2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Making Autonomous Cars Safer – One chip at a time

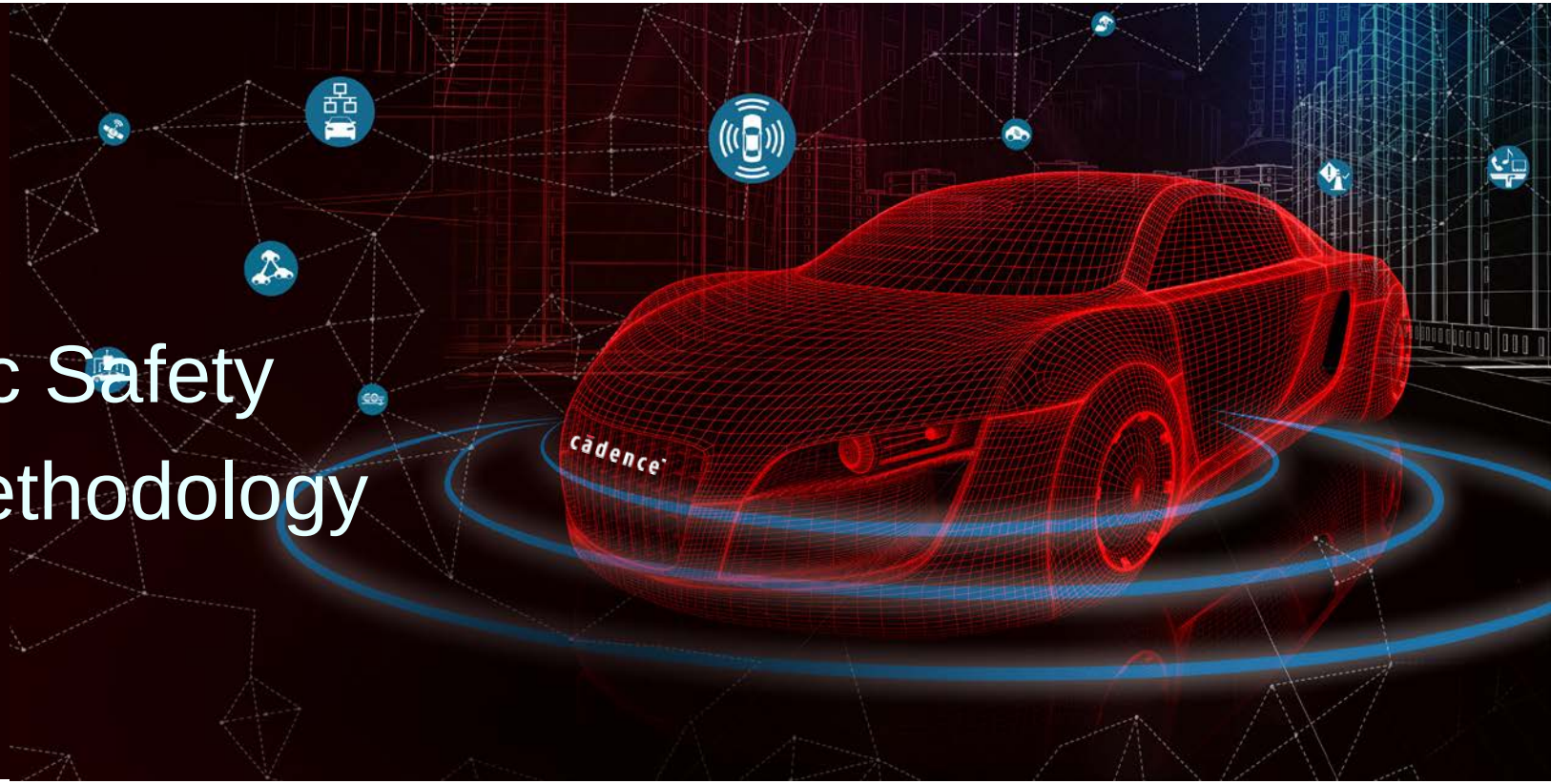
Apurva Kalia: Vice President R&D
Ann Keffer: Product Management Director

cadence®

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

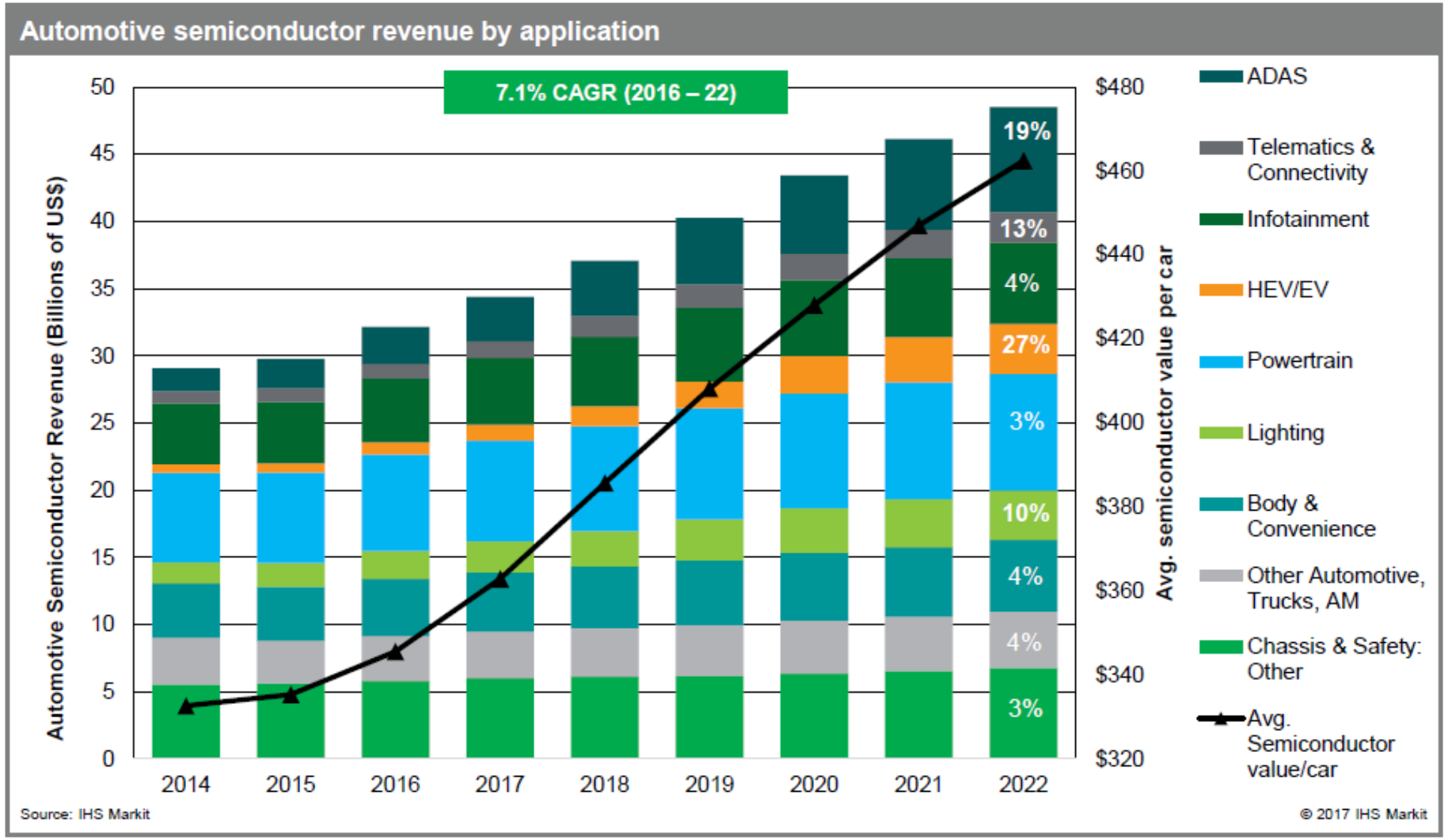
Agenda

- Automotive Market
- Complex Challenges
- ISO 26262 and Basic Safety
- Functional Safety Methodology



The Automotive Market

Automotive Semiconductor Growth



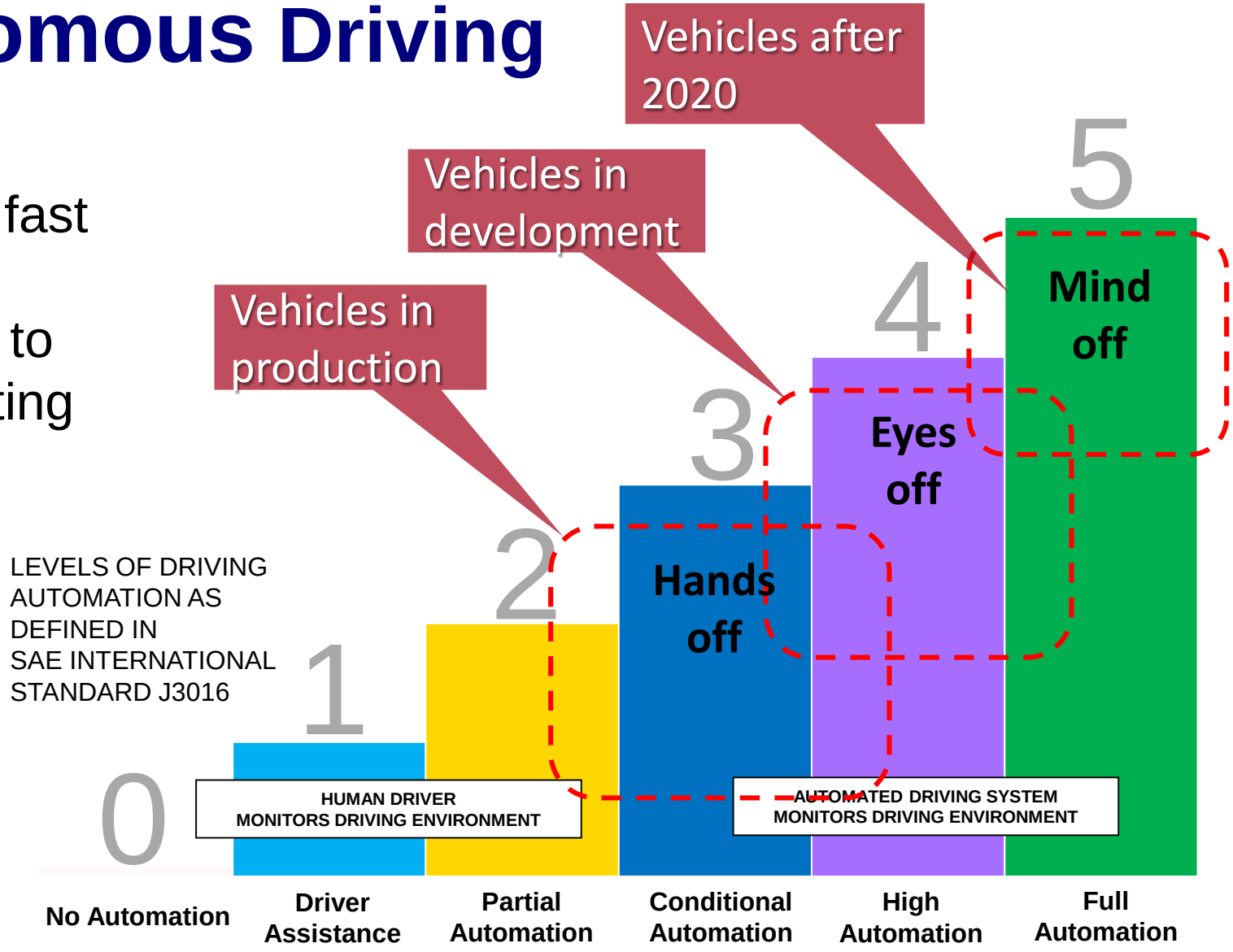
Forces Shaping the Automotive Industry

"Automotive Revolution – Perspective towards 2030" – a **2016 McKinsey Report identified 4 areas that deemed particularly important in shaping the auto industry thru 2030**

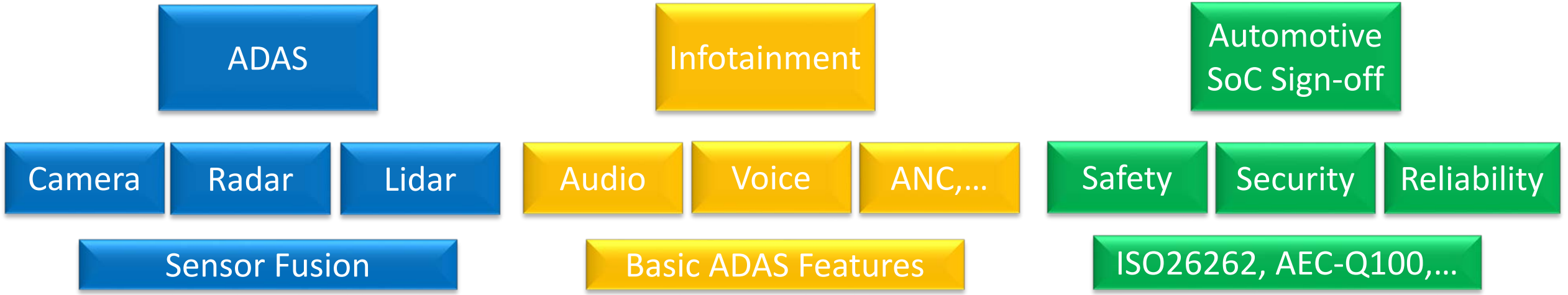
Vehicle electrification	Increased Connectivity	Growth of Autonomous Driving	Shared Mobility Services
Advances to solve • High battery costs • Proliferation of charging infrastructure	Advances to • 5G deployment • Telematics services • V2I; V2V	ADAS deployment • Cost effective Level 3 and Level 4 by 2020~2025	Proliferation of • Ride sharing services • Car sharing services

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



Automotive Opportunities and Focus Areas



High-performance computing
- Scalability
- High resolution
- Low power
- Vision + CNN
- Memory bandwidth
- Safety and Security is a must!

