Functional Safety for Semiconductor Designs

Alessandra Nardi, Software Engineering Group Director, Automotive Solutions

Electronic Design Process Symposium 2018
Milpitas - September 14, 2018

ELECTRONIC DESIGN PROCESS SYMPOSIUM
IEEE-EDPS.ORG
September 13 -14, 2018
In Milpitas, California

cadence®

Contributors: Antonino Armato

Overview

- Introduction and automotive overview
- Basics of Functional Safety (ISO26262)
- Functional Safety Analysis
- Functional Safety requirements driving the traditional design flow
- Conclusions

What are we Talking About?

- (9 deaths & 1000+injuries)/day due to distracted driving
- Social/economical push to autonomous driving/ADAS (*)

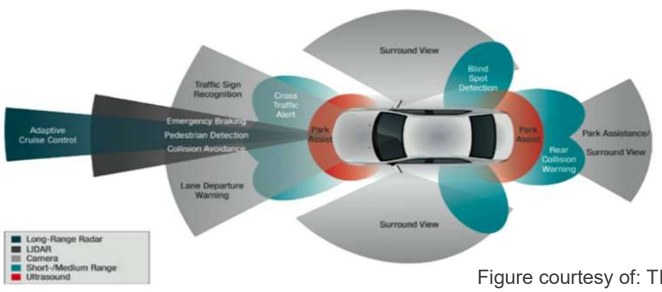
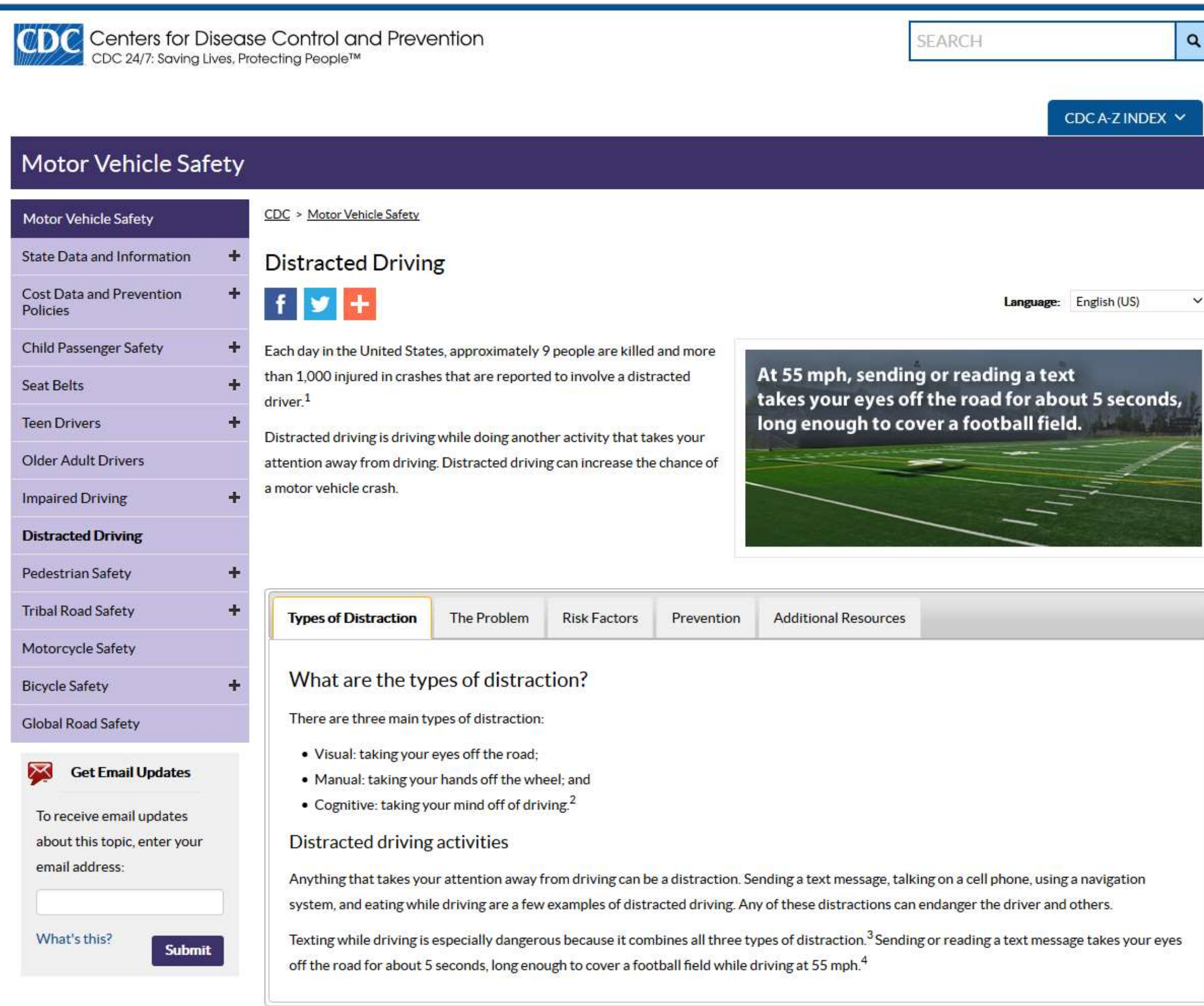


