance of Fault Injection Campaigns on the STL (continued)

- STL is very low level
  - Written in C and assembly
  - Contains a number of functions for testing the “health” of the processor
- Requirements are strict on size and execution time
  - They can be “online” as well as “offline”
- Fault injection used to measure the effectiveness of STL



Example: FMEDA

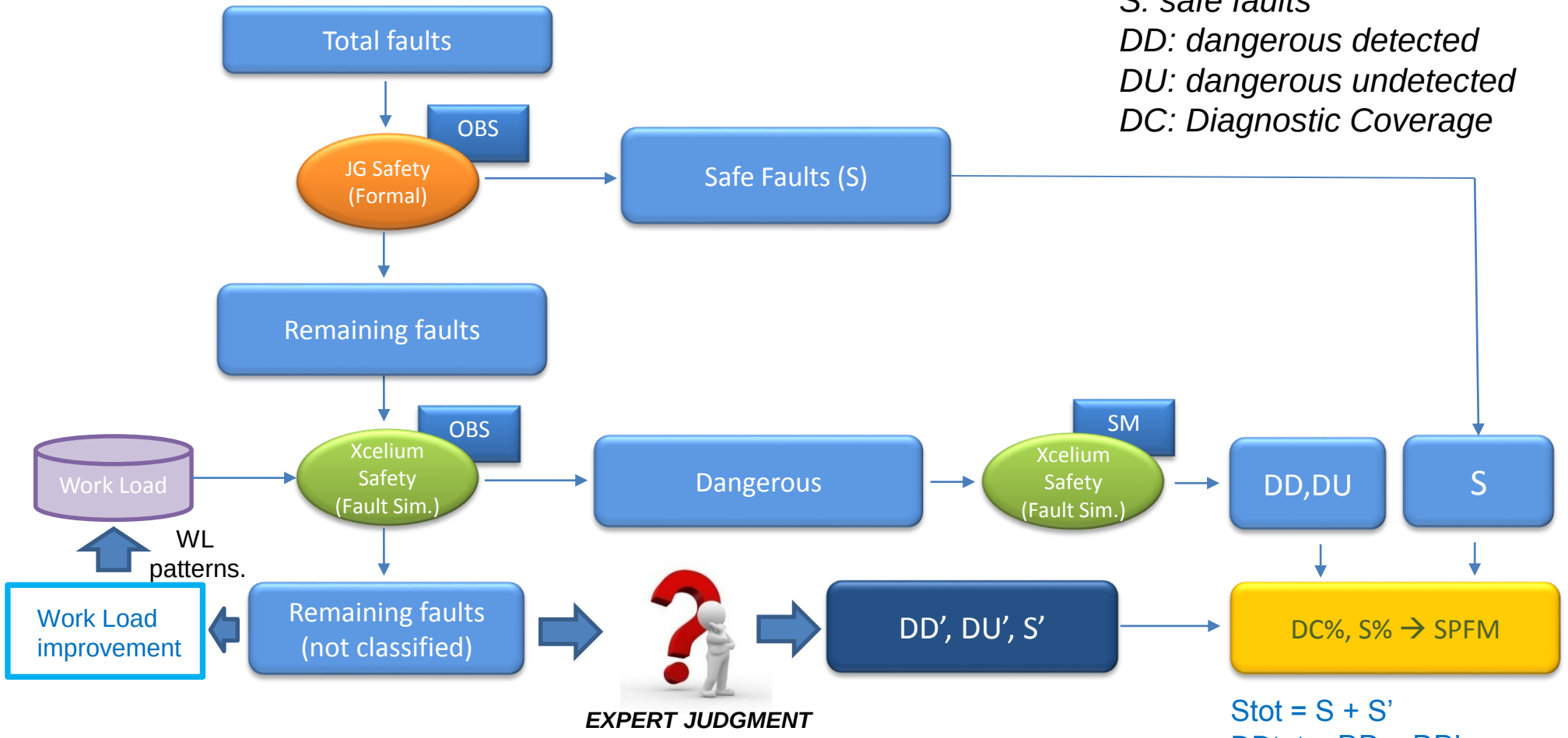
FMEDA reports are done for the processor with the STL, and also for the SBIST controller.

Permanent faults								
FM distribution	Failure rate	Fsafe	Fault detection and control mechanism	Diagnostic coverage (Krf)	Residual / single point failure rate	Multiple point failure rate	Diagnostic coverage for latent faults (Klat)	Latent multiple point failure rate
0.000%	3.57E-04	0.000%	STL	89.240%	3.84E-05	3.19E-04	100.00%	0.00E+00

How to Validate an FMEDA

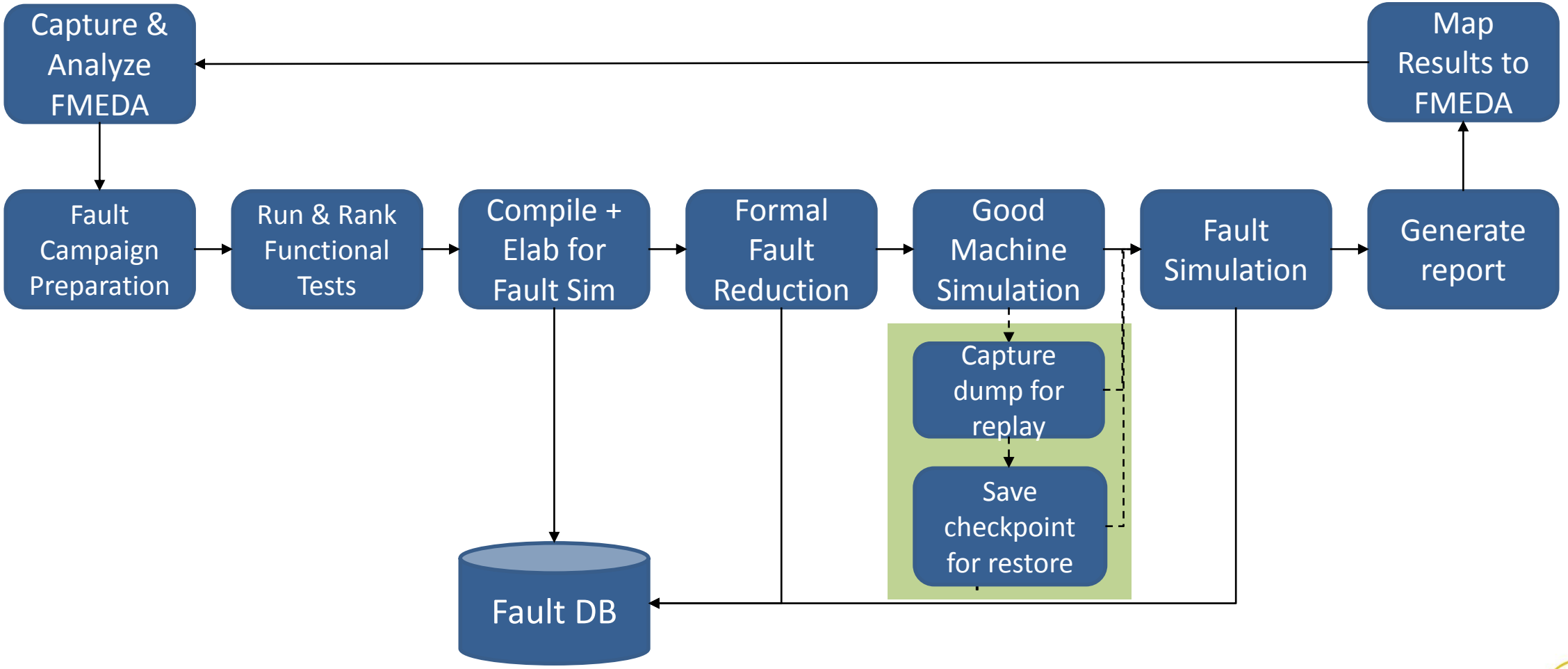
ISO26262 Compliant DC/S Calculation

Agenda:
S: safe faults
DD: dangerous detected
DU: dangerous undetected
DC: Diagnostic Coverage