Highly integrated cockpit
- Scalability
- Connectivity
- In-vehicle networking
- SW app availability
- Comprehensive I/F support
- Basic ADAS features

Qualification of new SoCs
- Safety, Security and Reliability
- FMEDA not sufficient for SoCs
- Integrated FMEDA and safety verification flow
- Interfaces to RM & Tracing tools

Complex Challenges

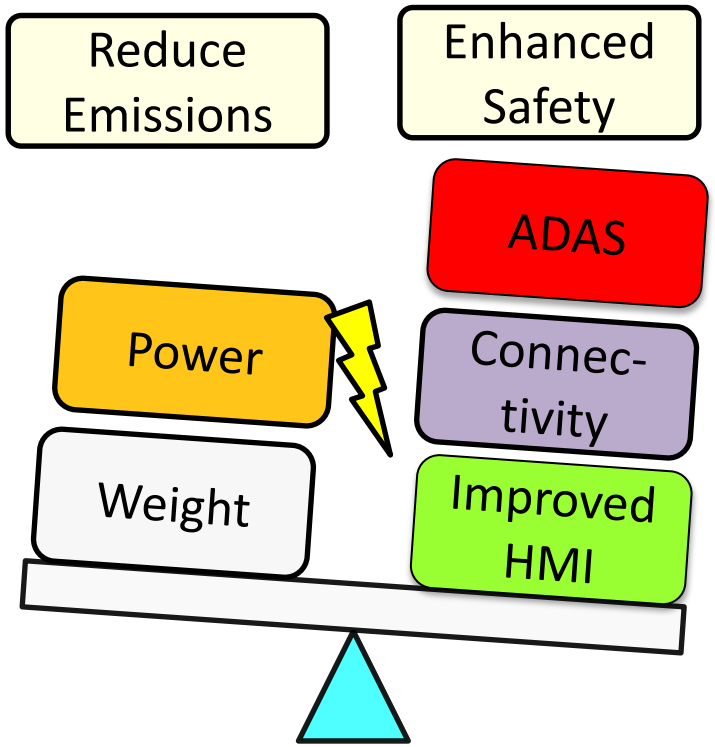
The Megatrends Dilemma

Efficient Electric Vehicles

Source: BMW

Government Regulations

EURO NCAP Program



Safe Autonomous Cars

Source: Volvo

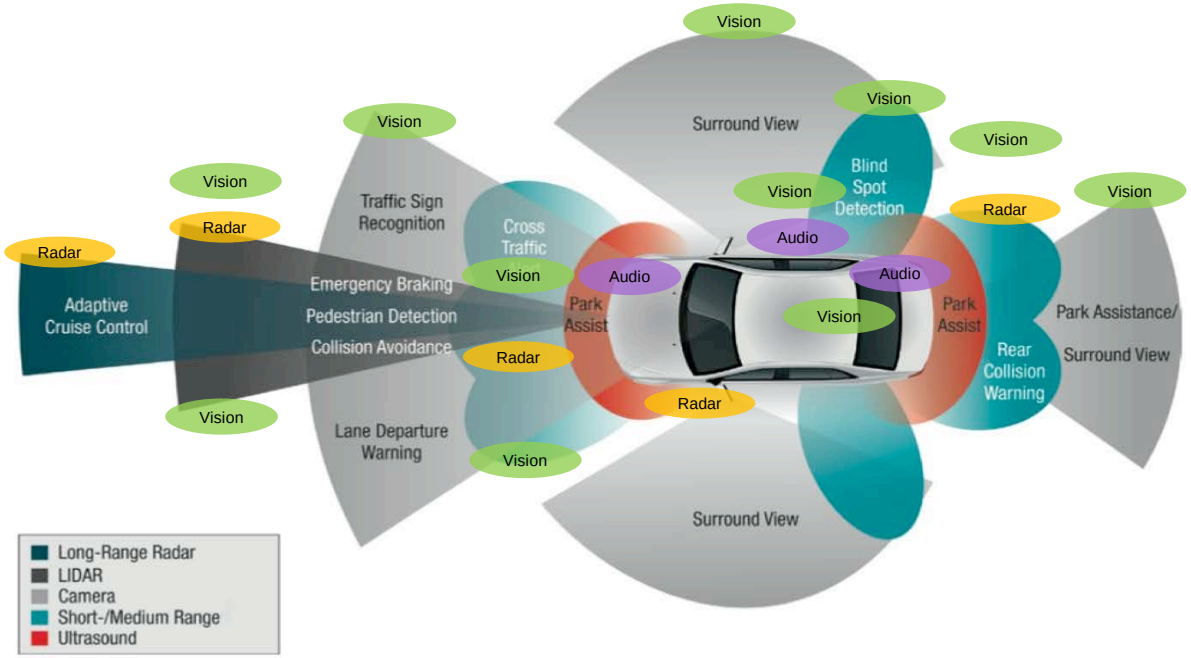
Need low-power, small footprint, high-performance SoCs

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Making a Car Autonomous

Audio

Rear Object Detection
Parking Assist/Auto Park
Voice Recognition
Cabin Noise Reduction
Emergency Recognition
Spatial Audio for Warnings



Fusion

Radar, LIDAR, Image correlation
System Functional Safety
System Data Control

Radar

Front Collision Avoidance Braking
Adaptive Cruise Control
360 degree Hazard Awareness
Rear Collision Detection

Active Vision (LiDAR)

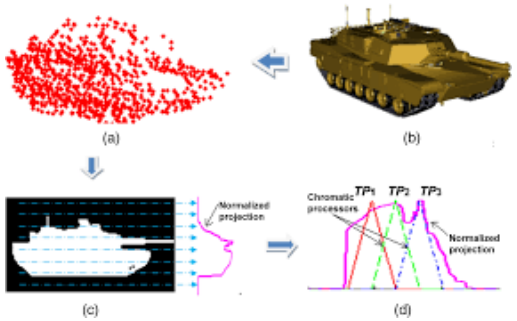
Adaptive Cruise Control
Collision Avoidance
Blind Spot Detection

Passive Vision

Rear View Camera
Vision Enhancement
Auto Dimming Headlights
Blind Spot Detection
360 View
Parking Assist
Lane Detection and Following
Sign Recognition
Traffic Signal Recognition
Rain, Snow, /Fog Removal
Pedestrian Tracking /Avoidance
Eye Focus Detection
Driver Monitoring
Vehicle Detection/Avoidance

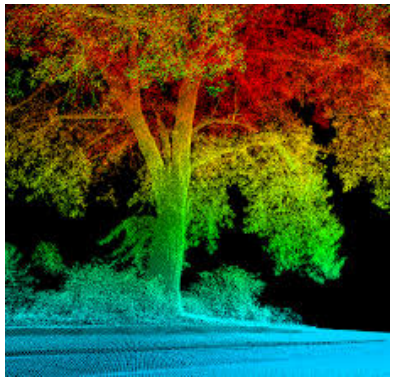
Complicated Convolutional Neural Networks

Radar Point Cloud



~10-100 KB/sec

Lidar Point Cloud



~10-70 MB/sec

Digital Camera

~20-40 MB/sec

Automated and Reliable Object Recognition using Convolutional Neural Networks

Need a high-performance, low-power hardware platform to combine and analyze point clouds and accurately identify objects

Automotive SoC Verification Challenges

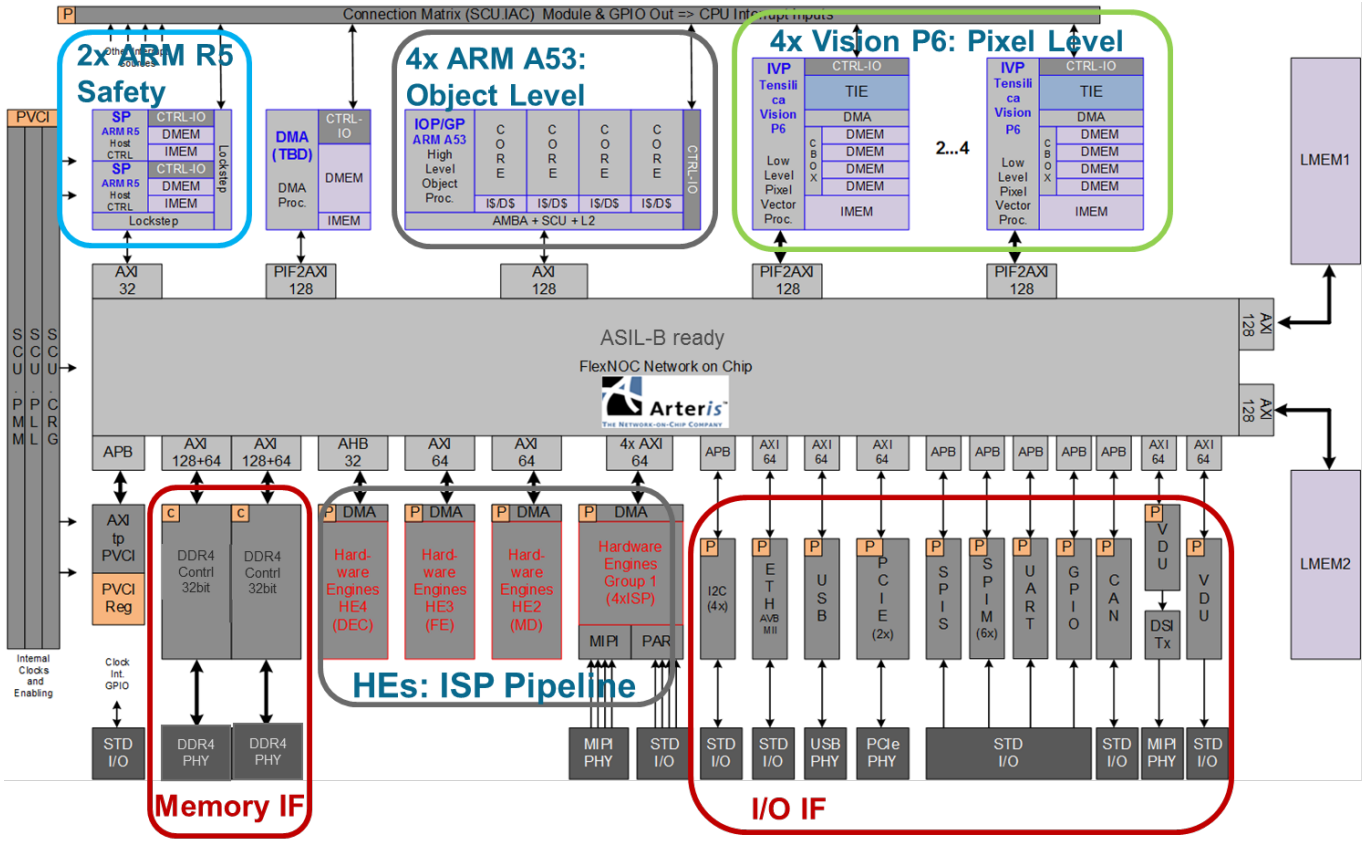
Systematic Failure Verification

- Concurrent SW Development
- Requirements Traceability
- Use Case Verification
- Performance Verification
- Security Verification
- **Automotive Protocol Verification**
- Mixed Signal Verification

Random Failure Verification

- Functional Safety Verification

ADAS SoC

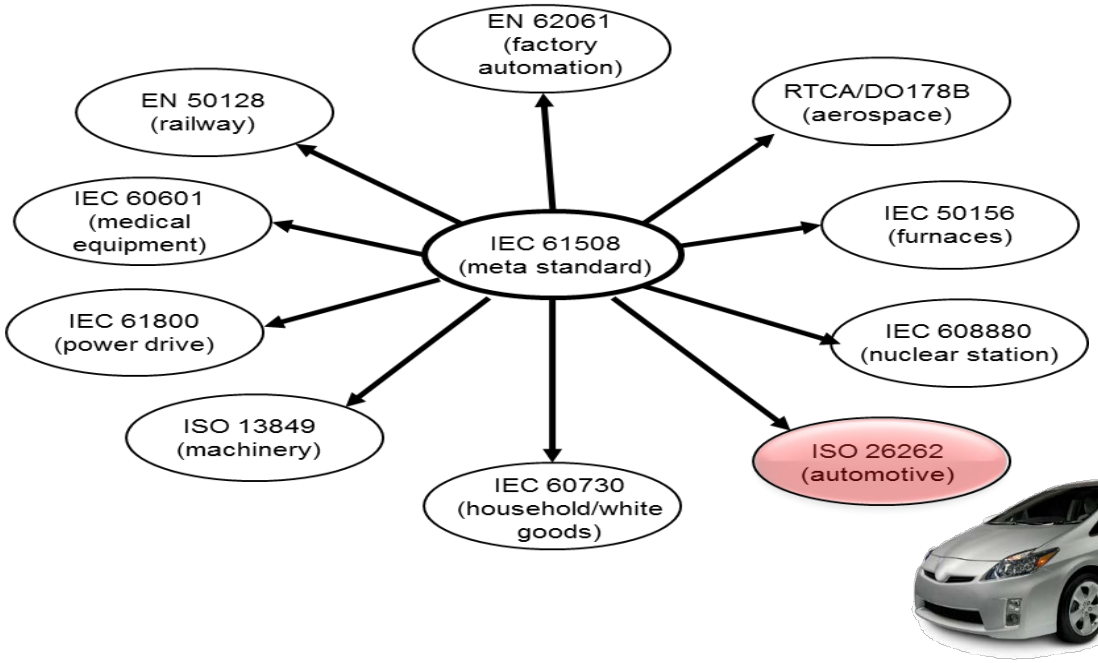


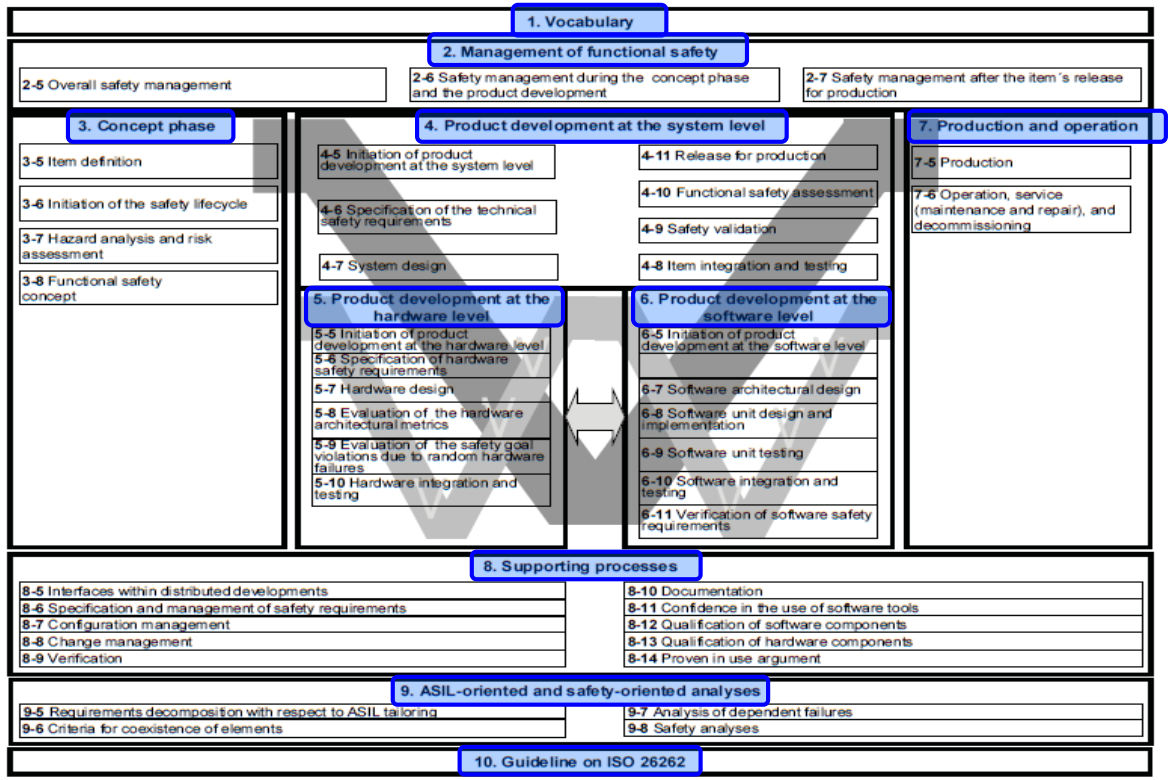
Multiple verification and validation platforms