Figure courtesy of: TI

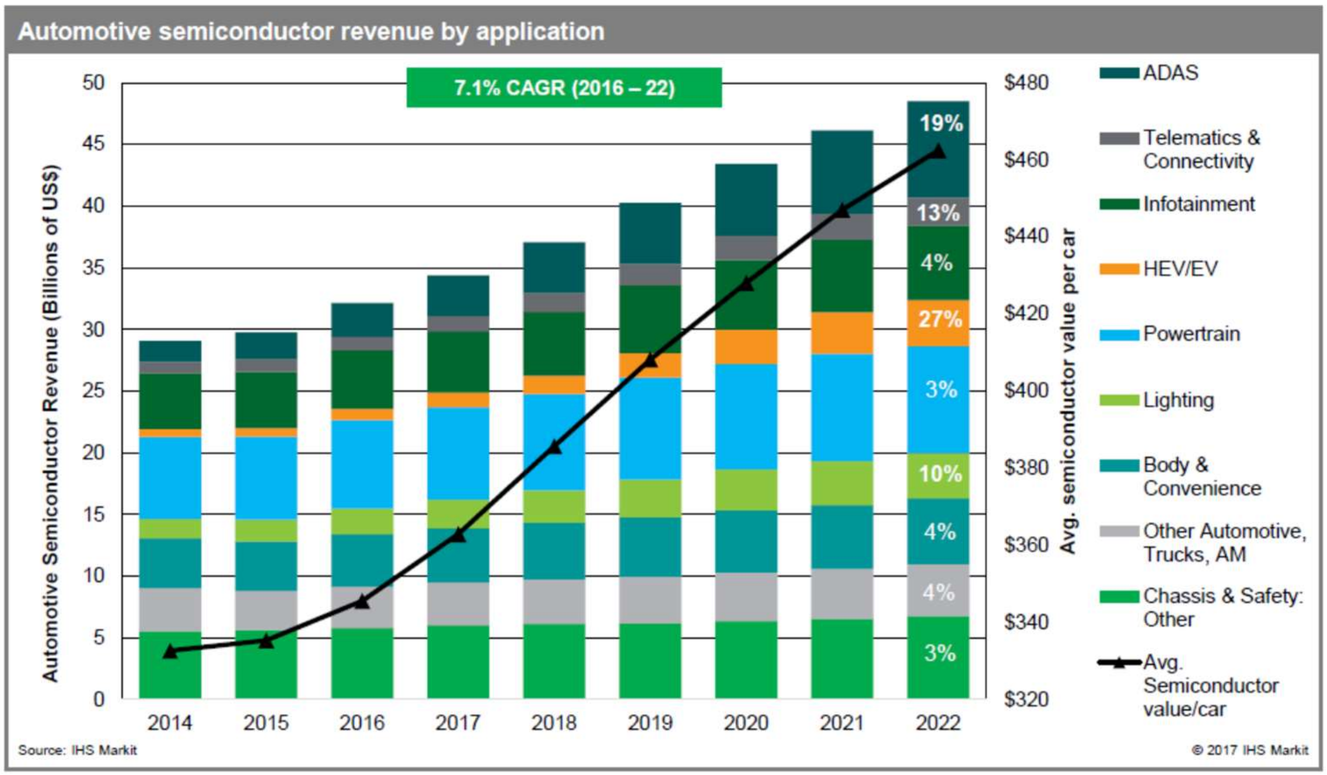


Source: https://www.cdc.gov/motorvehiclesafety/distracted_driving/index.html

(*) ADAS: Advanced Driver Assistance Systems

Automotive Semiconductor Growth

Major forces shaping the automotive industry



Vehicle electrification

Growth of Autonomous Driving

Increased Connectivity

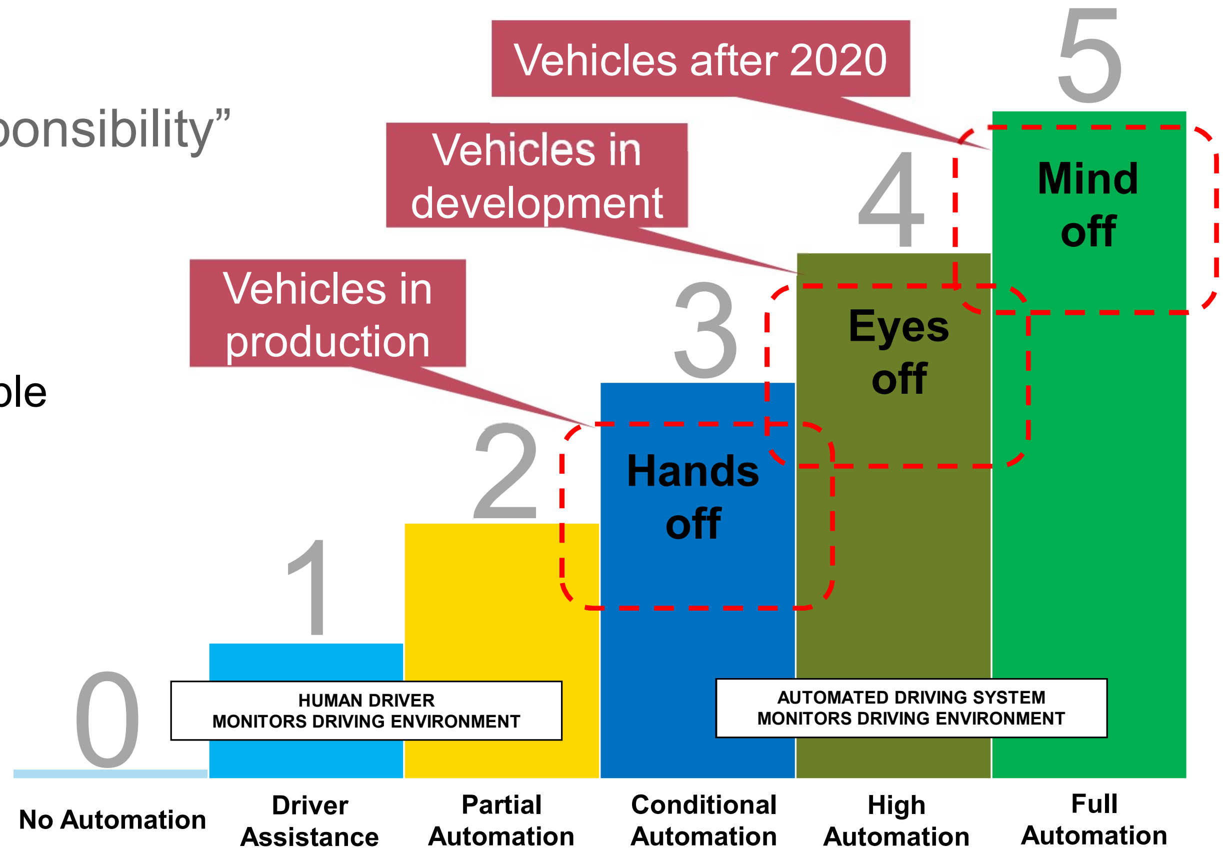
IVI – AR/VR

cādence®

Autonomous Driving

"With great power comes great responsibility"

- 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
  – Quality