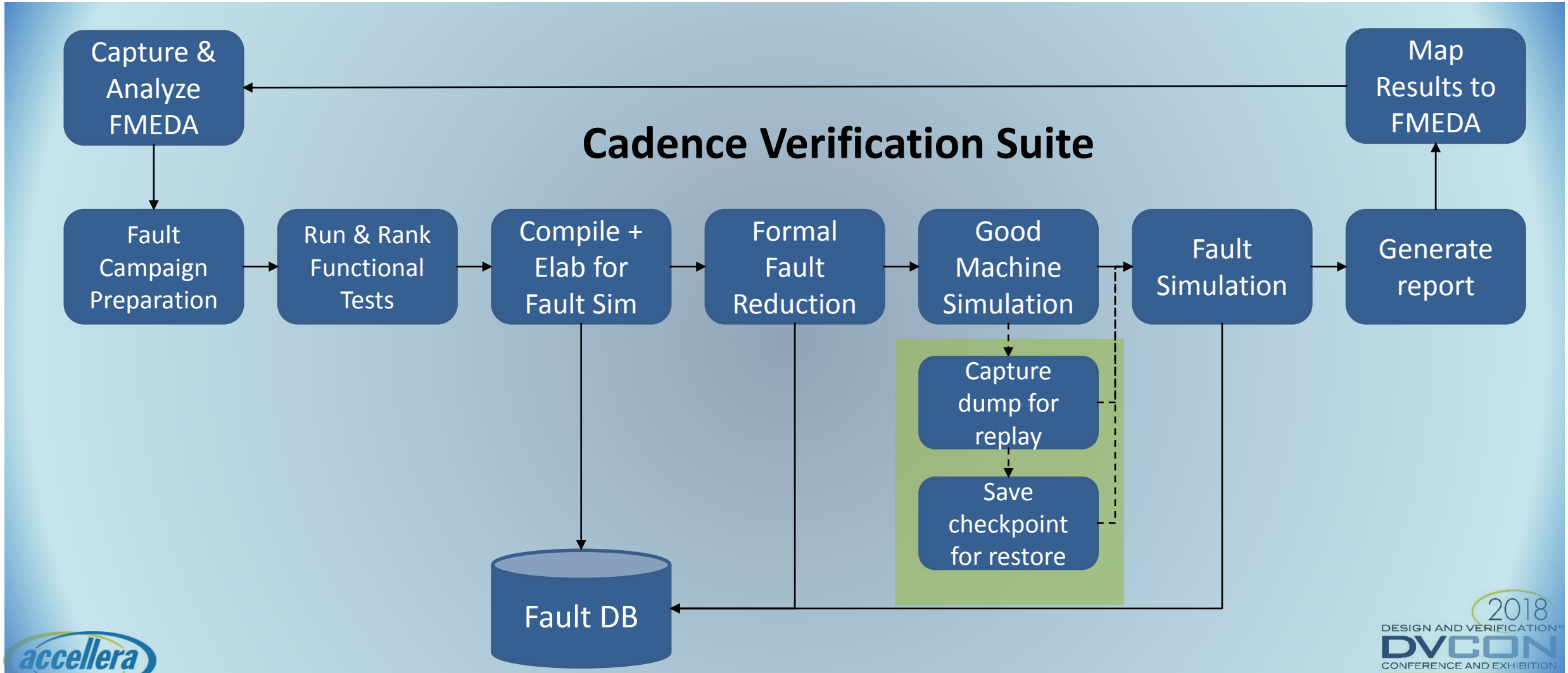


Stot = S + S'
DDtot = DD + DD'
DUtot = DU + DU'

Functional Safety Flow



Cadence Functional Safety Flow



Questions?