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

ISO 26262 and Failure Mode Effects and Diagnostic Analysis

Functional Safety standards



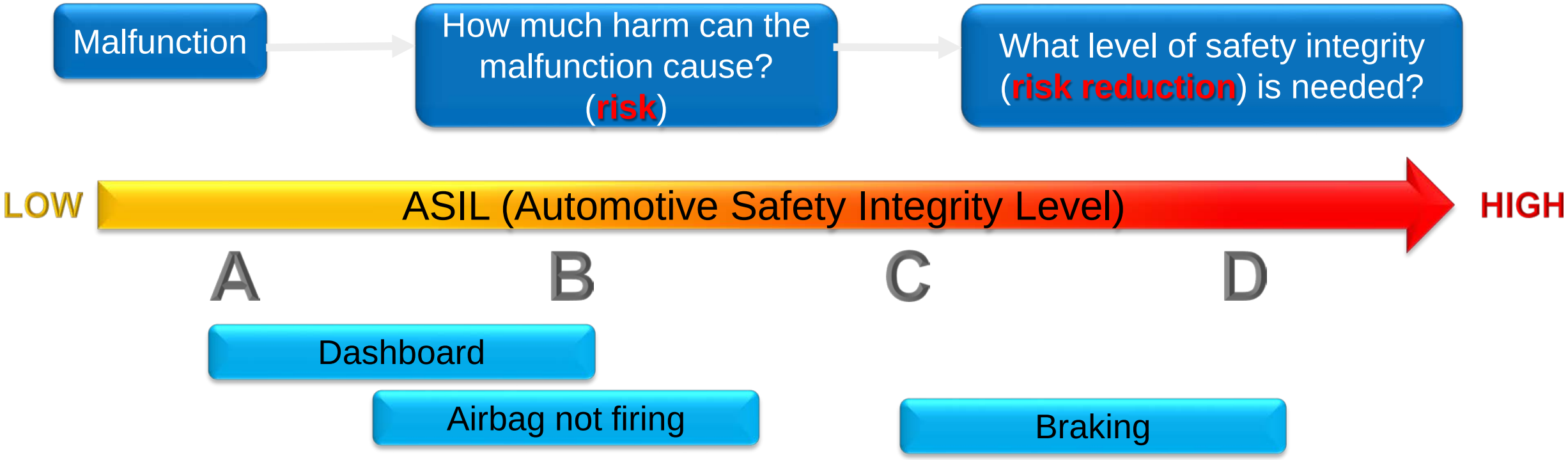


ISO 26262 defines

- 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

For illustration purposes only

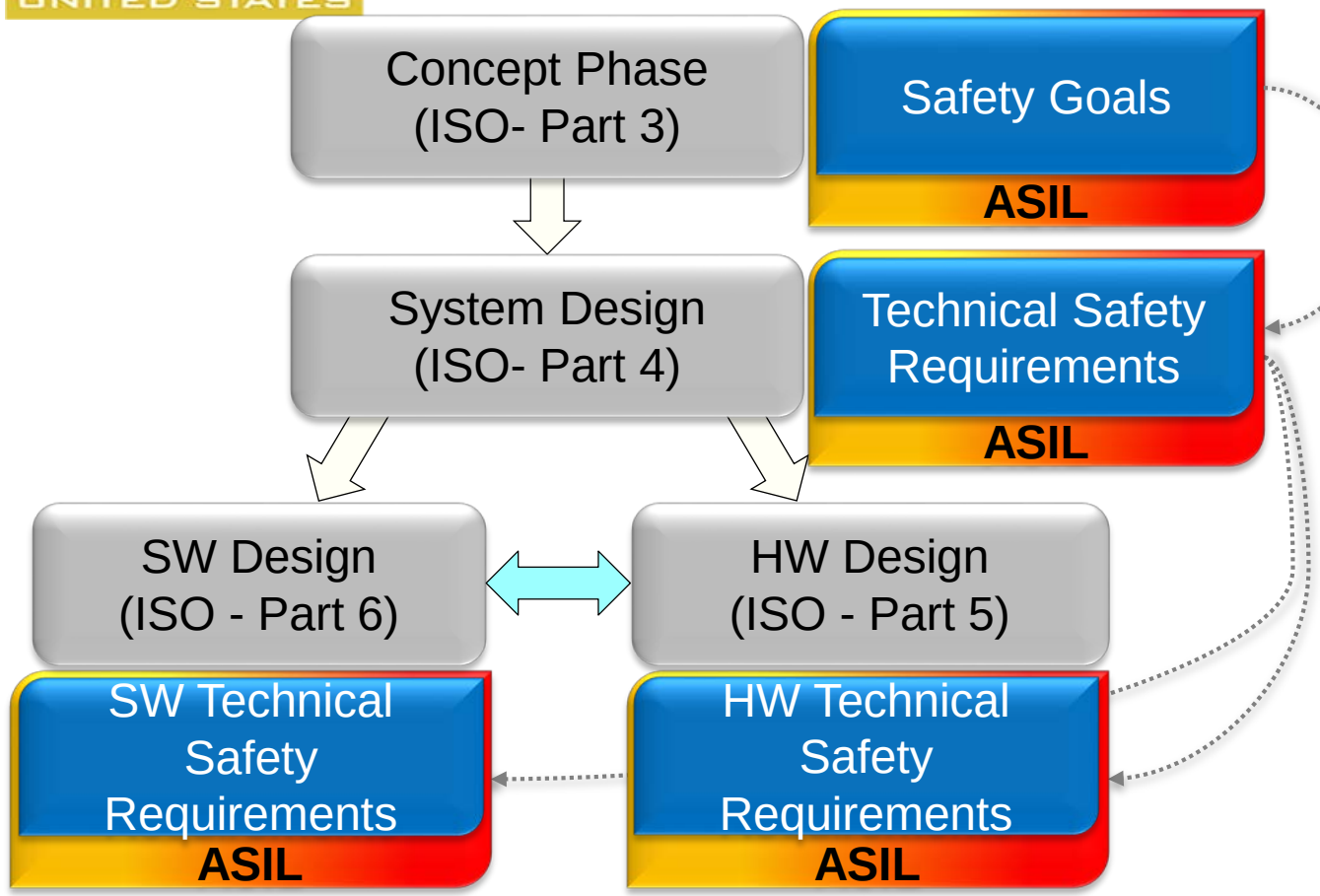
Malfunction	ABS system failure	⟺	Safety Goal: **Prevent ABS failure**
Hazard Analysis	What unintended situations (hazards) could happen? → Loss of stability on split-μ surface		
Risk Analysis	• **How likely** is the hazard to happen? (Exposure) → oil spill, gravel, water potholes, …. • **How harmful** 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)**		

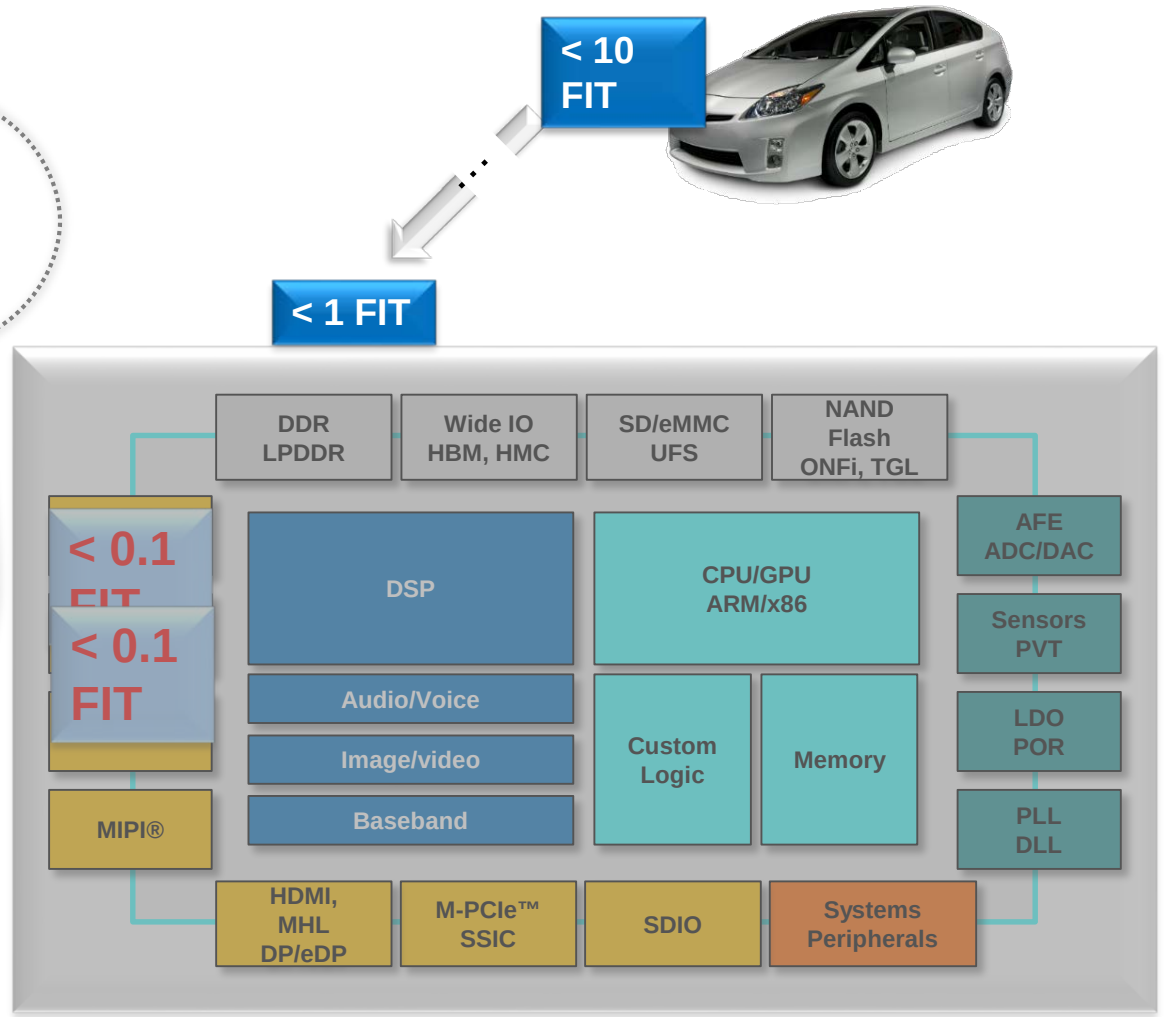
LOW — ASIL (Automotive Safety Integrity Level) — HIGH

A B C D

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

ISO 26262—Design and Safety Flow





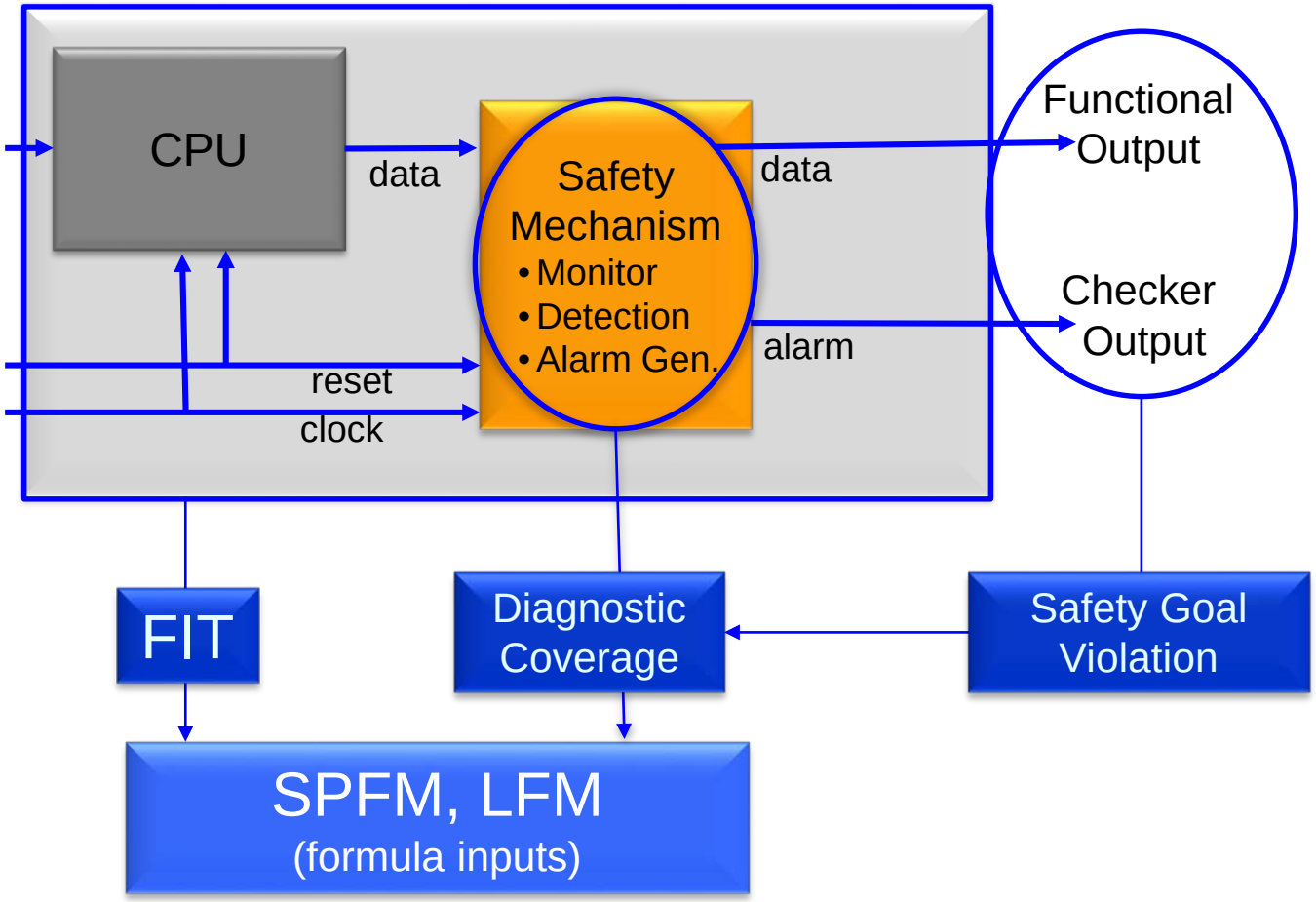
FIT gets distributed from the item to each of the elements

Break

ASIL Hardware Metrics

ASIL	Failure Rate	SPFM	LFM
A	< 1000 FIT	Not relevant	Not Relevant
B	< 100 FIT	> 90%	> 60%
C	< 100 FIT	> 97%	> 80%
D	< 10 FIT	> 99%	> 90%

- FIT Failure In Time (1 Failure / 10^9 hours)
  - PMHF Probalbilistic Metric for Random HW failures
- SPFM Single Point Fault Metric
- LFM Latent Fault Metric



Functional Safety Life Cycle Main Tasks

- Silicon provider is asked to execute five main activates to implement a Functional Safety life cycle in light of the hardware random capability.