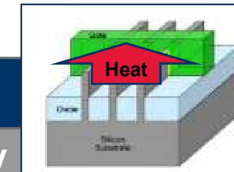


LEVELS OF DRIVING AUTOMATION AS DEFINED IN SAE INTERNATIONAL STANDARD J3016

- ADAS and autonomous driving are changing the game:
- Requirements are rippling down the chain
- Functional Safety requirements have entered the traditional design flow

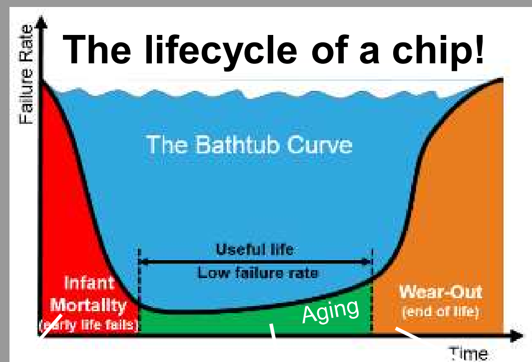
cādence

What Makes Automotive so Challenging?

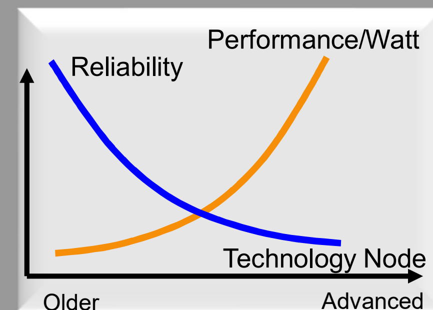


Lifetime Reliability issues

- Aging
- Electro-Migration
- Process Variation
- Thermal Fatigue
- ESD



ADAS complexity amplifies the problem!