Demo

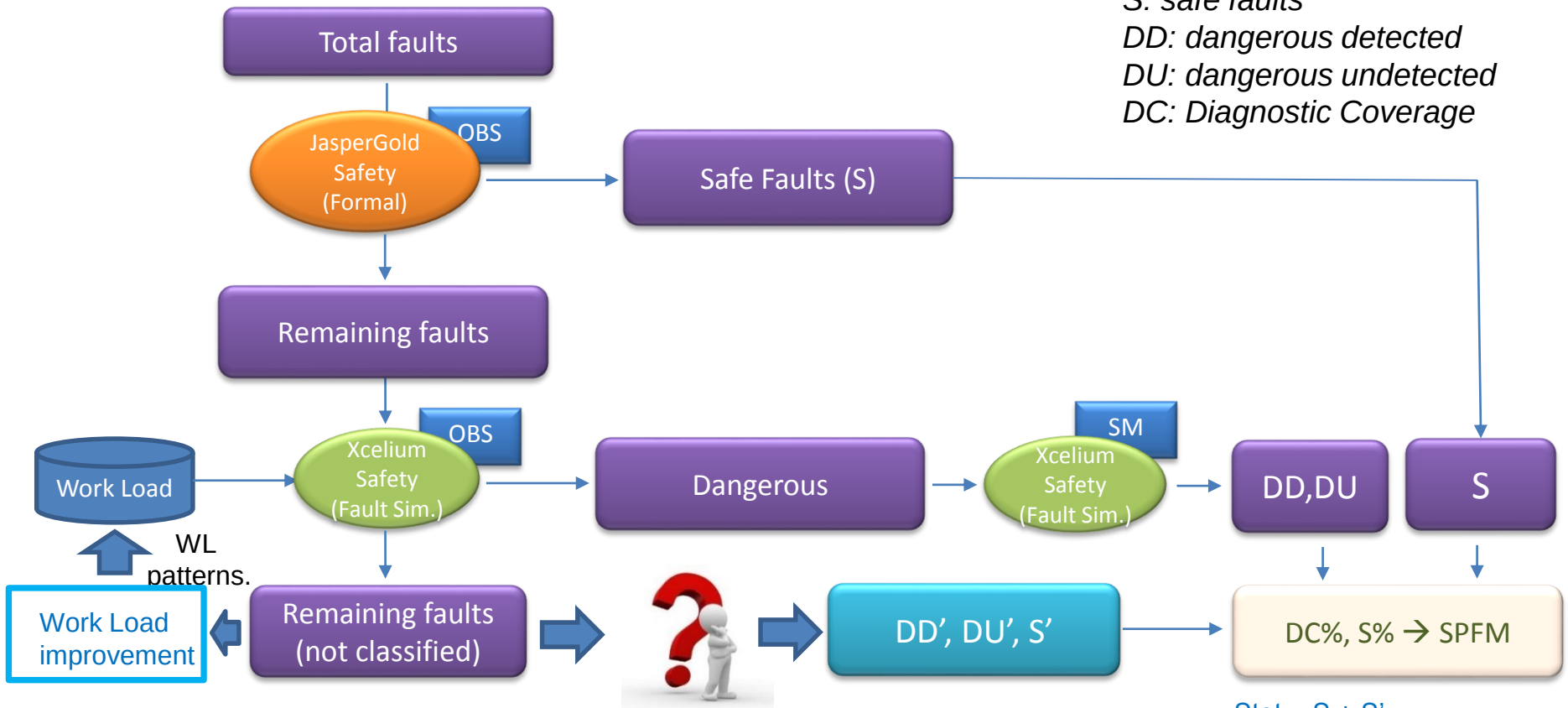
FMEDA Demo

How to Validate an FMEDA – Fault Classification Flow

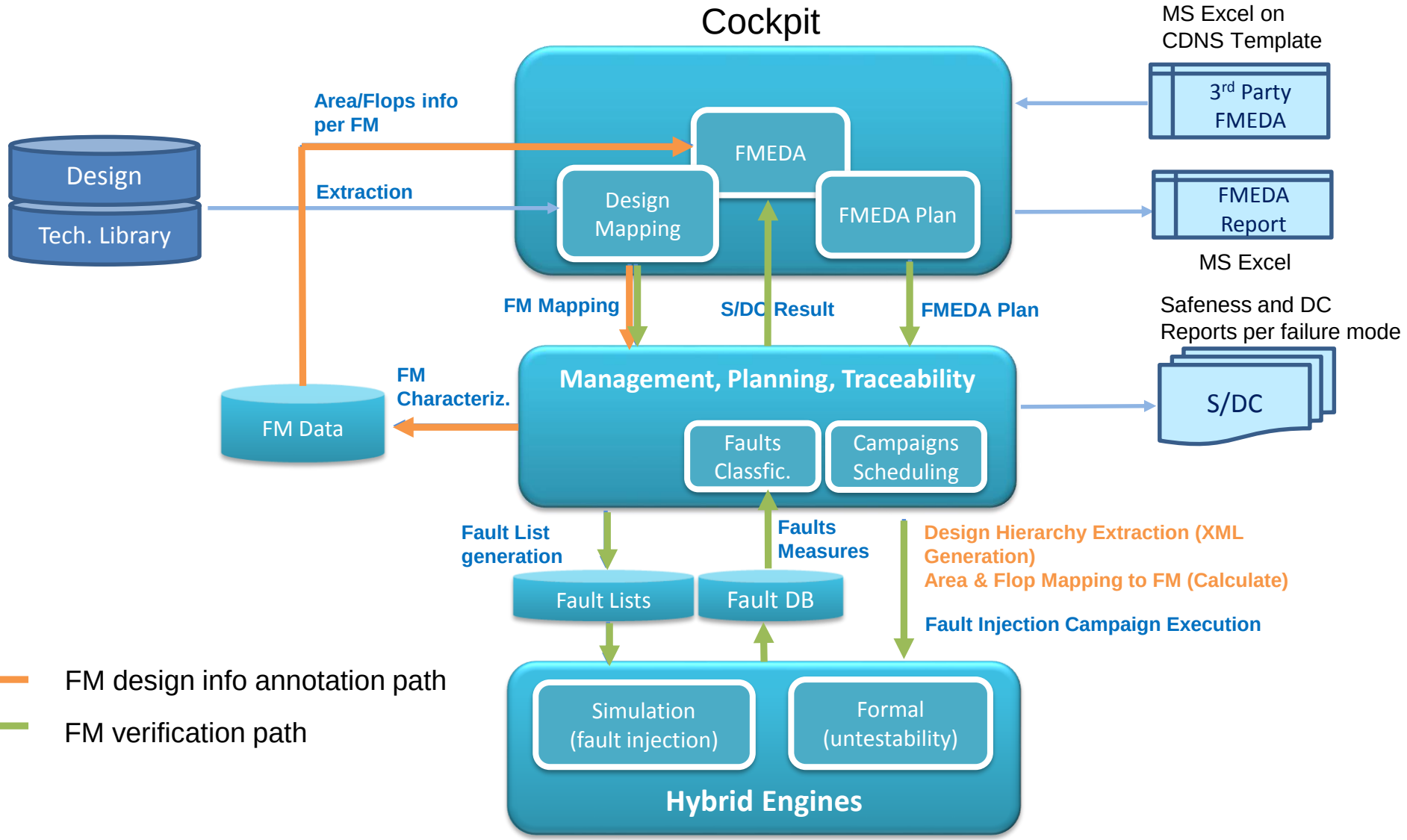
- Fault classification can't be ideal in practice, not classified faults (NC) can be exposed to user expert judgment to be further classified

ISO26262 Compliant DC/S Calculation

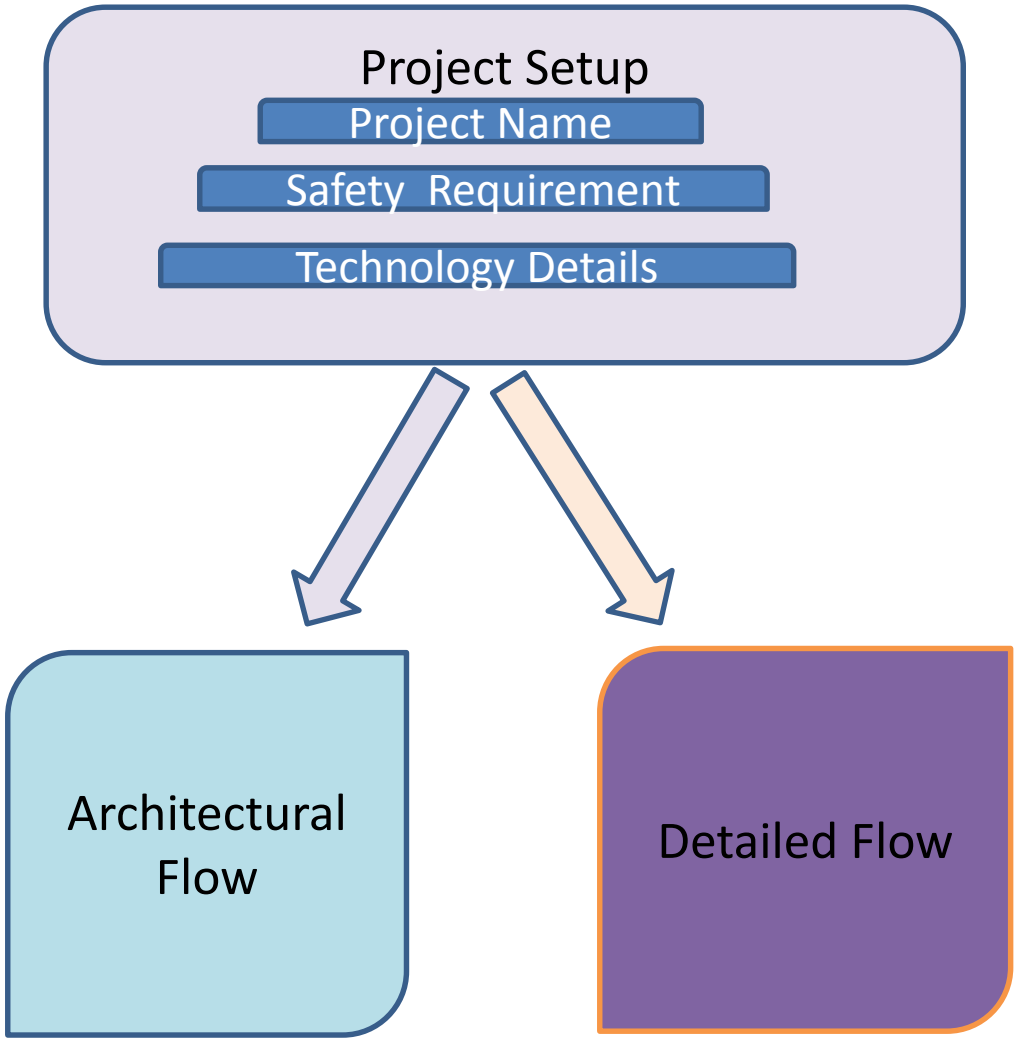
Agenda:
S: safe faults
DD: dangerous detected
DU: dangerous undetected
DC: Diagnostic Coverage