Life Cycle	Selection of ISO26262 Process Requirements and tailoring of the development process for the specific SoC (**Safety Manger**).
Safety Concept	Assumed Safety Requirements definition for the HW component for the Development of the SoC (**Safety Architect**)
Safety Analysis	Safety Analysis: FMEA/**FMEDA**/DFA (**Safety Engineer**)
Metrics Computation	Compute Hardware Architecture Metrics (**SPFM, LFM**), **PMHF** based on the defined Safety Concept (**Safety Engineer**)
Reviews/Confirmations	Perform applicable Verification Reviews, Confirmation Reviews, Safety Audit and Assessment (**Auditor**)

- **Safety Manager** is the person in charge to define and track the Functional Safety process, define the work products, define the template documentation and execute internal reviews

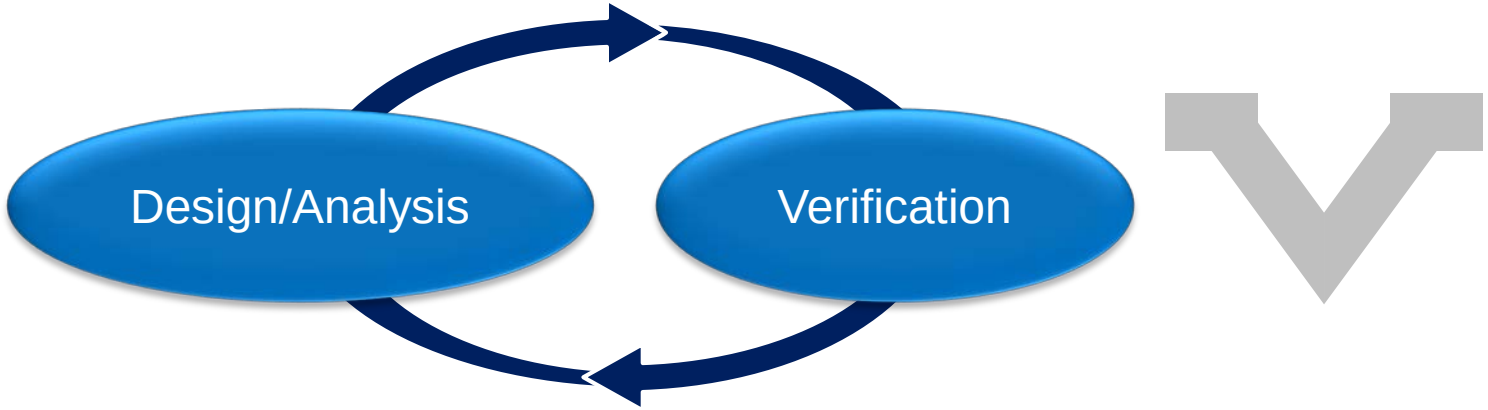
ISO26262—Functional Safety Principles

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



ISO 26262 covers random and systematic errors

Functional Safety Metrics

- Target metrics values according to ASIL (Automotive Safety Integrity Level)
- Architectural Matrices (measured in %)

  – **SPFM**: Single Point Fault Metrics

  *The single point fault metric reveals whether or not the coverage by the safety mechanisms (i.e. the **DC**), to prevent risk from single point faults in the hardware architecture, is sufficient. Single point faults are faults in an element that leads directly to the violation of a safety goal* → **SPFM high means that the set of Safety Mechanisms have high capacity to cover dangerous faults, resulting in high DC.**

  – **LFM**: Latent Fault Metrics

  The latent fault metric reveals whether or not the coverage by the safety mechanisms, to prevent risk from latent faults in the hardware architecture, is sufficient. Latent faults are multiple-point faults whose presence are not detected by a safety mechanism. Latent faults become dangerous when a second faults appears and it will be not detected due to the latent fault previously occurred → ***LFM high means that the set of Safety Mechanism have high capability to cover multiple faults (multiple = 2) scenario.***

Functional Safety Metrics

- Absolute Metrics
  - **PMHF**: Probabilistic Metric for (Random) Hardware Failures

Is the sum of the single point, residual and multipoint fault metrics. Is expressed in FITs → **PMHF low means a low probability that the SoC, including its safety mechanisms, fails without any detection. It is measured in FIT: 1FIT = probability that one failure occur in 10^9 hours. It represents the probability to violate the safety goal**

Work Products and Documentations

- List of the most relevant documents to be produced during a Functional Safety Development and to be used during an assessment

Work Products	Content	ISO26262 References
Safety Plan	Company realted process quality standards, product life cyle, product responsibilities, tools qualificaiton, project activities plan,...	ISO 26262-2:2018, 6.4.3.9
Configuration Management Plan	Process to control that work products can be uniquely identified and reproduced in a controlled manner at any time, e.g. bugs tracking and documentation	ISO 26262-8:2018
Change Management Plan	Process to changes to safety-related work products throughout the safety lifecycle, impact analysis, revisioning, ...	ISO 26262-8:2018
Safety Requirements	Design and safety mechanisms requirements compliant with technical safety report and system requirements (traceable)	ISO 26262-5:2018, Clause 6
Requirements traceability report	Show the traceability backward ans forward of the requirements.	ISO 26262-5:2018 - 7.4.2.5
HW Design Verification Plan	Descripition of the techniques and masures to avoid systematic capability: the pass and fail criteria for the verification, the metrics; the verification environment; the tools used for verification; the regression strategy.	ISO 26262-11:2018, 5.1.9 - table 30 ISO 26262-8:2018, Clause 9 ISO 26262-5:2018, 7.4.4 table3
HW Design Verification Report	Results of the verification measures (typcally metrics driven verification), derogation, ...	ISO 26262-8:2018, Clause 9
Safety Analysis Report	FMEA, FMEDA. Safety scope description, Base failure rate calculation, Fault models applied, Analysis assumptions, Analysis results , Fault injection strategy (how to execute the measures, which WL, sampling,..., expert Judgment evidences ,...	ISO 26262-9:2018, Clause 8 ISO 26262-11, 4.6
Analysis of Dependent Failure report	DFA analysis, assumption, adopted measures and results	ISO 26262-9:2018, Clause 7
Confirmations Measure Reports	Confirmation reviews of: saftey plan, safety analysis, software tool criteria evaluation report, completeness of the safety case, ...	ISO 26262-2:2018, Table 1
Safety Manual	Applied Safety Life Cycle, safety goal, safety scope, AoU description, fault models, Safety Mech. Description, Safety results summary, ...	ISO 26262-11, 4.5.4.9

FMEDA – Capture and Analyze Safety Goals

SoC Part

IP Subpart

Failure Mode

Failure Rate

Safe Fraction

Failure Mode Distribution

Diag. Cov.

HW Safety Mechanism

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

SPFMp	59,97%	SPFMt	52,76%
LFM	not calculated		

						PERMANENT					TRANSIENT								
ID	PART	SUBPART	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
2		DECODER	Incorrect Instruction Flow caused by a fault the decode logic	326	9	0,004	0,01	0,003885	0,00004	100,00%	0,015298	15%	0,013003	0,002295	100,00%	60%	CTRL FLOW, WD	60%	CTRL FLOW, WD
3		VIC	Un-intended execution/not executed interrupt request	141	4	0,002	0,26	0,001256	0,00044	100,00%	0,006793	40%	0,004076	0,002717	100,00%	60%	INT MONITOR	60%	INT MONITOR
4		ALU	Corrupt data or value caused by a fault in the register bank shadow			0,018	0,01	0,017841	0,00018	20,13%	0,089709	15%	0,059252	0,010456	19,81%	60%	PARITY	60%	PARITY
5			Incorrect Instruction Result caused by a fault in the multiplier	7465	206	0,009	0,01	0,008998	0,00009	10,15%	0,035685	15%	0,030332	0,005353	10,14%	90%	HW REDUNDANT RANGE CHK	90%	HW REDUNDANT RANGE CHK
6			Incorrect Instruction Result caused by a fault in the adder			0,002	0,01	0,002229	0,00002	2,51%	0,008508	15%	0,007232	0,001276	2,42%	90%		90%	
7			Incorrect Instruction Result caused by a fault in the divider			0,002	0,01	0,001256	0,00035	1,42%	0,006779	15%	0,005763	0,001017	1,93%	90%		90%	
8			Corrupt data or value caused by a fault in the register bank			0,030	0,01	0,029329	0,00030	33,09%	0,115579	15%	0,098242	0,017337	32,85%	95%	STL	0%	-
9			Incorrect Instruction Flow caused by a fault the pipeline controller			0,029	0,01	0,028984	0,00029	32,70%	0,115579	15%	0,098242	0,017337	32,85%	40%	CTRL FLOW, WD	40%	CTRL FLOW, WD
10		FETCH	Incorrect Instruction Flow caused by a fault the branch logic (Wrong Branch Prediction)	1606	44	0,001	0,01	0,001025	0,00001	5,35%	0,003422	15%	0,002908	0,015639	0,04574	25%	STL, WD	15%	WD
11			Incorrect Instruction Flow caused by a fault the fetch logic			0,018	0,01	0,018115	0,00018	94,65%	0,071387	15%	0,060679	0,015639	0,95426	19%	STL	0%	-
12																			
13																			
14																			
15																			
16																			
17	BUS																		
				10374	286			0,120364	0,00452				0,403188	0,104706					

LINK

A SM can cover more the one FMs

One FM can be covered by multiple SMs

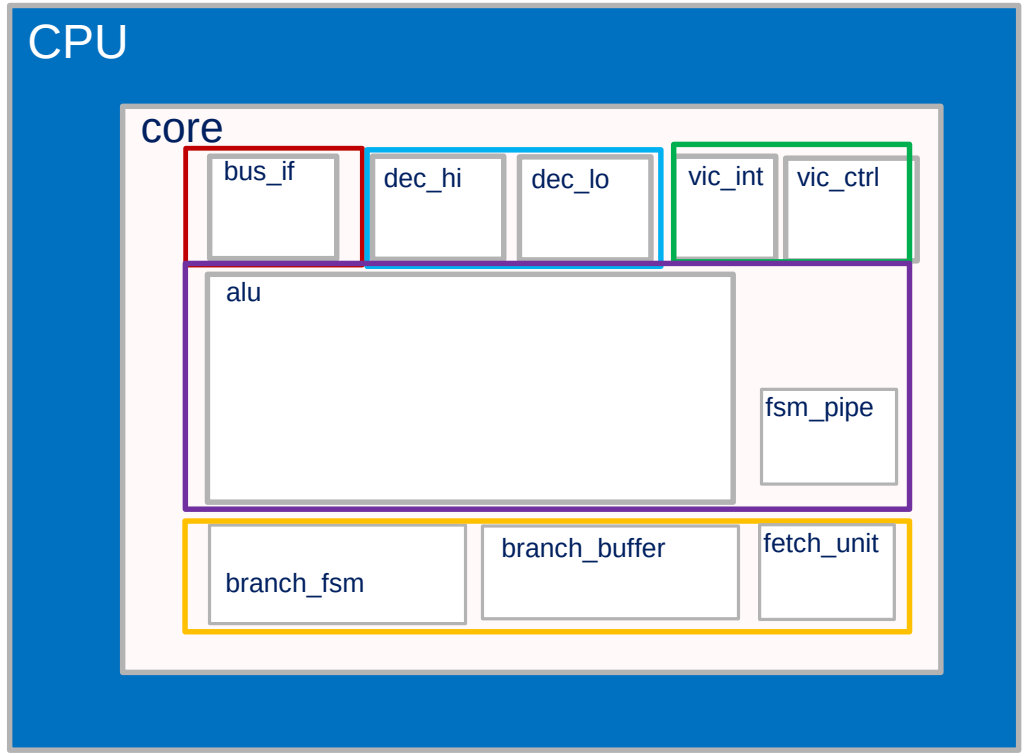
FMEDA Analysis

- User defines the FMEDA Hierarchy starting from design requirements
- Part and Subpart are not one by one with the physical implementation

FMEDA Hierarchy

ID	PART	SUBPART	Failure Mode
1	CPU	BUS_ITF	
2		DECODER	
3		VIC	
4		ALU	
5			
6			
7			
8			
9			
10		FETCH	
11			

Design Hierarchy: from requirements



FMEDA Analysis

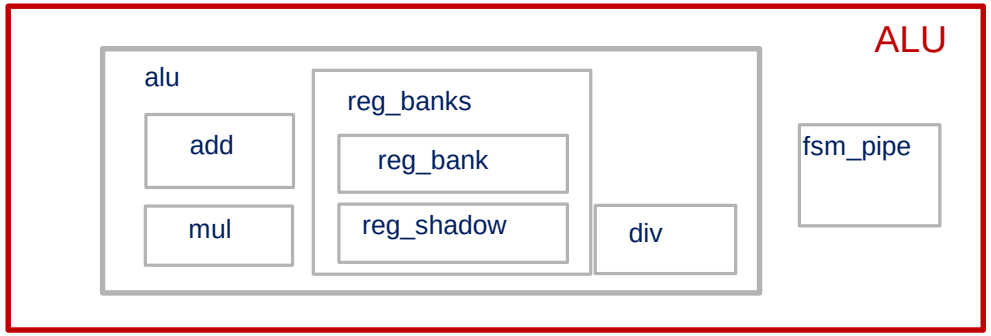
- User provides textual description of the FMs (for every subpart) figured-out during the failure functional analysis

ID	PART	SUBPART	Failure Mode
1	CPU	BUS_ITF	Wrong Data Transaction caused by a fault in the AHB interface
2		DECODER	Incorrect Instruction Flow caused by a fault the decode logic
3		VIC	Un-intended execution/not executed interrupt request
4		ALU	Corrupt data or value caused by a fault in the register bank shadow
5			Incorrect Instruction Result caused by a fault in the multiplier
6			Incorrect Instruction Result caused by a fault in the adder
7			Incorrect Instruction Result caused by a fault in the divider
8			Corrupt data or value caused by a fault in the register bank
9			Incorrect Instruction Flow caused by a fault the pipeline controller
10		FETCH	Incorrect Instruction Flow caused by a fault the branch logic (Wrong Branch Prediction)
11			Incorrect Instruction Flow caused by a fault the fetch logic

FM definition: comes from a cause-effect user analysis starting from specs or RTL

FM4: “Corrupt data or value caused by a fault in the register bank shadow”

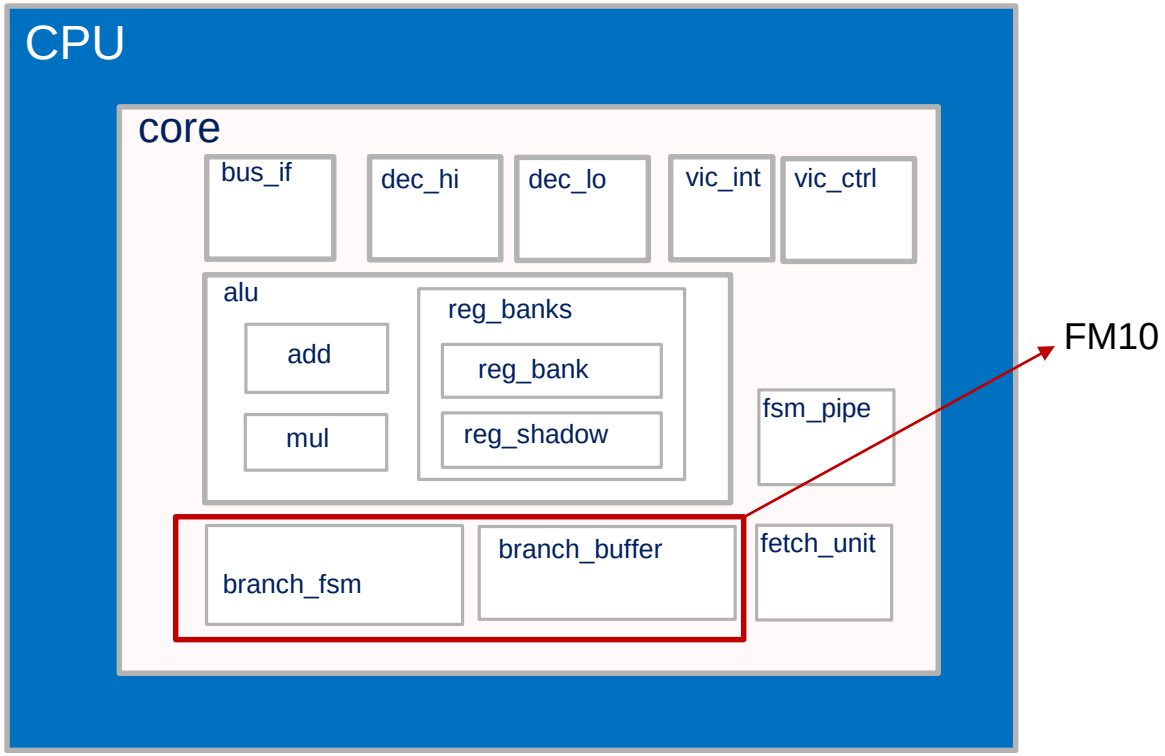
e.g. The ALU function has six different way to fail



FMEDA Validation

- **FM mapping** is performed by the user associating FMs (defined into the FMEDA) to Design Instances (hierarchical full path name)