Quality (Zero Defect)

"Design for Test"
Eliminate early life failing parts

- **Design meets specifications at start of life**
- Target is 0 DPPM (Defective Parts Per Million)

Production test = Wafer probe + 3 temp. | Minimize area imp. Optimize Test Time

Reliability (AEC-Q100)

"Design for Robustness"
Minimize lifetime reliability issues

- **Design meets specification until the end of life**
- FIT (Failures per billion hours of operation)

Aging | ESD, Latchup | EMI | Thermal

Functional Safety (ISO26262)

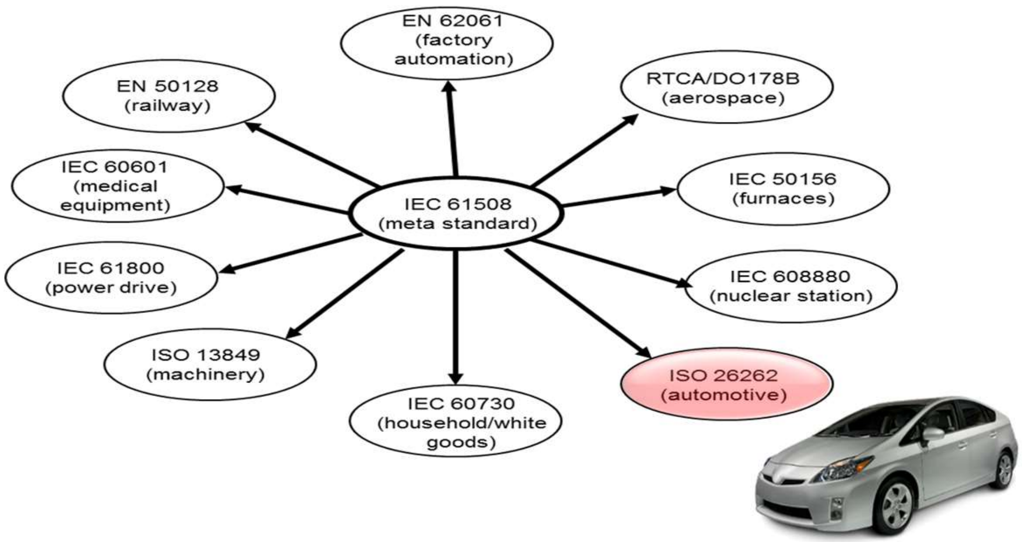
„Design for Safety"
Detect faults and protect the system integrity

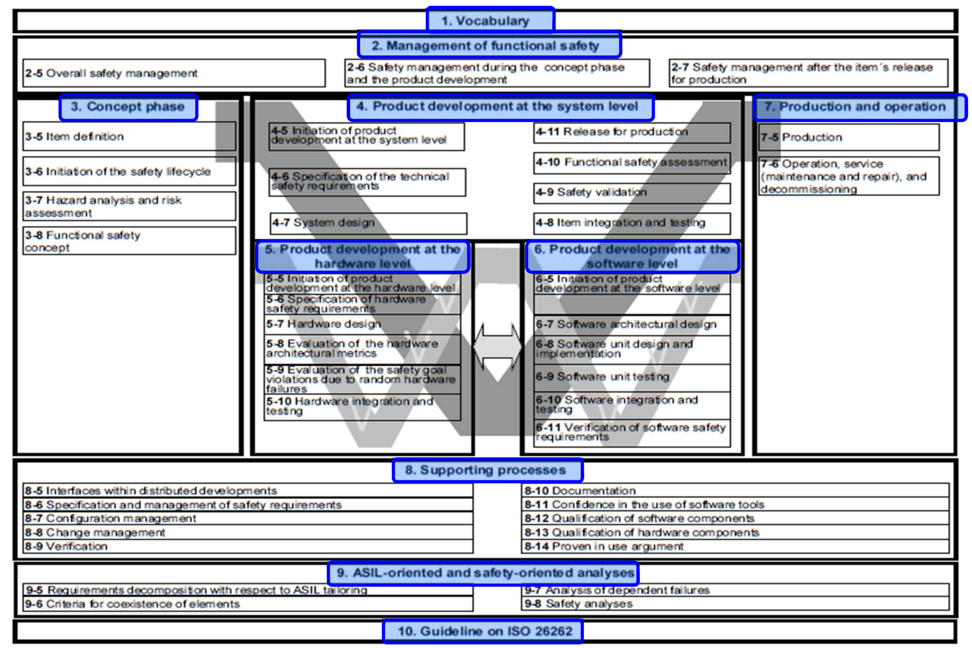
- **Design is amended with safety mechanisms which detect faults**