FMEDA Solution – Top Level View



Flow support



Architectural FMEDA Flow:

- Rough estimation of the overall metrics
  - Course grained FMEDA
- Metrics evaluation before RTL/design information availability
- Analysis is to address feasibility study
- Analysis aimed to figure out primary SM requirements.

Detailed FMEDA Flow:

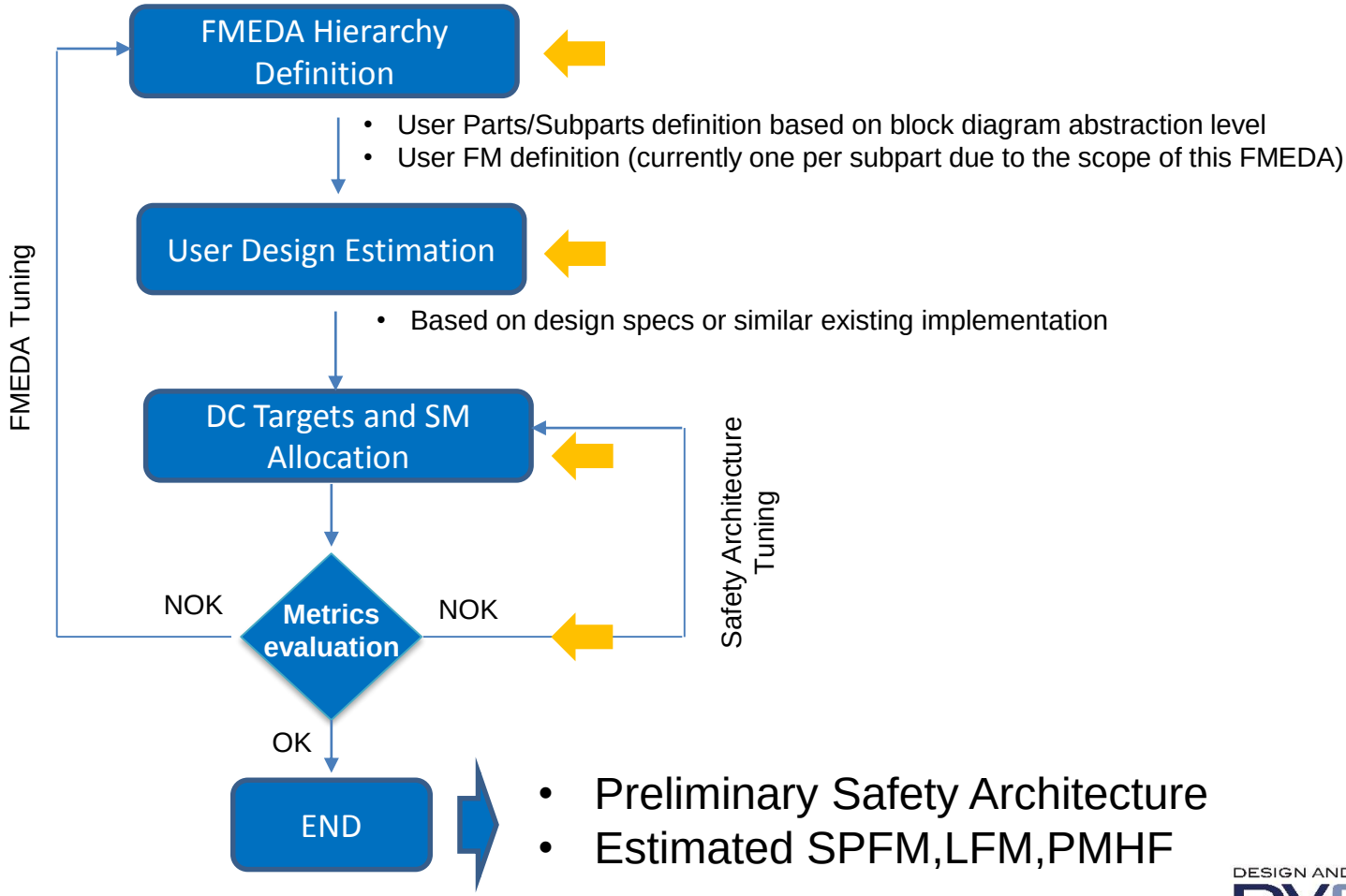
- Accurate estimation of the overall metrics
  - Fine grained FMEDA
- Metrics evaluation using actual design & assumed DC targets
- Evidence to prove the achieved safety goals
  - Data for ASIL certification
- Validation of metrics – Fault Injection, Formal Analysis

FMEDA Solution

Architectural FMEDA: based on Design Estimation (before RTL availability)