Design Hierarchy: instances full path names



MODULES	DES INFO	
Not Stuctural	#gates	#flops
bus_if	810	21
dec	295	12
vic_int	70	2
vic_ctrl	50	6
reg_shadow	1650	20
add	1100	40
mul	1200	60
div	1500	80
reg_bank	2240	60
fsm_pipe	2320	73
branch_fsm	98	4
branch_buffer	1420	35
tot	***12753***	***413***

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

FMEDA Validation

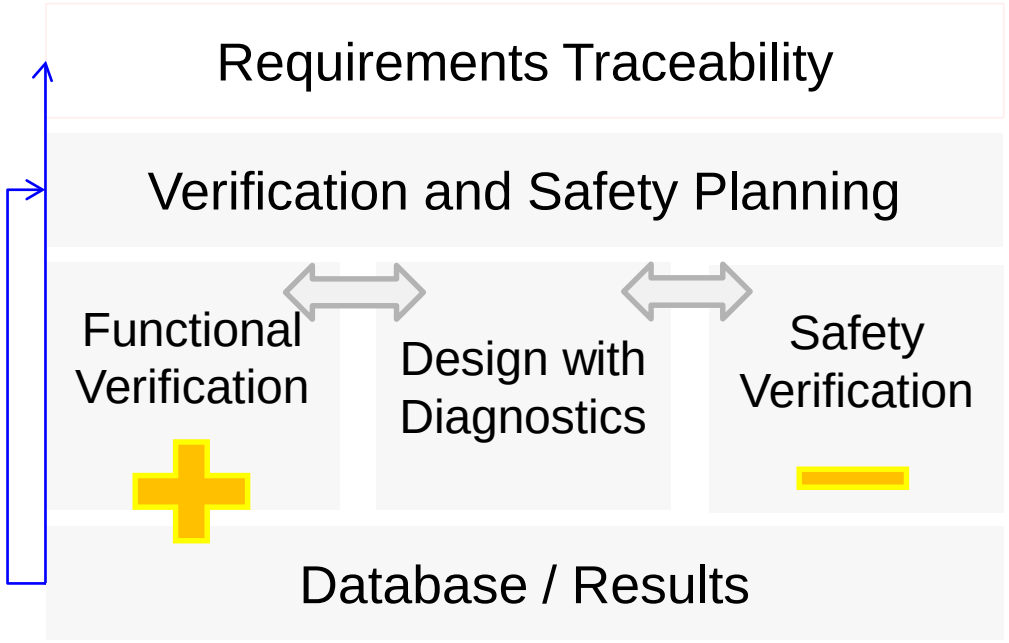
- Before executing the fault injection campaigns an FMEDA Plan shall be finalized
- The FMEDA validation is executed on a FM basis, meaning that a specific fault campaign is executed for every FM.
- The user supplies, still on a FM basis, observation points and detection points according to the verification requirements supplied by the safety engineer

Use Cases

- When the SoC complexity grows a **modular approach** is required to initiate an FMEDA and execute its validation
- An FMEDA **team based approach** should be also supported to allow splitting the job among different teams, enabling an IP-based methodology
- IP could be provided from **3rd party IP provider** and will come with it's own FMEDA

Functional Safety Methodology

Build a Holistic Solution



- Integrate Safety Mechanisms to reduce the FIT
- Positive testing (functional verification)
  - Verify proper functionality prior to safety verification
- Negative testing (assess diagnostic capability):
  - Targeted tests to confirm failure mode assumptions
  - Statistical tests to ensure design function integrity
  - Transient faults testing to provide evidence safety mechanisms integrity

Build Chips for Safe Autonomous Automobiles

Current Need

- A dedicated functional safety verification methodology and process for these safety-critical IPs and SoCs
- Safety analysis in semiconductor such as fault injection, fault metrics, base failure rate estimation, interfaces within distributed developments, handling of Hardware Intellectual Property (IP)

Methodology

- Holistic methodology which combines analytical methodologies such as FMEDA with dynamic fault simulation and formal analysis based methodologies to significantly reduce the safety verification effort and achieve faster product certification

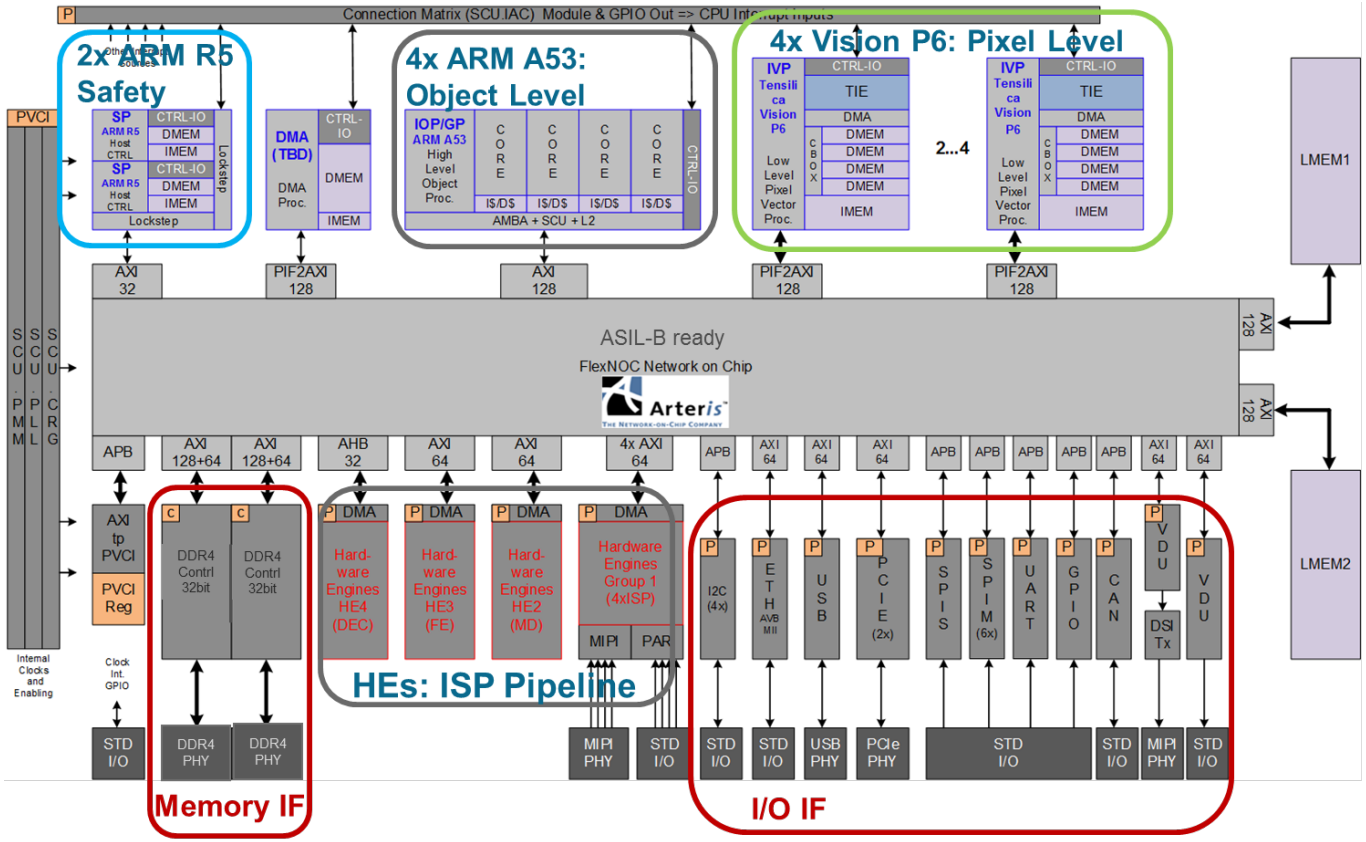
Metrics

- ISO26262 recommends single point fault metric (SPFM) and Latent Fault Metric (LFM) for the component (IP and SoCs)
- Will be measured for each of the identified Safety Goals associated with the safety critical modules within the IPs and/or SoCs.

Safety Verification Challenges and More

- Failure Mode Definition
- Safety Mechanism Design
- Fault Campaign Planning
- Safety Requirement Traceability
- Fault Set (+Optimization) Execution
- Verification Environment Re-use
- Multiple Engines Support
- Link to FMEDA (Metrics Calculation)
- Tool Confidence Level (TCL)

ADAS SoC Example



Break

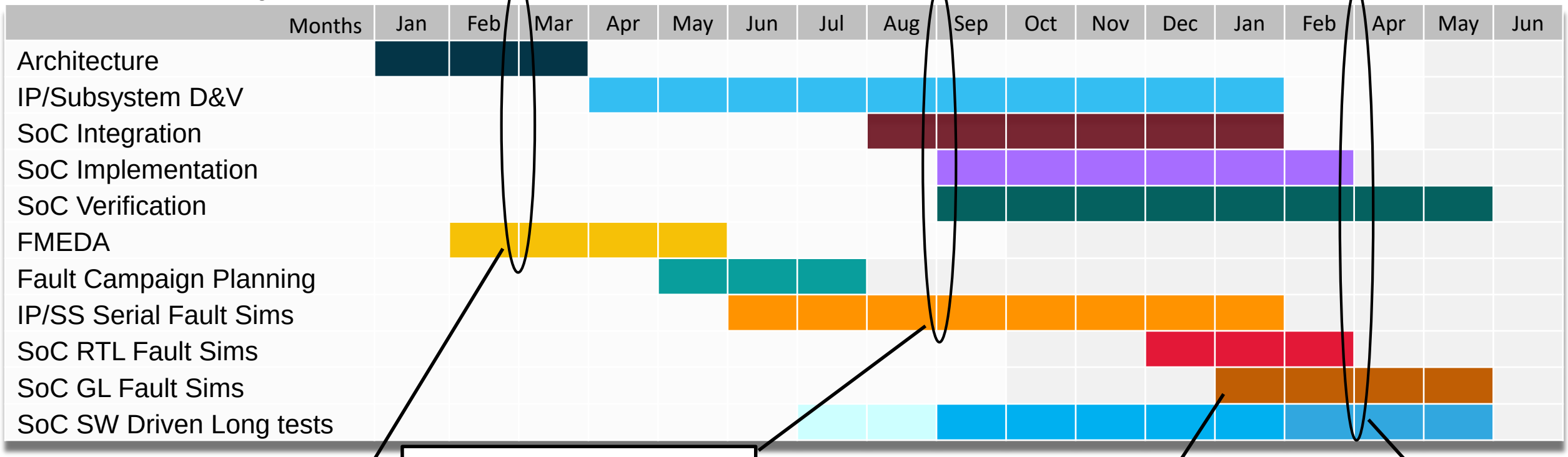
2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Safety Verification Methodology

FMEDA FS architecture analysis key to reducing overall FS efforts

Start serial fault injection early on RTL. Reuse same TB and coverage

Common fault coverage DB to integrate results across engines



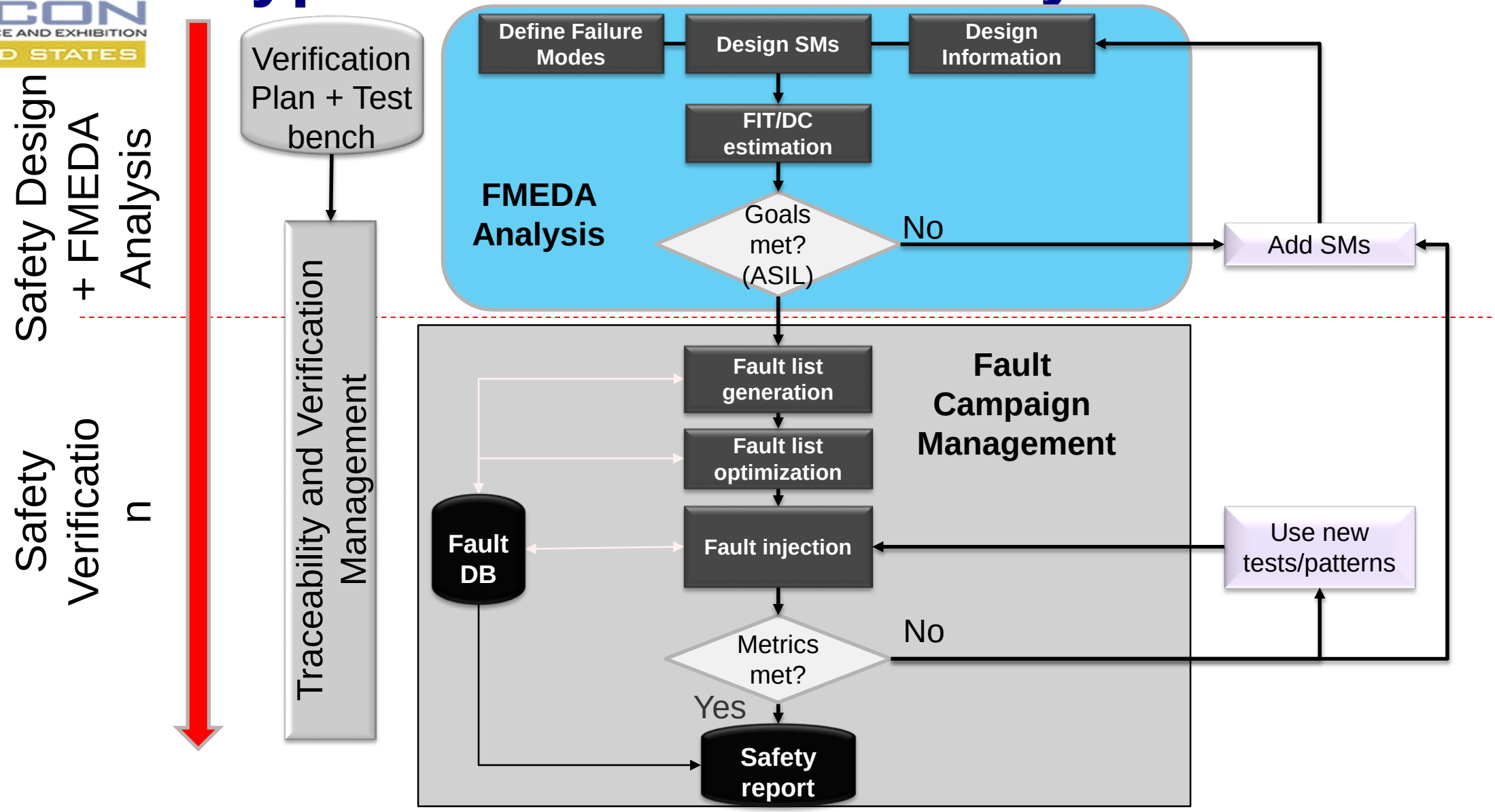
Start early with FMEDA, Fault Campaign planning, and flow set up

RTL (serial) fault campaigns to clean flow select tests, & debug Lesser faults, RTL sim is faster