FMEDA FIT rate | Safety SoC Architecture | Safety Verification

cadence

Functional Safety Standards



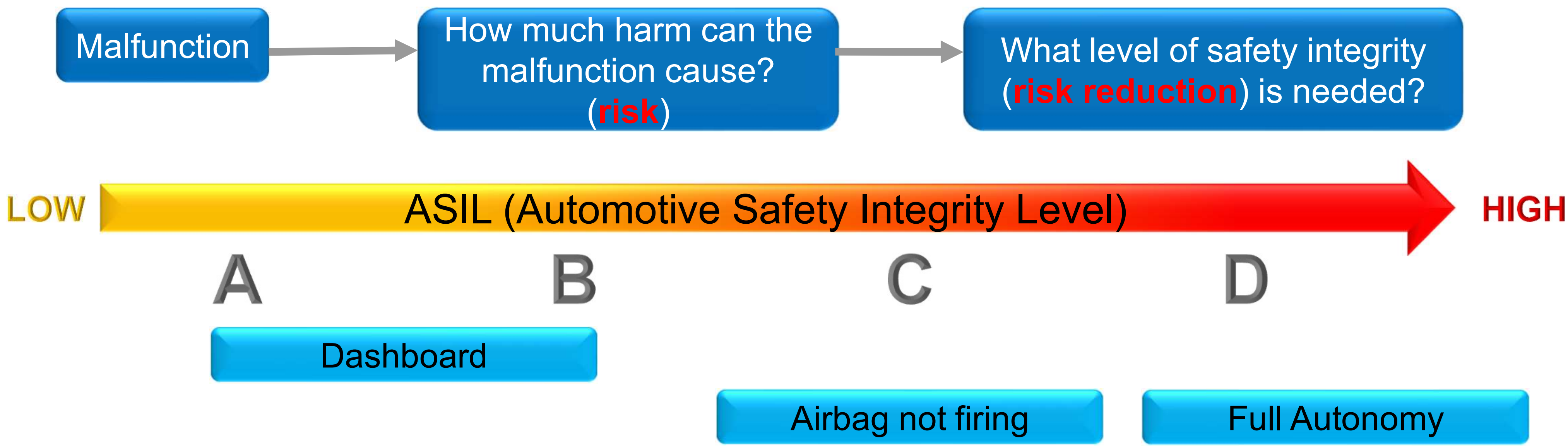


ISO 26262 defines

- 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)

cādence®

Functional Safety Principles

Covers random and systematic errors

What can go wrong ?

Systematic Failures

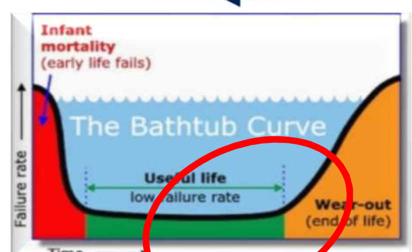
- Example: Software bugs
- Addressed by development processes (planning, traceability, documentation, specs, …)

Functional Safety Management

Random Failures

- Example: component malfunction, noise injection
- Addressed by design/verification of safety mechanisms to correct/detect faults

Design/Analysis ⇄ Verification



Hardware metrics to become part of the traditional design flow

cadence®

Hardware Random Failure Metrics

A measure of the effectiveness of the solution to detect random failures

LOW — ASIL (Automotive Safety Integrity Level) — HIGH

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

PMHF: Probabilistic Metrics for Hardware Failures

- Absolute value representing the residual likelihood of failure
- Expressed in FIT (Failure in Time), $1FIT=10^{-9}/h$

SPFM: Single Point Fault Metric

- Relative value reflecting the robustness to single point faults by coverage from safety mechanisms