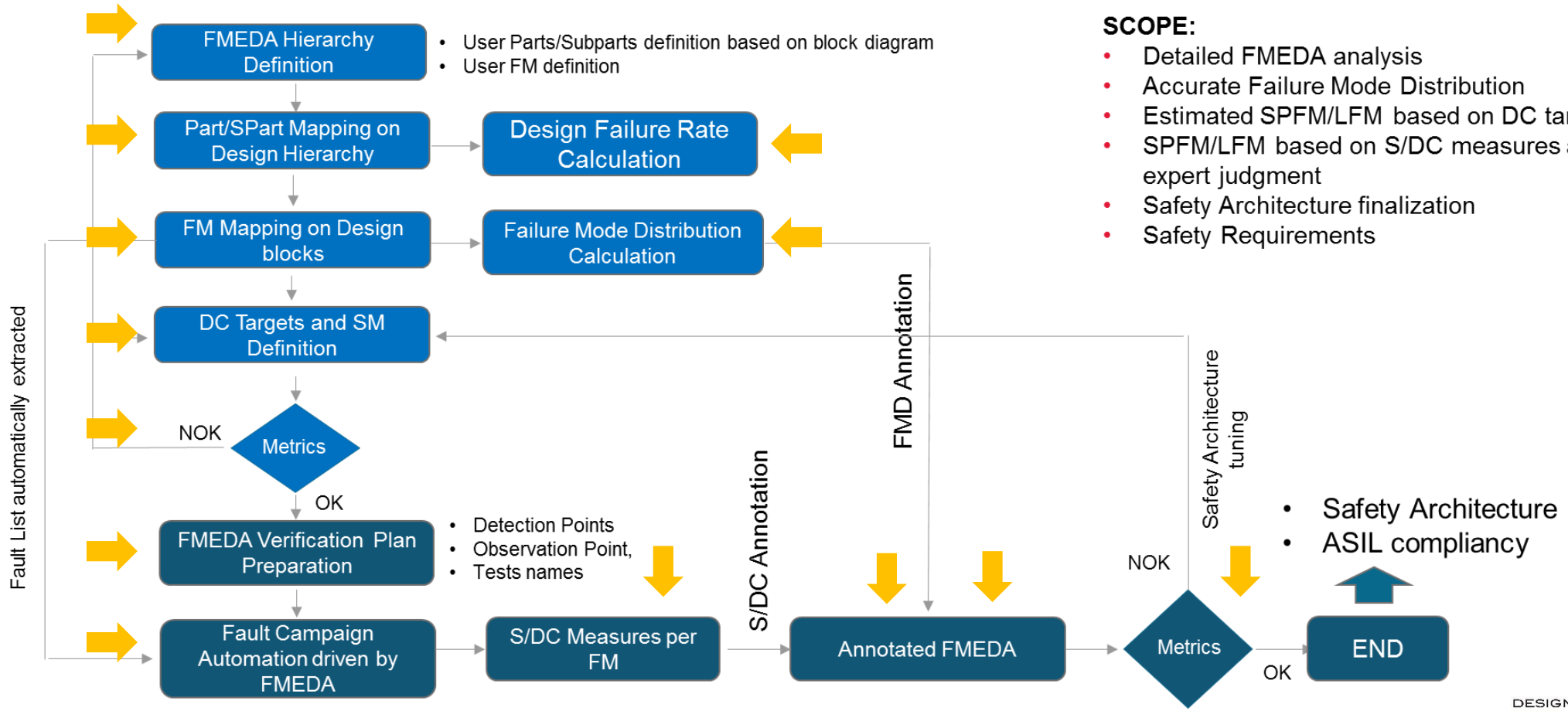
SCOPE:

- Course grained FMEDA analysis
- Preliminary Safety Architecture
- Preliminary Safety Requirements
  - S, DC associated with mapped Safety Mechanism
- Estimated SPFM,LFM, PMHF
  - For Permanent and Transient



FMEDA Solution

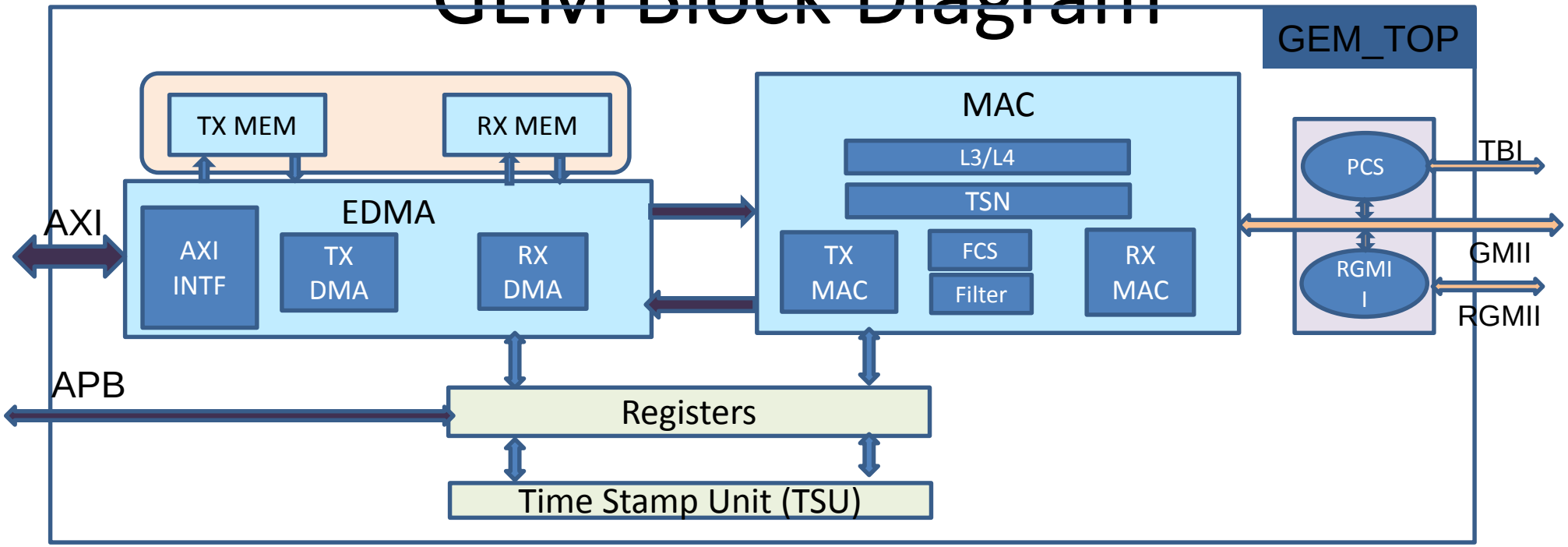
Detailed FMEDA: based an actual design information (after RTL, Netlist availability)

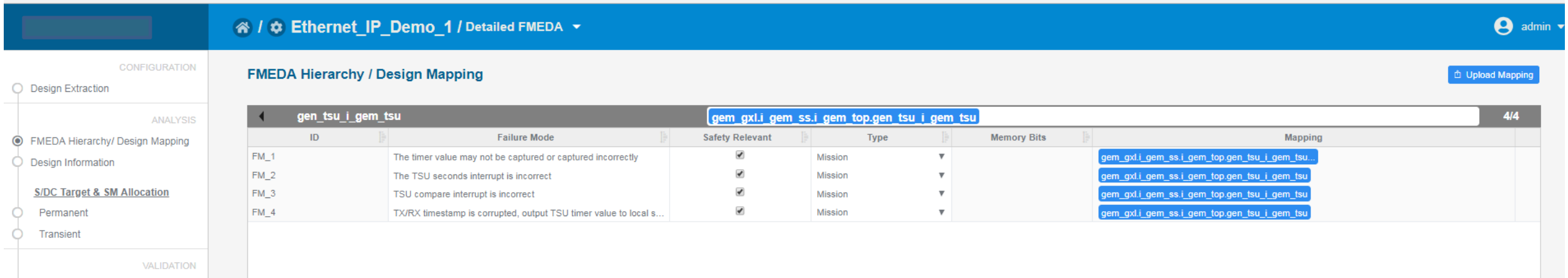


SCOPE:

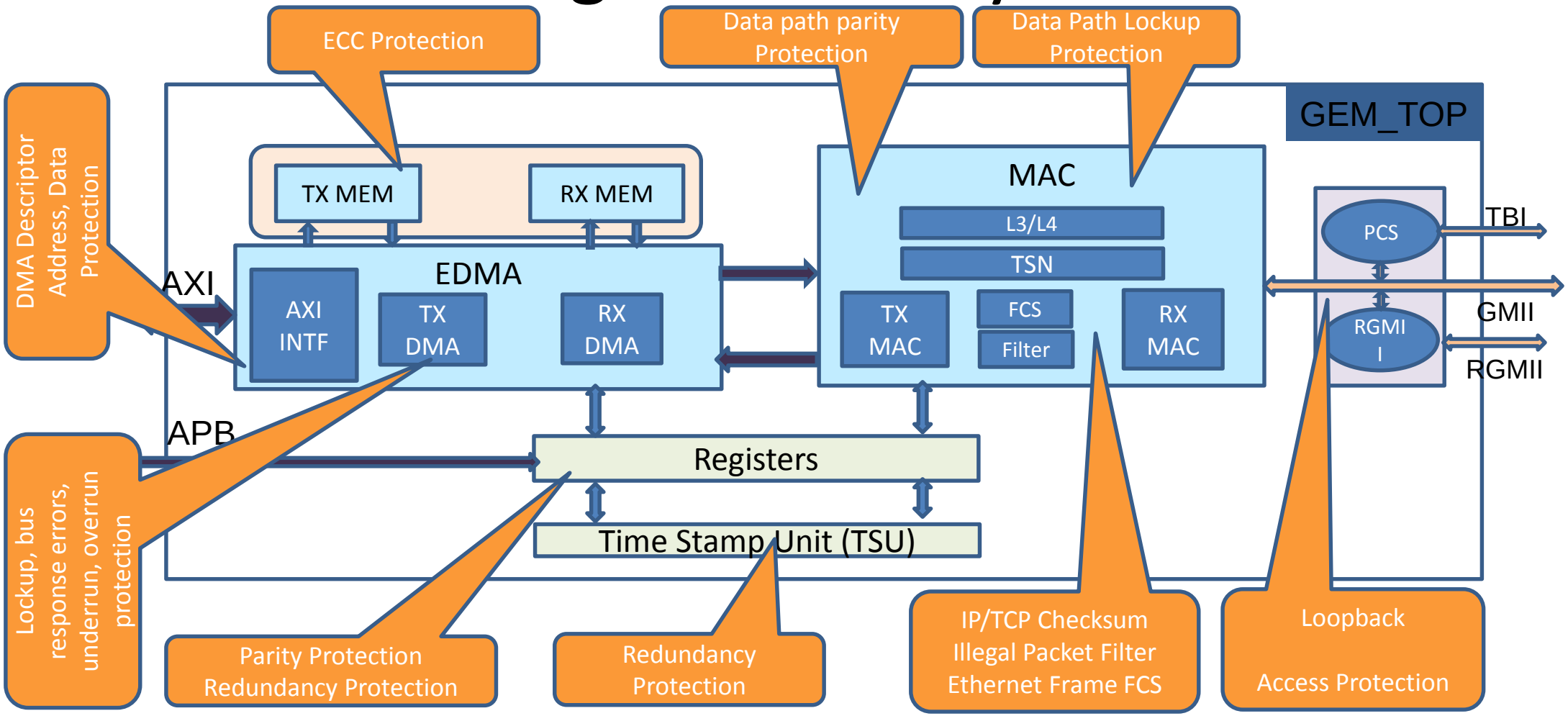
- Detailed FMEDA analysis
- Accurate Failure Mode Distribution
- Estimated SPFM/LFM based on DC targets
- SPFM/LFM based on S/DC measures and expert judgment
- Safety Architecture finalization
- Safety Requirements

GEM Block Diagram





GEM Block Diagram - Safety mechanism view



FMEDA Template – Instance Mapping Formats

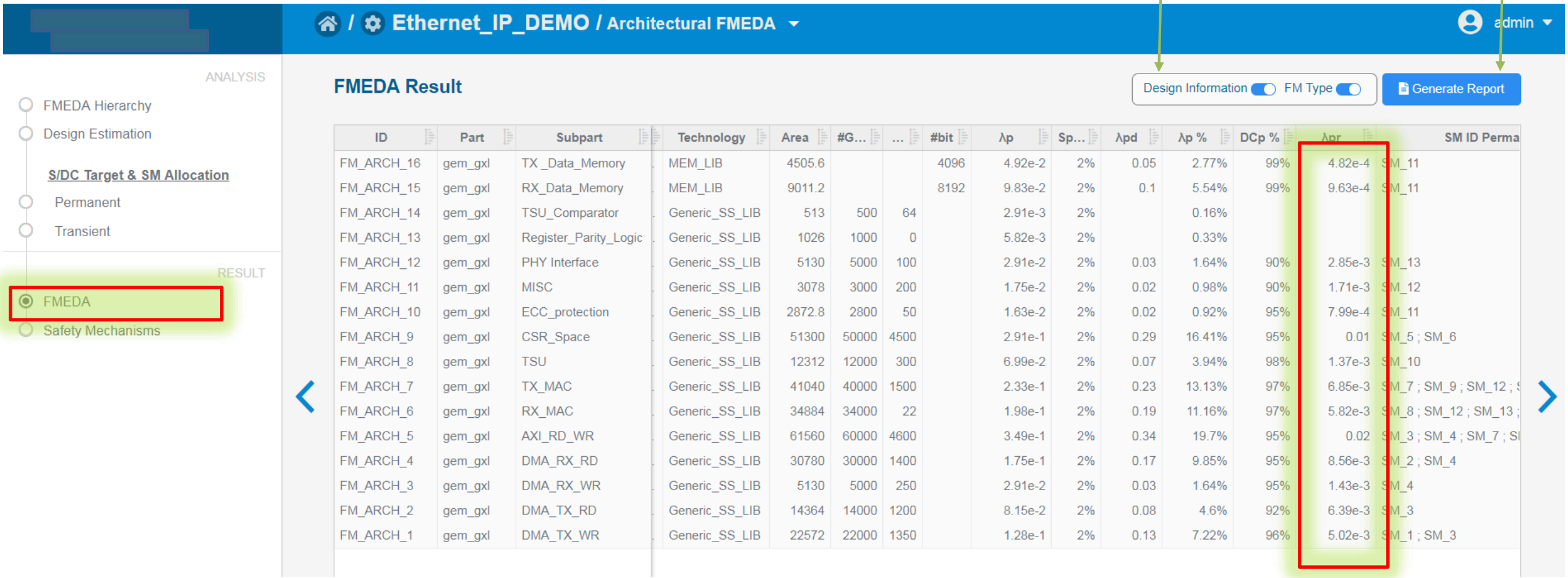
DESIGN MAPPING INFORMATION						
Part (Block)	**Part Instances mapping**	**Subpart (Sub-block)**	**Description**	**Subpart Instances mapping**	**Failure Mode**	**Failure Mode Instances mapping**
i_gem_top	gem_gxl.i_gem_ss.i_gem_top	gen_tsu_i_gem_tsu	Time Stamp Unit	gem_gxl.i_gem_ss.i_gem_top.gen_tsu_i_gem_tsu	The timer value may not be captured or captured incorrectly	gem_gxl.i_gem_ss.i_gem_top.gen_tsu_i_gem_tsu…
					The TSU seconds interrupt is incorrect	gem_gxl.i_gem_ss.i_gem_top.gen_tsu_i_gem_tsu
					TSU compare interrupt is incorrect	gem_gxl.i_gem_ss.i_gem_top.gen_tsu_i_gem_tsu
					TX/RX timestamp is corrupted, output TSU timer value to local system will be invalid, Timer value read back in registers is also invalid. May also cause TX lockup for time based scheduling.	gem_gxl.i_gem_ss.i_gem_top.gen_tsu_i_gem_tsu
		i_gem_reg_top.i_gem_registers	Etherent IP Registers	gem_gxl.i_gem_ss.i_gem_top.i_gem_reg_top.i_gem_registers…	Fault in Parity Generators of	gem_gxl.i_gem_ss.i_gem_top.i_gem_reg_top.i_gem_registers.gen_axi_i_gem_parity_gen_dma_config_burst_len; gem_gxl.i_gem_ss.i_gem_top.i_gem_reg_top.i_gem_registers.i_gem_parity_gen_dma_config_byte_1; gem_gxl.i_gem_ss.i_gem_top.i_gem_reg_top.i_gem_registers.i_gem_parity_gen_default; gem_gxl.i_gem_ss.i_gem_top.i_gem_reg_top.i_gem_registers.i_gem_parity_gen_int_mask_disable
						em_ss.i_gem_top.i_gem_reg_top.i_gem_registers…;!gem_gxl.i_gem_ss.i_gem_top.i_gem_reg_top.i_gem_registers.gen_axi_i_gem_parity_gen_dma_config_burst_len; i_gem_ss.i_gem_top.i_gem_reg_top.i_gem_registers.i_gem_parity_gen_dma_config_byte_1; m_gxl.i_gem_ss.i_gem_top.i_gem_reg_top.i_gem_registers.i_gem_parity_gen_default; l.i_gem_ss.i_gem_top.i_gem_reg_top.i_gem_registers.i_gem_parity_gen_int_mask_disable
					Fault in dynamic control outputs from the registers	gem_gxl.i_gem_ss.i_gem_top.i_gem_reg_top.i_gem_registers;