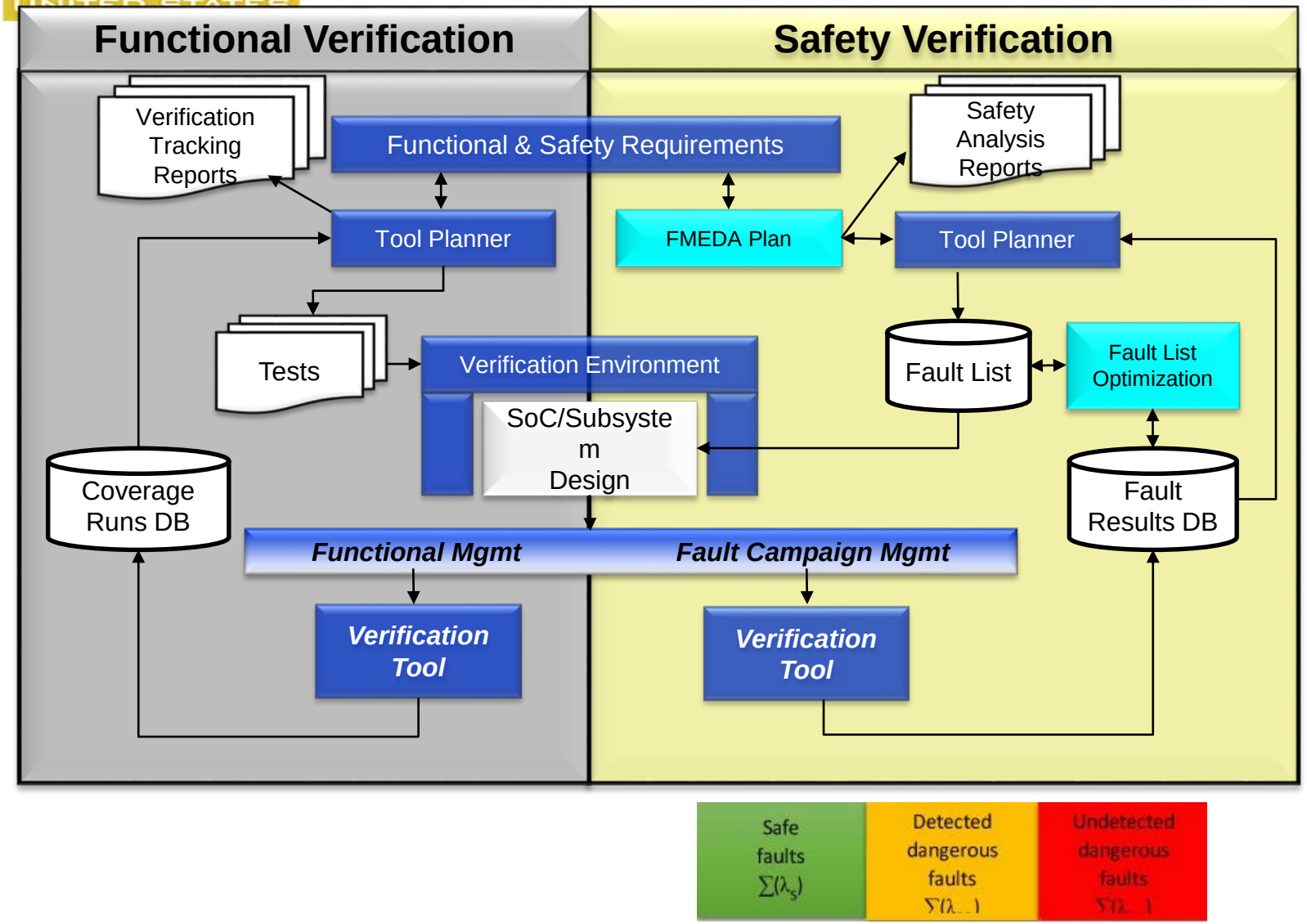
Complete fault campaigns at gate level for signoff (concurrent)

Hardware accelerated fault simulation

Typical Functional Safety Workflow



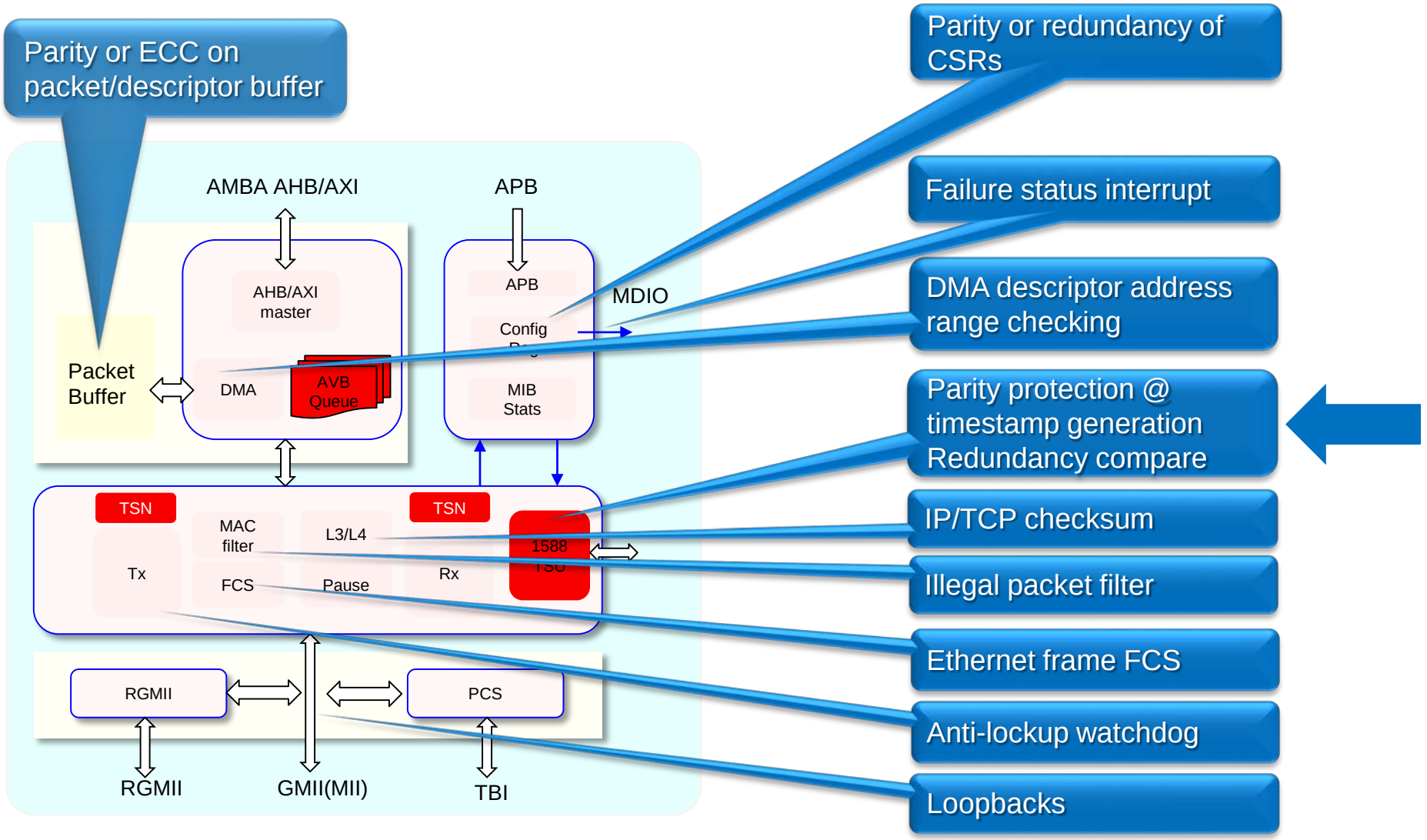
Safety Verification Solution



- Unified functional + safety verification flow and engines
- Integrated fault campaign management across formal, simulation, and emulation
- Common fault results database unifies diagnostic coverage
- Proven requirements traceability, enabling FMEDA integration

Example Design and FMEDA

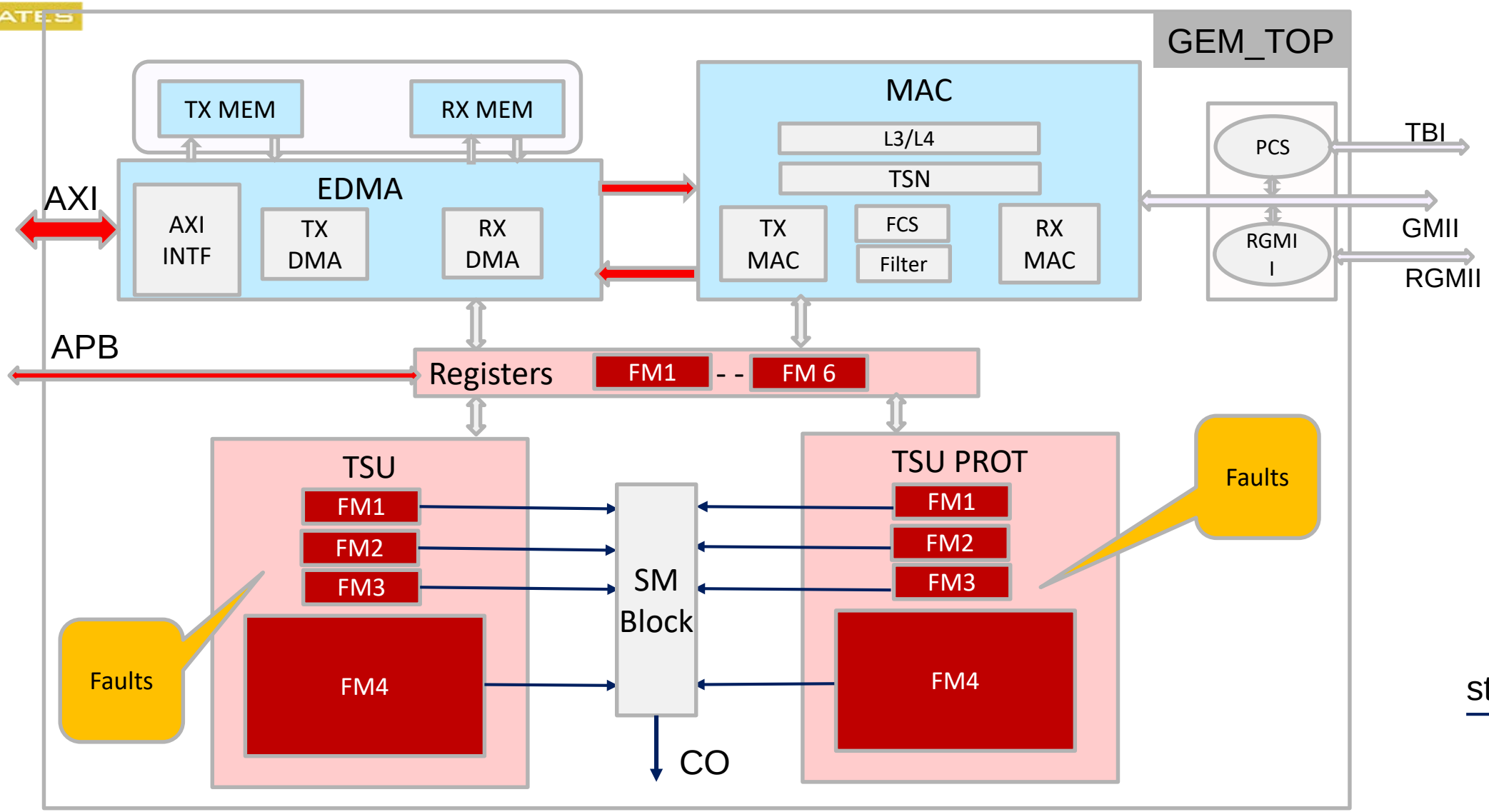
Safety Mechanisms in Ethernet IP



GEM Block – FMEDA Analysis

Block or Subblock	λ [FIT]	Failure Mode	FM Distribution	Effect Description of FM	SM Implemented
TSU	0.0719	Fault in TSU compare pulse	0.9%	TSU compare interrupt is incorrect	Compare logic is duplicated
TSU	0.0719	Fault in TSU seconds increment pulse	0.9%	The TSU seconds interrupt is incorrect	Interrupt logic is duplicated
TSU	0.0719	Fault in generation of the TSU strobe pulse to the registers	0.9%	The timer value may not be captured or captured incorrectly	Strobe Pulse Logic is duplicated
TSU	0.0719	Fault in TSU timer output value	97.3%	TX/RX timestamp is corrupted, output TSU timer value to local system will be invalid, Timer value read back in registers is also invalid.	Timer logic is duplicated
Registers	0.3013	Fault in static configuration outputs from the registers	95%	Unpredictable behavior of IP	Parity generation and detection