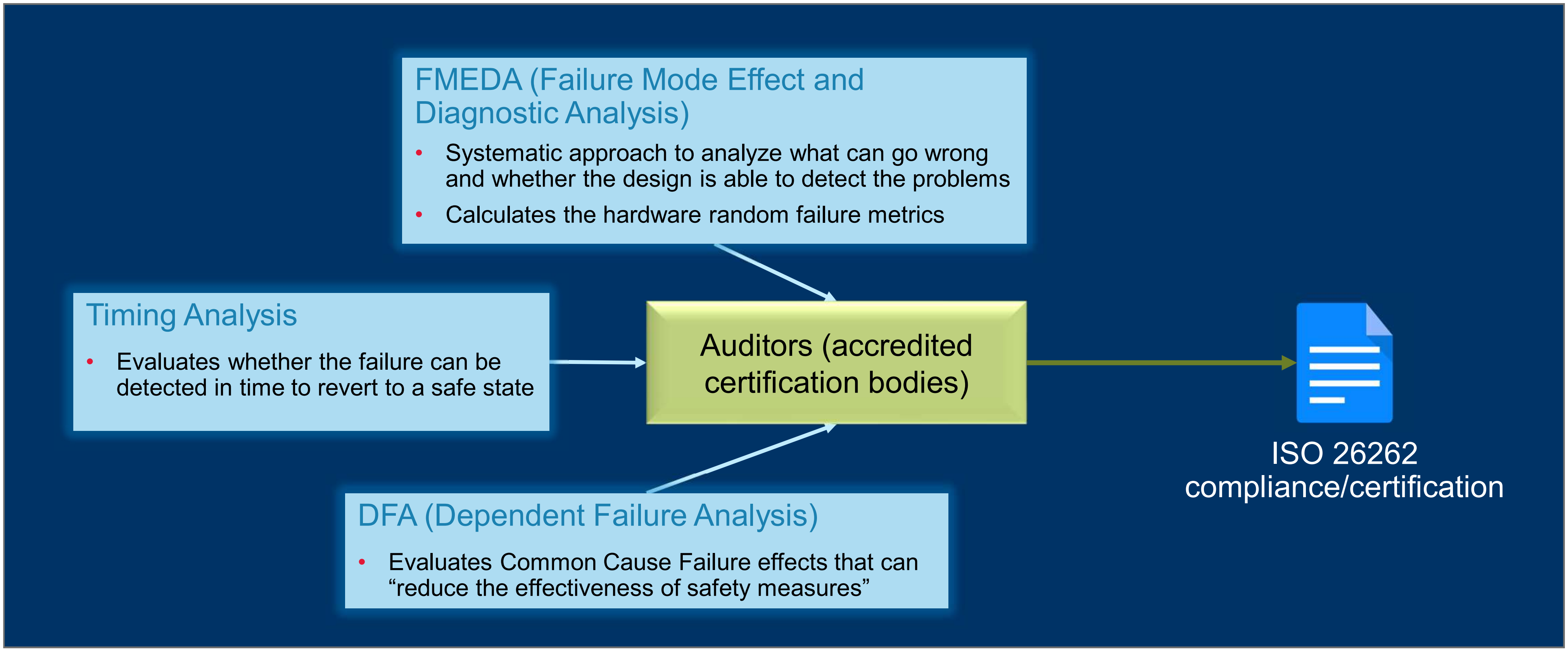
LFM: Latent Fault Metric

- Relative value reflecting the robustness to latent faults by coverage from safety mechanisms or by the driver recognizing that the fault exists before the violation of the safety goal

cādence®

Functional Safety Analysis

How do we measure Functional Safety?



cādence®

FMEDA - Failure Mode Effect and Diagnostic Analysis

- FMEDA is a structured approach to define the failure modes and the diagnostic capabilities of a hardware component
- It evaluates Safety Architecture (collection of safety mechanisms) and calculates the safety performance of the system (SPFM, LFM, PMHF).

					Total SPFM		94.78%
Part	Sub-part	Failure mode	Safety Goal	Failure Rate	Safety Mechanisms	DC	SPFM
CPU	Decoder	Incorrect Instruction Flow caused by a fault the decode logic	SG1	3.92E-03	SM1	90%	90%
	Multiplier	Incorrect Instruction Execution caused by a fault in the multiplier	SG1	9.09E-03			0%
	Adder	Incorrect Instruction Execution caused by a fault in the adder	SG1	2.25E-03	SM2	90%	90%
	Divider	Incorrect Instruction Execution caused by a fault in the divider	SG1	1.60E-03			0%
	Fetch	Incorrect Instruction Flow caused by a fault the fetch logic	SG1	1.83E-02	SM3	60%	60%
	Cache	Wrong data cell caused by a fault in the cache	SG1	3.98E-01	SM4	99%	99%

Can I potentially affect a safety goal ?

How likely is my block to fail ?

Can I catch the failure ?

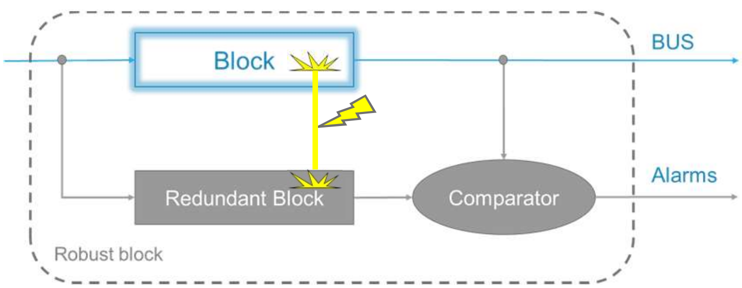
How likely that I can detect the failure ?

FMEDA example for illustration purposes only

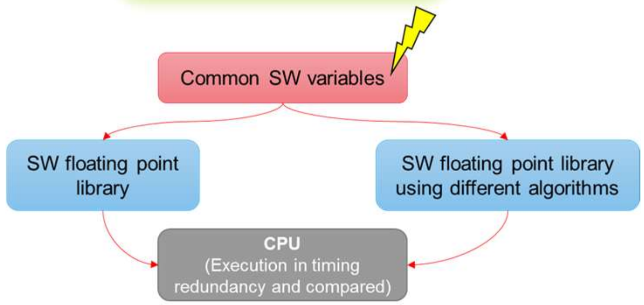
DFA – Dependent Failure Analysis

- Functional Safety can be achieved through redundancy of functionality
- This is effective only if redundant elements are independent
- DFA identifies single causes that could invalidate independence and violate a safety goal, e.g. it is an analysis of Common Cause Failures (CCF)
- For example, it considers architectural features such as:
  - similar and dissimilar redundant elements
  - different functions implemented with identical software or hardware elements