a.b.c.d… -> Include all the submodules instances inside the module ‘d’
a.b.c.d -> Include only Submodule ‘d’
!a.b.c.d.e… -> Exclude all the submodules instances inside the module ‘e’
!a.b.c.d.e -> Exclude only Submodule ‘e’

Architectural Flow

Controls visibility

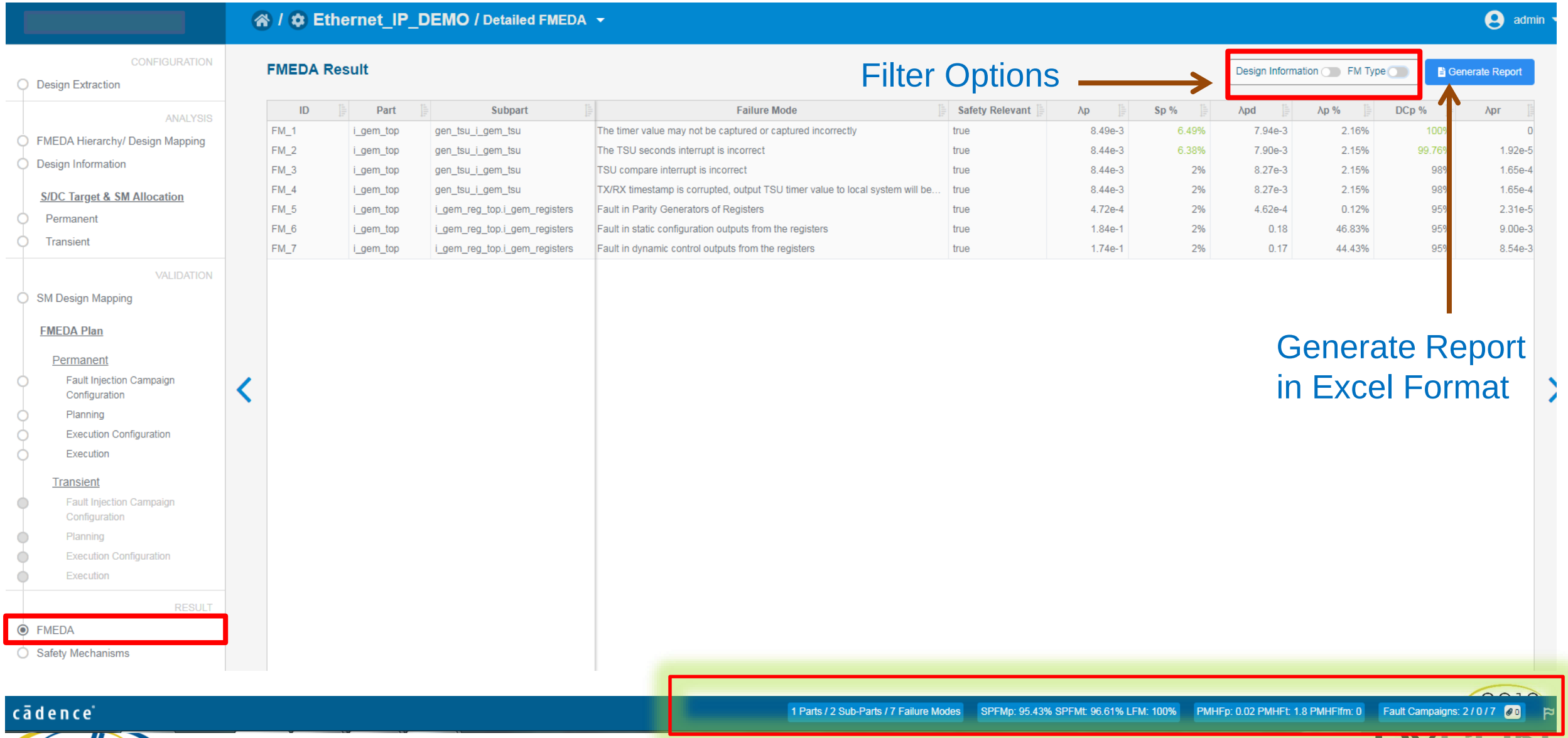
Report



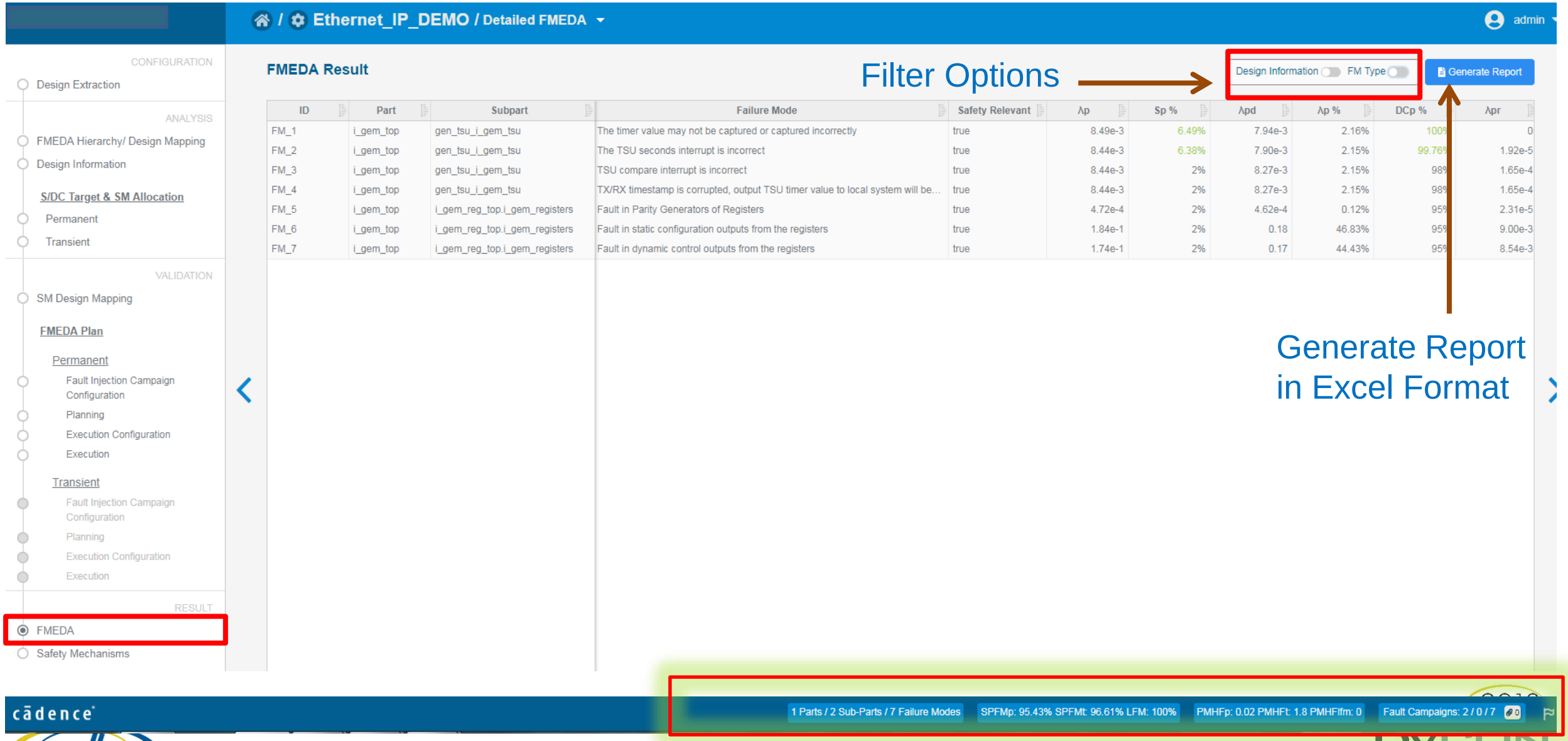
ID	Part	Subpart	Technology	Area	#G...	...	#bit	λp	Sp...	λpd	λp %	DCp %	λpr	SM ID Perma
FM_ARCH_16	gem_gxl	TX _Data_Memory	MEM_LIB	4505.6			4096	4.92e-2	2%	0.05	2.77%	99%	4.82e-4	SM_11
FM_ARCH_15	gem_gxl	RX_Data_Memory	MEM_LIB	9011.2			8192	9.83e-2	2%	0.1	5.54%	99%	9.63e-4	SM_11
FM_ARCH_14	gem_gxl	TSU_Comparator	Generic_SS_LIB	513	500	64		2.91e-3	2%		0.16%			
FM_ARCH_13	gem_gxl	Register_Parity_Logic	Generic_SS_LIB	1026	1000	0		5.82e-3	2%		0.33%			
FM_ARCH_12	gem_gxl	PHY Interface	Generic_SS_LIB	5130	5000	100		2.91e-2	2%	0.03	1.64%	90%	2.85e-3	SM_13
FM_ARCH_11	gem_gxl	MISC	Generic_SS_LIB	3078	3000	200		1.75e-2	2%	0.02	0.98%	90%	1.71e-3	SM_12
FM_ARCH_10	gem_gxl	ECC_protection	Generic_SS_LIB	2872.8	2800	50		1.63e-2	2%	0.02	0.92%	95%	7.99e-4	SM_11
FM_ARCH_9	gem_gxl	CSR_Space	Generic_SS_LIB	51300	50000	4500		2.91e-1	2%	0.29	16.41%	95%	0.01	SM_5 ; SM_6
FM_ARCH_8	gem_gxl	TSU	Generic_SS_LIB	12312	12000	300		6.99e-2	2%	0.07	3.94%	98%	1.37e-3	SM_10
FM_ARCH_7	gem_gxl	TX_MAC	Generic_SS_LIB	41040	40000	1500		2.33e-1	2%	0.23	13.13%	97%	6.85e-3	SM_7 ; SM_9 ; SM_12 ; S
FM_ARCH_6	gem_gxl	RX_MAC	Generic_SS_LIB	34884	34000	22		1.98e-1	2%	0.19	11.16%	97%	5.82e-3	SM_8 ; SM_12 ; SM_13 ;
FM_ARCH_5	gem_gxl	AXI_RD_WR	Generic_SS_LIB	61560	60000	4600		3.49e-1	2%	0.34	19.7%	95%	0.02	SM_3 ; SM_4 ; SM_7 ; SI