Ethernet IP – GEM Block

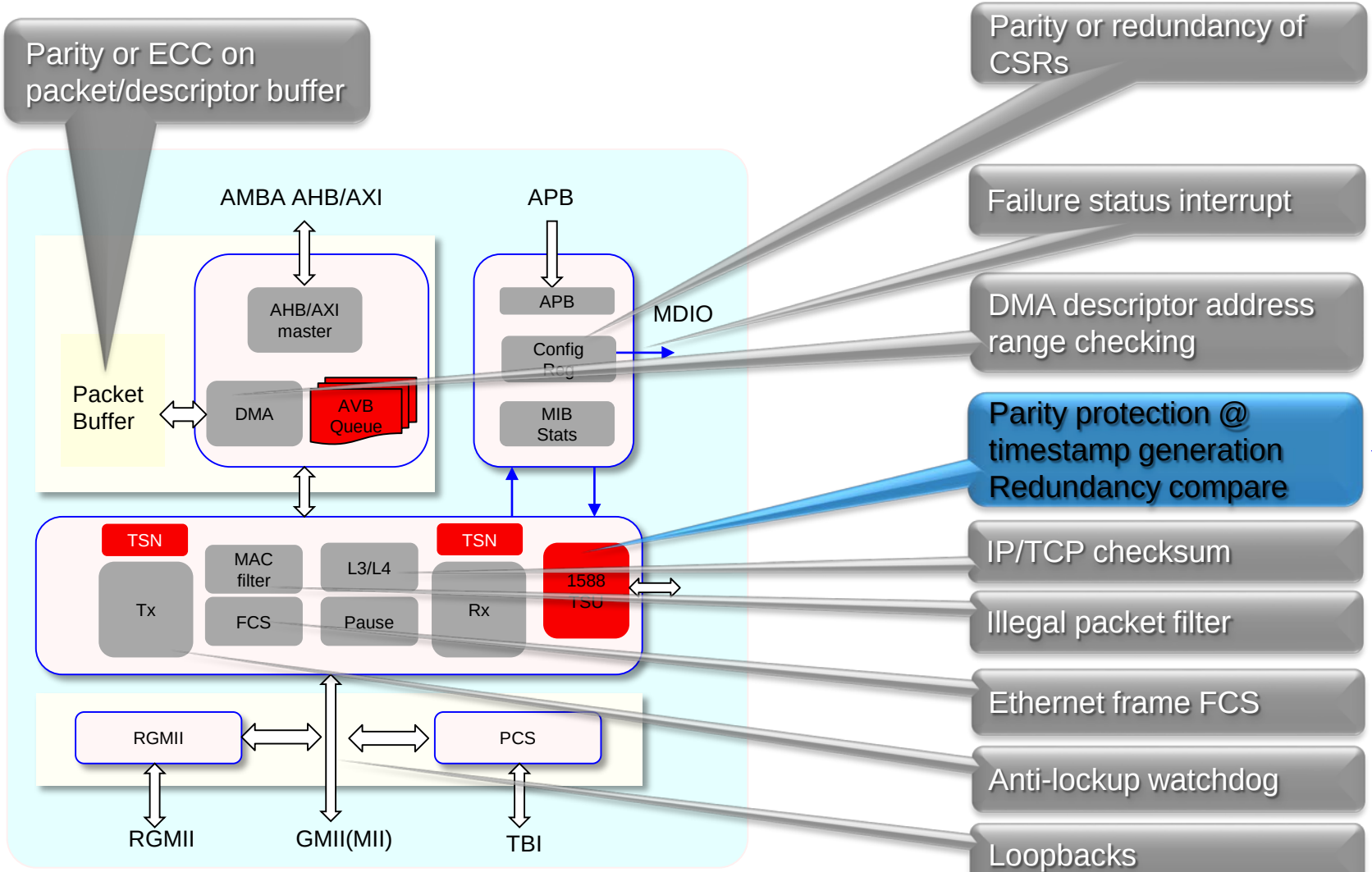


GEM Block – FMEDA Verification

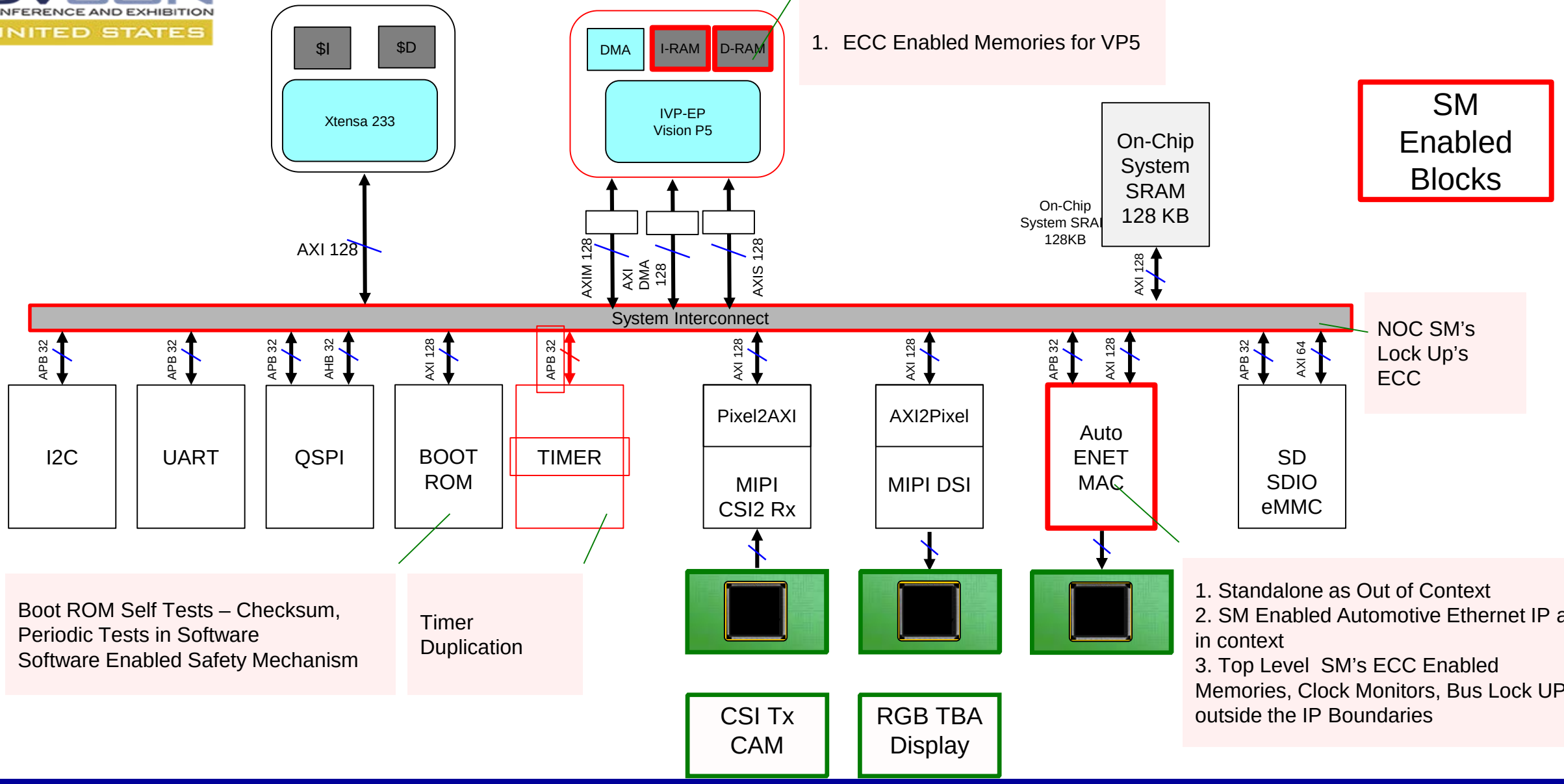
Block or Subblock	λ [FIT]	Failure Mode	FM Distribution	DC Number Estimated	DC Number Achieved
TSU	0.0719	Fault in TSU compare pulse	0.9%	95%	96%
TSU	0.0719	Fault in TSU seconds increment pulse	0.9%	95%	98%
TSU	0.0719	Fault in generation of the TSU strobe pulse to the registers	0.9%	95%	78%
TSU	0.0719	Fault in TSU timer output value	97.3%	95%	100%
Registers	0.3013	Fault in static configuration outputs from the registers	95%	90%	92.5%

Demo

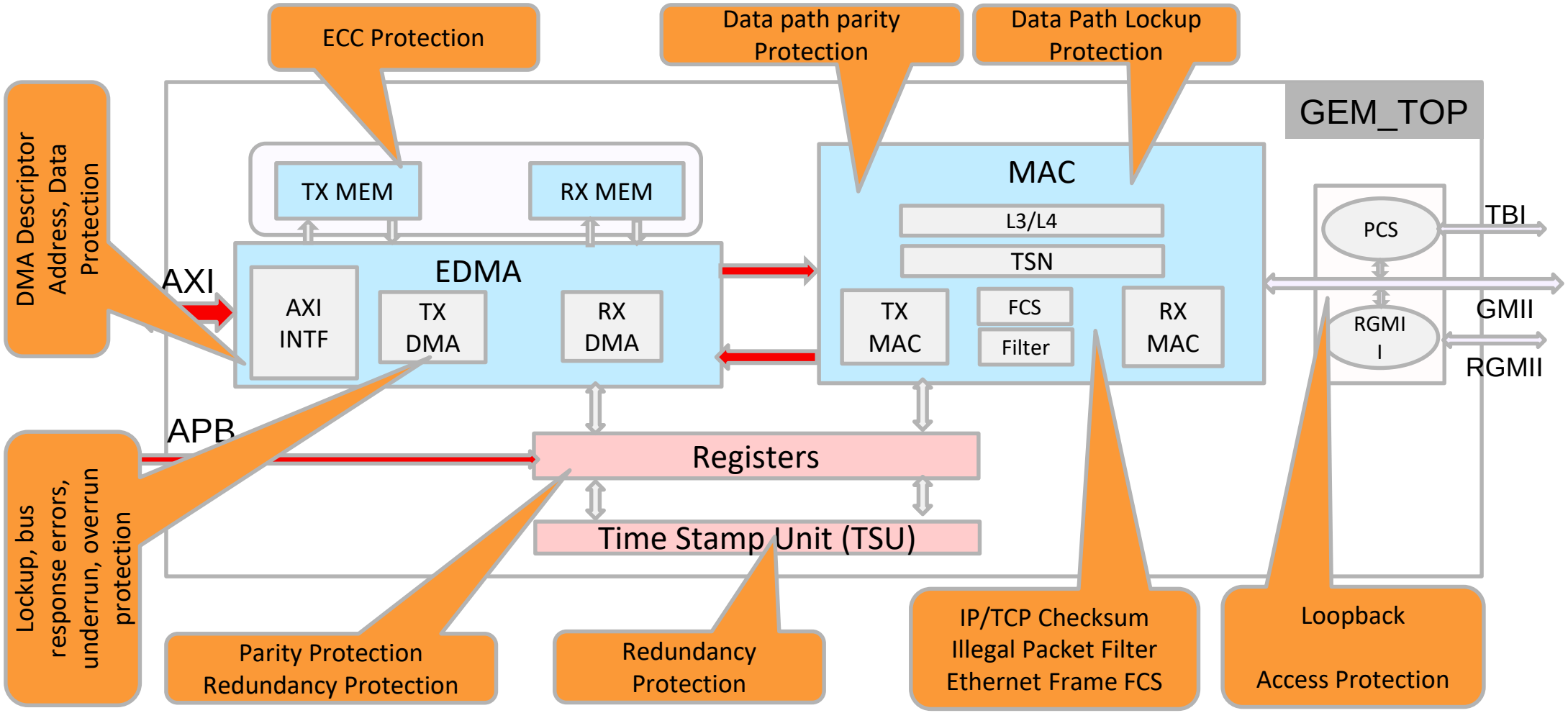
Safety Mechanisms in Ethernet IP



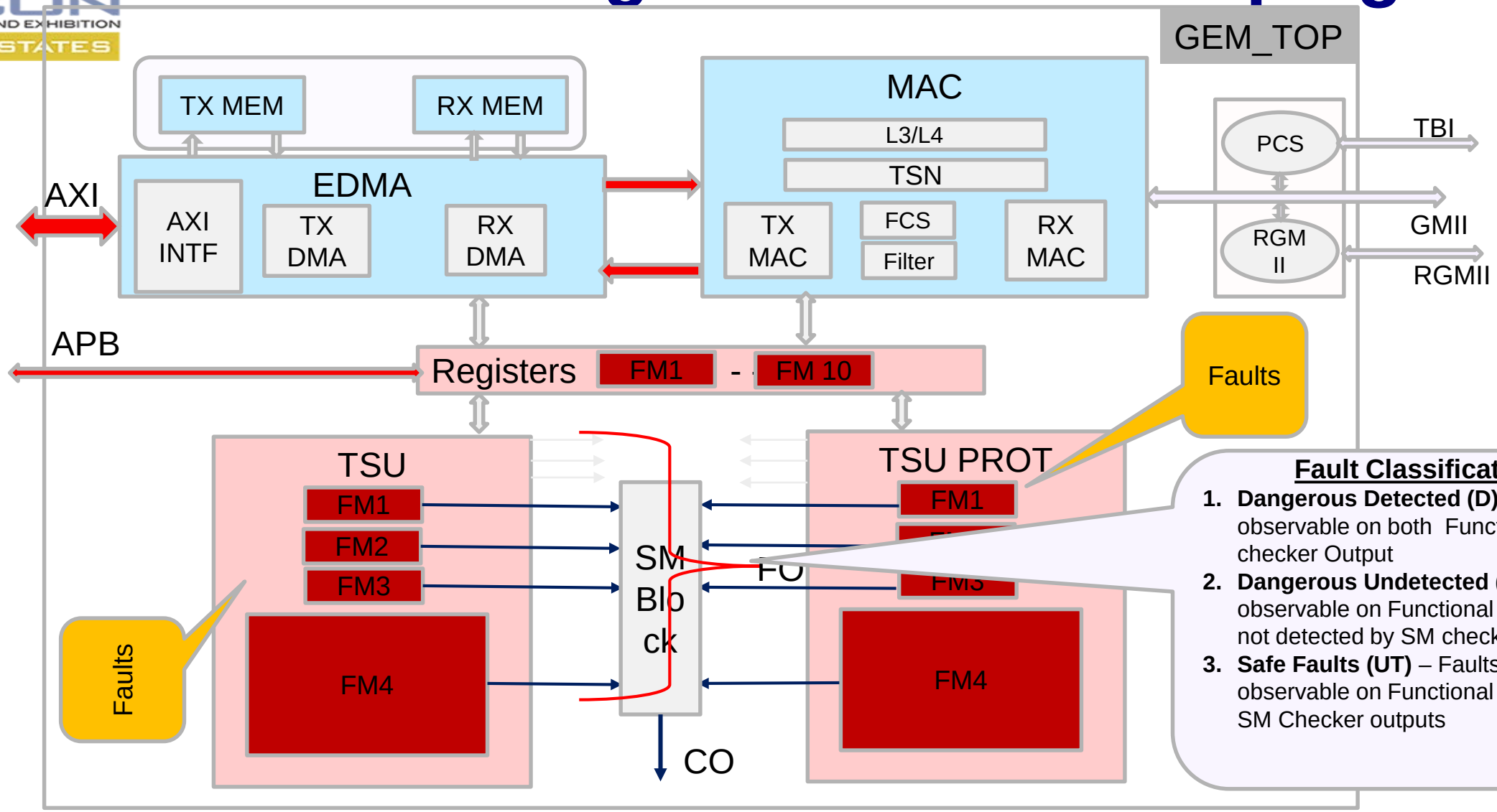
ADAS Platform – SM Enabled



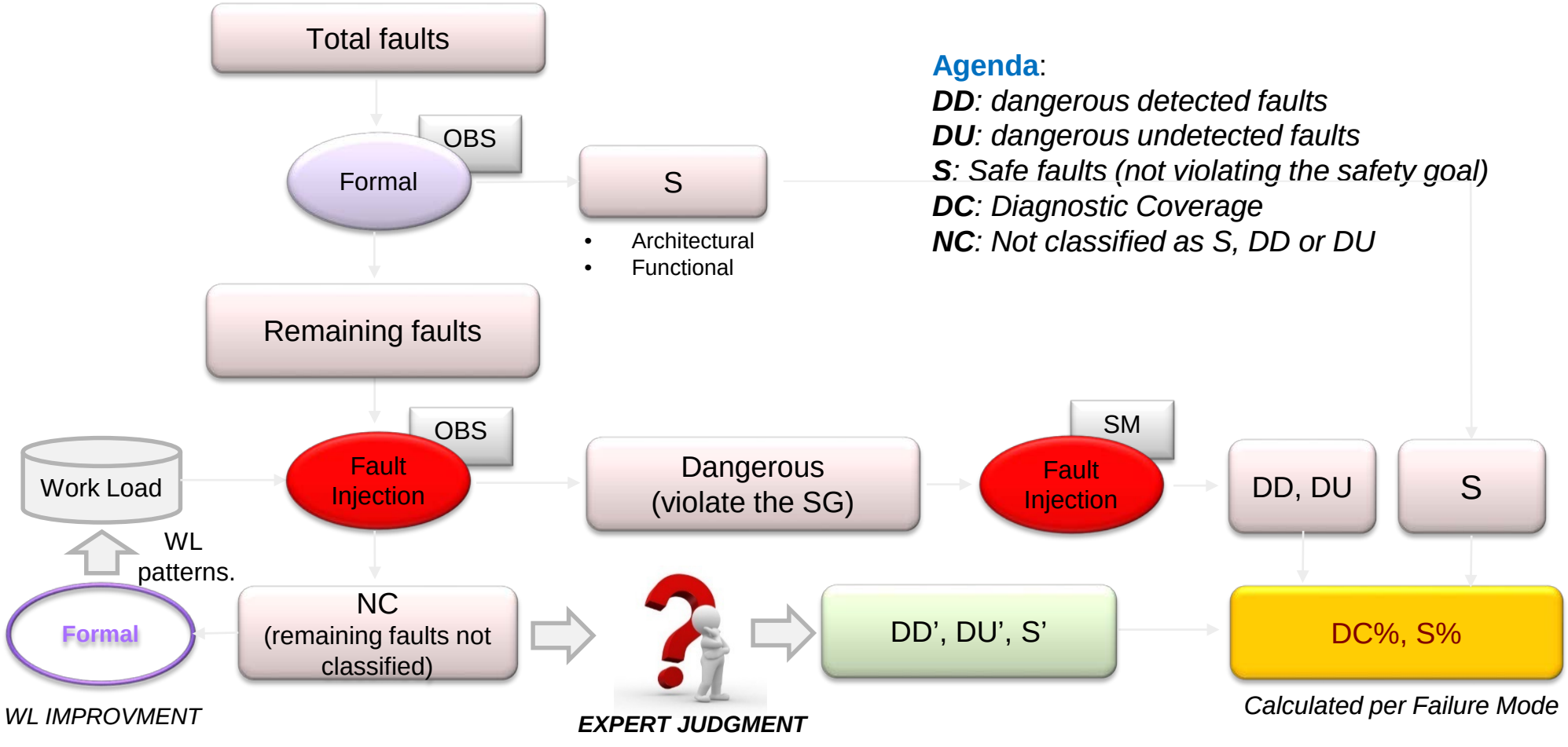
GEM Block Diagram – SM View



GEM Block Diagram – Fault Campaign view



ISO26262 Compliant Fault Classification



Agenda:
***DD**: dangerous detected faults*
***DU**: dangerous undetected faults*
***S**: Safe faults (not violating the safety goal)*
***DC**: Diagnostic Coverage*
***NC**: Not classified as S, DD or DU*