HW redundancy

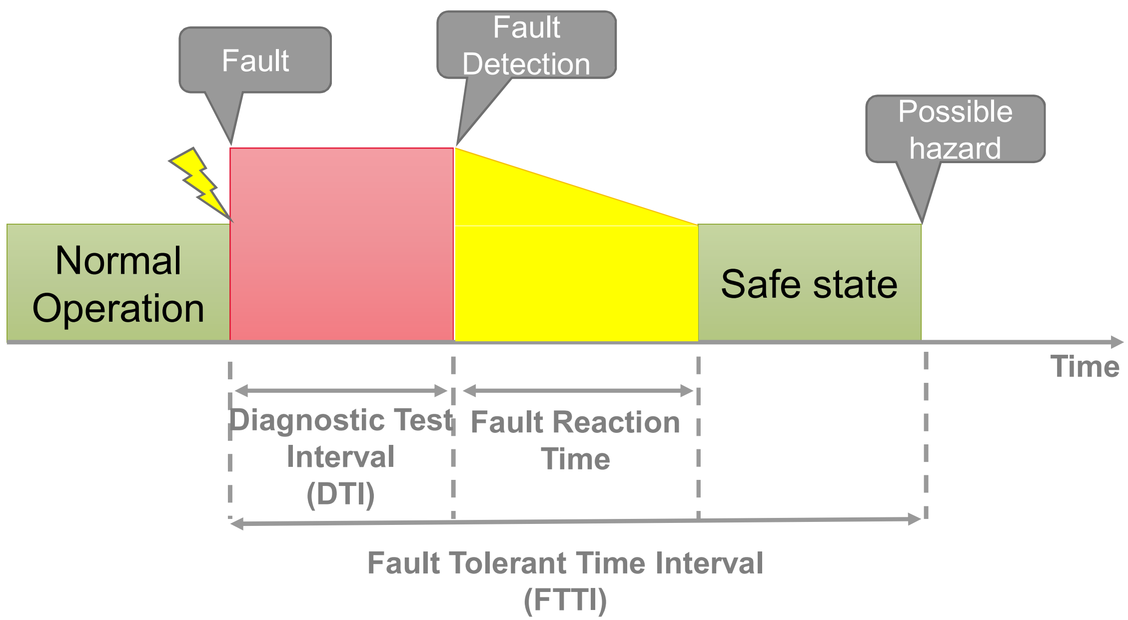


SW redundancy



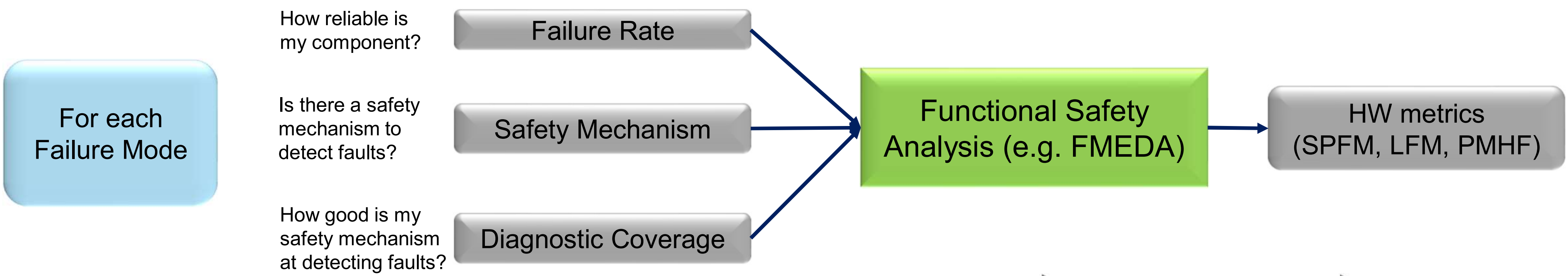
Timing Analysis

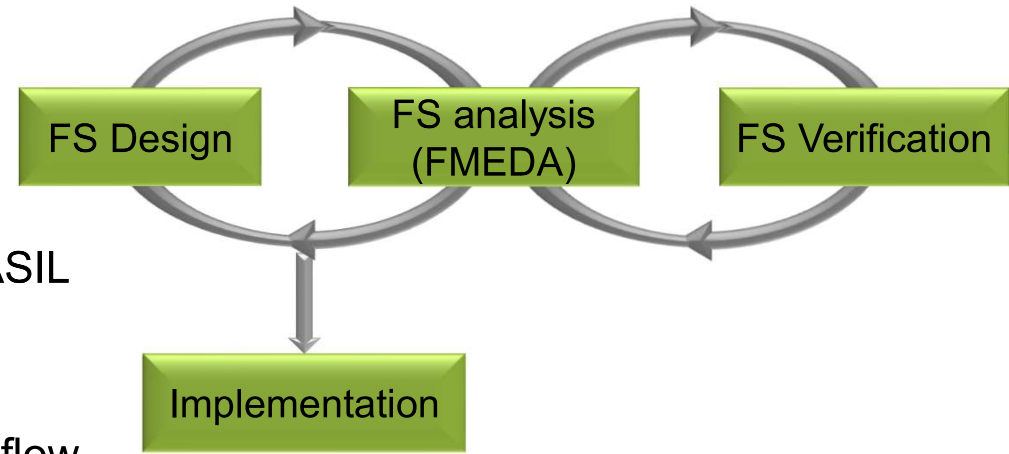
- Diagnostic Test Interval (DTI):
  – Amount of time between the executions of online diagnostic tests by a safety mechanism
- Fault Tolerant Time Interval (FTTI):
  – Time-span in which a fault or faults can be present in a system before a hazardous event occurs