FM_ARCH_4	gem_gxl	DMA_RX_RD	Generic_SS_LIB	30780	30000	1400		1.75e-1	2%	0.17	9.85%	95%	8.56e-3	SM_2 ; SM_4
FM_ARCH_3	gem_gxl	DMA_RX_WR	Generic_SS_LIB	5130	5000	250		2.91e-2	2%	0.03	1.64%	95%	1.43e-3	SM_4
FM_ARCH_2	gem_gxl	DMA_TX_RD	Generic_SS_LIB	14364	14000	1200		8.15e-2	2%	0.08	4.6%	92%	6.39e-3	SM_3
FM_ARCH_1	gem_gxl	DMA_TX_WR	Generic_SS_LIB	22572	22000	1350		1.28e-1	2%	0.13	7.22%	96%	5.02e-3	SM_1 ; SM_3

1 Parts / 16 Sub-Parts / 16 Failure Modes | SPFMp: 95.85% SPFMt: 95.14% LFM: 99.95% | PMHFp: 0.07 PMHFt: 6.9 PMHFlfm: 8.56e-4

Detailed Flow



Detailed Flow



Summary

- Integrated functional + safety verification flow and engines
  - Reduce effort of developing & maintaining different environments
  - Highest performance native fault simulation
- vPlan-based requirements traceability, with FMEDA plan based metrics analysis
  - Use FMEDA user strategy to reduce amount of required fault simulation
  - Integrate design data for accurate analysis
- Automate the functional safety verification tool flow
  - Reduce the human effort whenever possible, automate all possible steps
  - Minimize the Fault Injection Campaign Set-up time
  - Reuse of Functional Verification Environment
- Execution core
  - Optimized flow to improve the TAT and optimized usage of each involved feature
  - Use coverage data to reduce the number of test to be executed
  - Utilization of formal techniques to reduce the fault space
  - Intelligent algorithms to minimize the number of fault to be injected and executed
- Cadence provides the most comprehensive solution for Functional Safety

Thanks