Optional. *If applied the user shall to provide additional evidences in place of fault injection and formal analysis to justify the expert judgment*

Demo Setup and Run the VNC

Chosen 5 Failure modes for the demo showcasing the solution and automation capabilities

- fm_tsu_comp_pulse - Fault in TSU comp pulse – show cases the ranking capability and undetected faults as SM is not implemented
- fm_tsu_tmr_op_val - Fault in TSU timer Output value - show cases the ranking capability and detected faults as SM is implemented
- fm_tsu_sec_incr - Fault in TSU seconds increment pulse – run campaign for the module TSU
- fm_tsu_tmr_op_val - Fault in generation of the TSU strobe pulse to the registers - run campaign for the module TSU

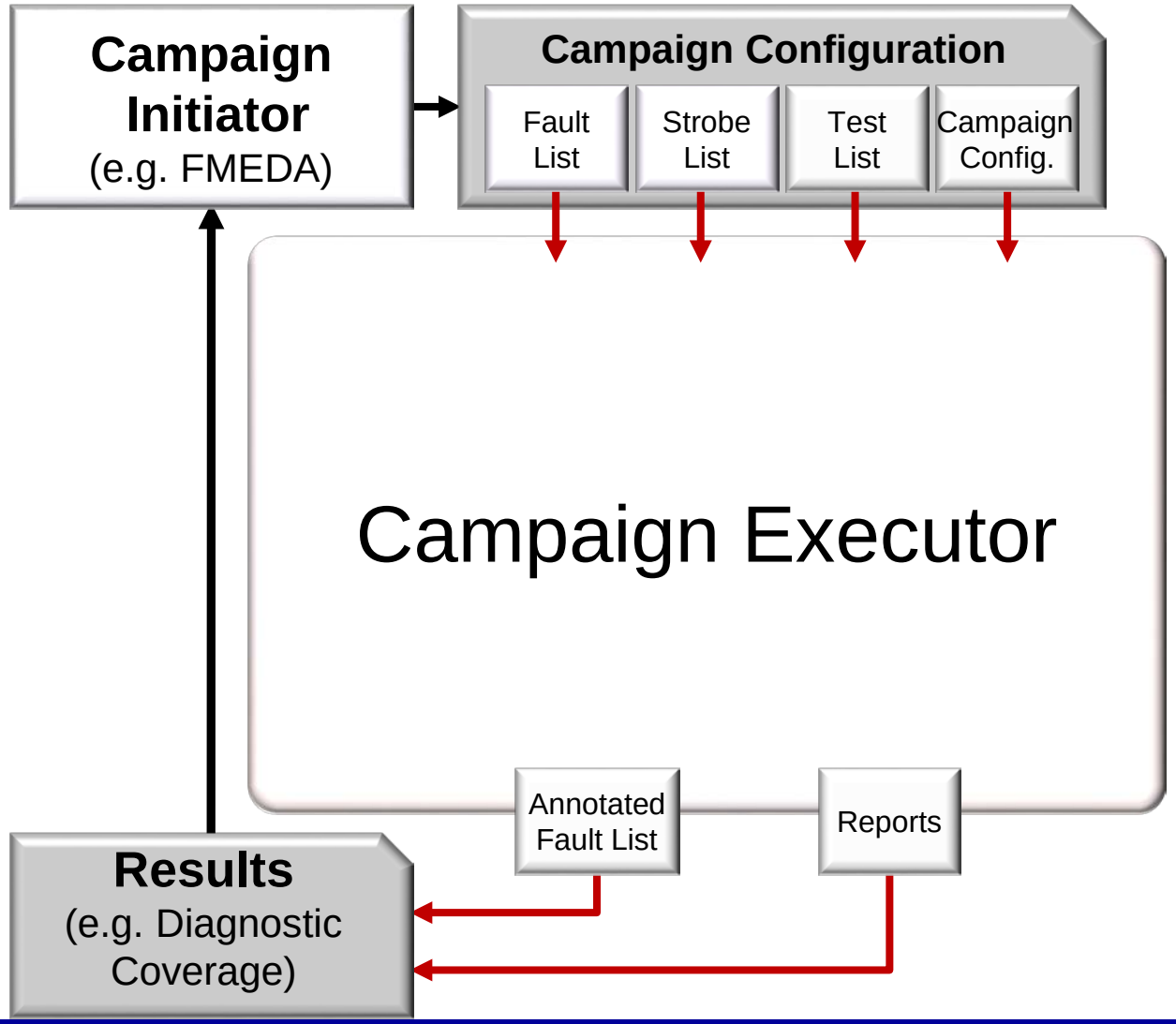
PRE RUN

Live Run

- fm_tsu_tmr_op_val_samp - Fault in generation of the TSU strobe pulse to the registers - run campaign for the module TSU

Block or Subblock	λ [FIT]	Failure Mode	FM Distribution	Effect Description of FM	SM Implemented
TSU	0.0719	Fault in TSU compare pulse	0.9%	TSU compare interrupt is incorrect	Incomplete
TSU	0.0719	Fault in TSU seconds increment pulse	0.9%	The TSU seconds interrupt is incorrect	Incomplete
TSU	0.0719	Fault in generation of the TSU strobe pulse to the registers	0.9%	The timer value may not be captured or captured incorrectly	Incomplete
TSU	0.0719	Fault in TSU timer output value	97.3%	TX/RX timestamp is corrupted, output TSU timer value to local system will be invalid, Timer value read back in registers is also invalid.	Timer is duplicated
Registers	0.3013	Fault in static configuration outputs from the registers	95%	Unpredictable behavior of IP	Parity generation and detection

Fault Campaign Executor - Interface



Inputs: FMEDA info

- Fault List
  - Definition of the faults to be injected
- Strobe List
  - Definition of the observation points

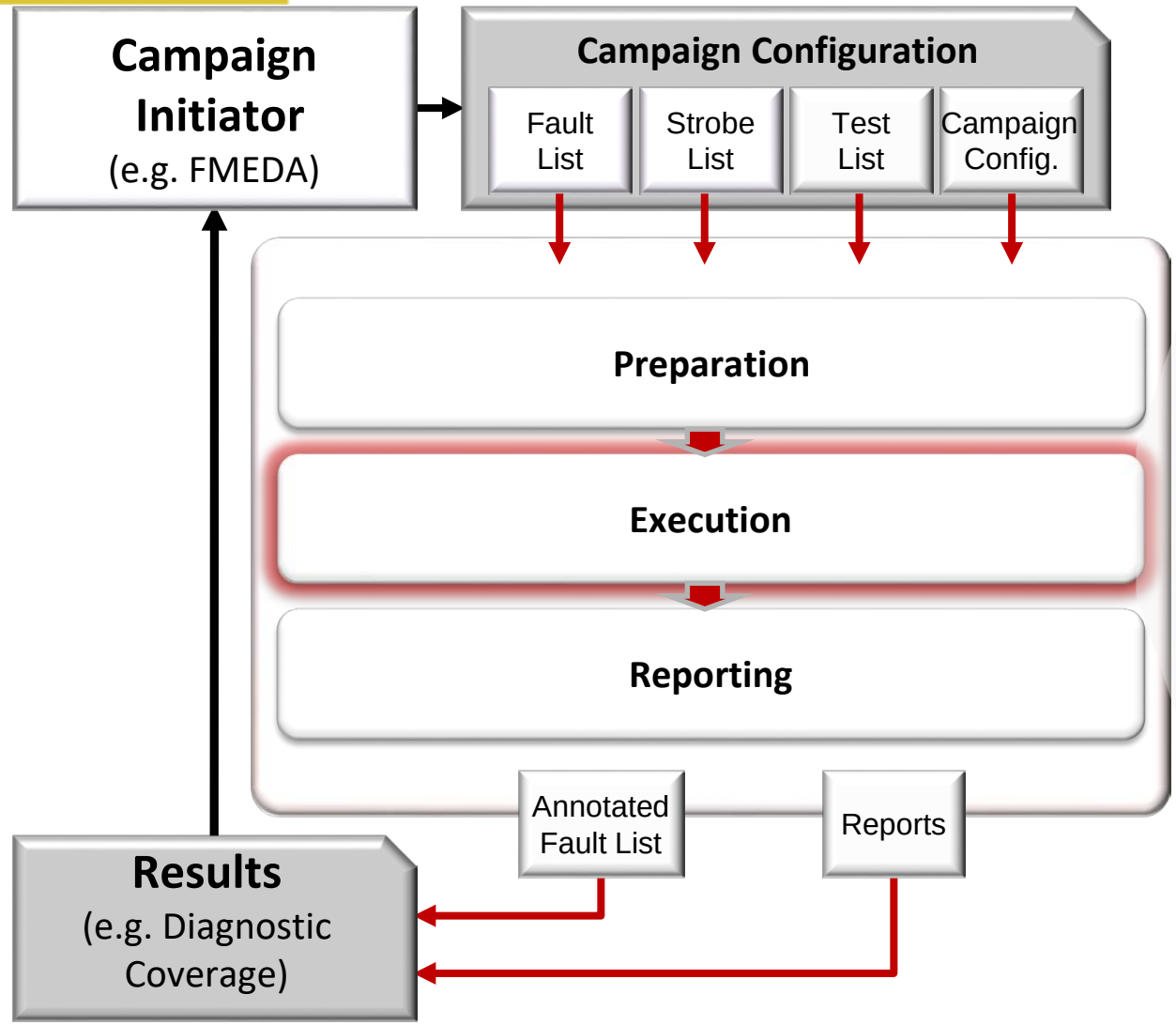
Inputs: FS Verification Engineer

- Test List
  - Tests to be used during the campaign
- Campaign Configuration:
  - Define the campaign parameters

Outputs:

- Annotated Fault List
  - Fault classification is back annotated
- Reports
  - Various kind according to the use case

Fault Campaign Executor - Interface



- Test selection
  - Execute the user defined list of tests
- Good Simulation
  - Fault instrumentation
  - Generate strobe data for each selected test
- Fault Simulation Setup
  - Prepare fault simulation including static and dynamic (formal) fault set optimization
- Fault Simulation Execution
  - Simulate each fault with the selected tests

Campaign Reports - Abstract

```
--- Campaign Execution: Statistics and D
Report Generation Date:             : 2018/02/06 12:49:11
Tool Version                        : XFS
FAULT_CAMPAIGN_NAME                 : fs_gem_demo_tmr_op_val
FAULT_CAMPAIGN_TYPE                 : permanent
FS_TC_CALC                          : pessimistic (PD faults considered U faults)
--- Fault analyzed                        faults                 prime faults
nr faults                      [F] :        964 [100.0%]          964 [100.0%]
nr faults untestable          [UT] :        336 [ 34.9%]          336 [ 34.9%]
nr faults testable             [T] :        628 [ 65.1%]          628 [ 65.1%]
--- Faultsim Execution Statistics
sampling factor                     :         100%
(100% means no sampling)
nr faults selected for faultsim     :        628
nr faults todo                      :          0 [  0.0%]
nr faults processed                 :        628 [100.0%]
nr faults faultsim runs             :       1661
nr faults faultsim runs / fault     :          2.64
--- Sampling options
Sampled Fault Type                  :    SA0+SA1
Sampling Seed                       :
Sampling Percentage                 :        100
Sampled Number                      :
Sampling Scope                      :   testable
--- Fault Classification (dual strobe)
nr faults UNKNOWN             [UK] :         36 [  3.7%]           36 [  3.7%]
nr faults UNDETECTED           [U] :          0 [  0.0%]            0 [  0.0%]
nr faults DETECTED             [D] :        592 [ 61.4%]          592 [ 61.4%]
--- Work Load Qualification
Toggle Coverage %                   : 41.53012642255185
--- Test Coverage
Test  Coverage                      :        1.0
Fault Coverage                      : 0.63793105
```

Safe Faults by Formal

Possibly need to improve work load

Detected Faults

Test Coverage = (D/(D+U))

Fault Coverage = (D/(D+U+UT))

Campaign Reports - Abstract

```
--------------------------------------------------------------------------------
--- Campaign Execution: Statistics and Data
--------------------------------------------------------------------------------
Report Generation Date:             : 2018/02/07 05:50:09
Tool Version                        : XFS
FAULT_CAMPAIGN_NAME                 : fs_gem_demo_tmr_op_val_ser_net
FAULT_CAMPAIGN_TYPE                 : permanent
FS_TC_CALC                          : pessimistic (PD faults considered U faults)
--------------------------------------------------------------------------------
--- Fault analyzed                        faults                prime faults
--------------------------------------------------------------------------------
nr faults                      [F] :      16170 [100.0%]         9499 [100.0%]
nr faults untestable          [UT] :       1049 [  6.5%]          551 [  5.8%]
nr faults testable             [T] :      15121 [ 93.5%]         8948 [ 94.2%]
--------------------------------------------------------------------------------
--- Faultsim Execution Statistics
--------------------------------------------------------------------------------
sampling factor                     :           5%
(100% means no sampling)
nr faults selected for faultsim     :       8948
nr faults todo                      :       8501 [ 95.0%]
nr faults processed                 :        447 [  4.0%]
nr faultsim runs                    :       1062
nr faultsim runs / fault            :          2.38
--------------------------------------------------------------------------------
--- Sampling options
--------------------------------------------------------------------------------
Sampled Fault Type                  :    sa0+sa1
Sampling Seed                       :
Sampling Percentage                 :          5
Sampled Number                      :
Sampling Scope                      : testable_prime
--------------------------------------------------------------------------------
--- Fault Classification (dual strobe)
--------------------------------------------------------------------------------
nr faults UNKNOWN             [UK] :        197 [  1.2%]          197 [  2.1%]
nr faults UNDETECTED           [U] :          0 [  0.0%]            0 [  0.0%]
nr faults DETECTED             [D] :        250 [  1.5%]          250 [  2.6%]
--------------------------------------------------------------------------------
--- Work Load Qualification
--------------------------------------------------------------------------------
Toggle Coverage %                   : 15.051711726986445
--------------------------------------------------------------------------------
--- Test Coverage
--------------------------------------------------------------------------------
Test  Coverage                      :        1.0
Fault Coverage                      : 0.19245574
```

Sampled Fault Processed

Test Coverage = (D/(D+U))

Fault Coverage = (D/(D+U+UT))

Summary

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Summary

- Autonomous cars are coming and ‘Mind-Off’ driving is expected to be real by the mid 2020s
- ISO 26262 is the automotive standard that defines the processes to follow, the performance level for hardware and software performance and the compliance process
- A systematic analysis technique such as the FMEDA is essential for meeting ISO 26262 metrics
- The complexity of ADAS SoCs requires a new holistic approach to functional verification and functional safety
  - Functional safety and functional verification are complementary problems
- A multi-engine automated solution is required to meet ASIL certification goals in a timely manner.

Questions

www.cadence.com/automotive

DVCon Slide Guidelines

- Use Arial or Helvetica font for slide text
- Use Courier-new or Courier font for code
- First-order bullets should be 24 to 28 point
  - Second-order bullets should be 24 to 26 point
    - Third-order bullets should be 22 to 24 point
    - Code should be at least 18 point
- Your presentation will be shown in a very large room
  - These font guidelines will help ensure everyone can read you slides!

~~No Company Logo except on title slide!~~

Code and Notes

Code should be enclosed in text boxes (using a background color is optional)

Code should be 18pt Courier-bold, or larger

```
module example
(input  logic foo,
 output logic bar
);

  initial begin
    $display (“Hello World!”);

endmodule
```

Informational boxes should be 18pt Arial-bold, or larger (using a background color is optional)