cādence®

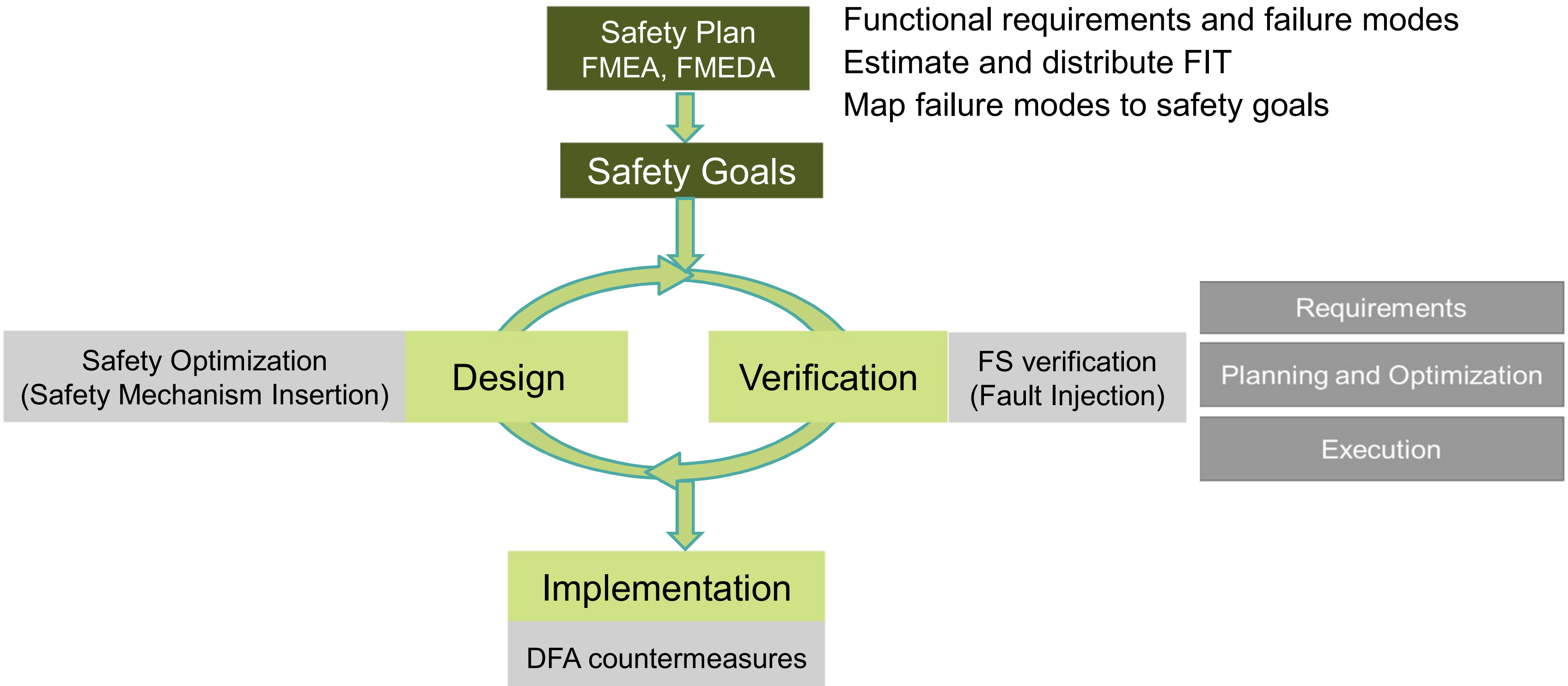
Functional Safety Analysis and Flow
Understanding and achieving ASIL HW metrics





- To improve the HW metrics and achieve the target ASIL
  - “Better” component
  - Better/Additional Safety Mechanism
- FS analysis drives the traditional design/verification flow

Functional Safety Design and Verification Solution



Functional Safety Analysis links to the traditional design/verification and implementation flow:

- To include safety mechanisms and meet the HW metrics/ASIL
- Safety metrics, ppa, verification time, automation are all to be considered

cadence

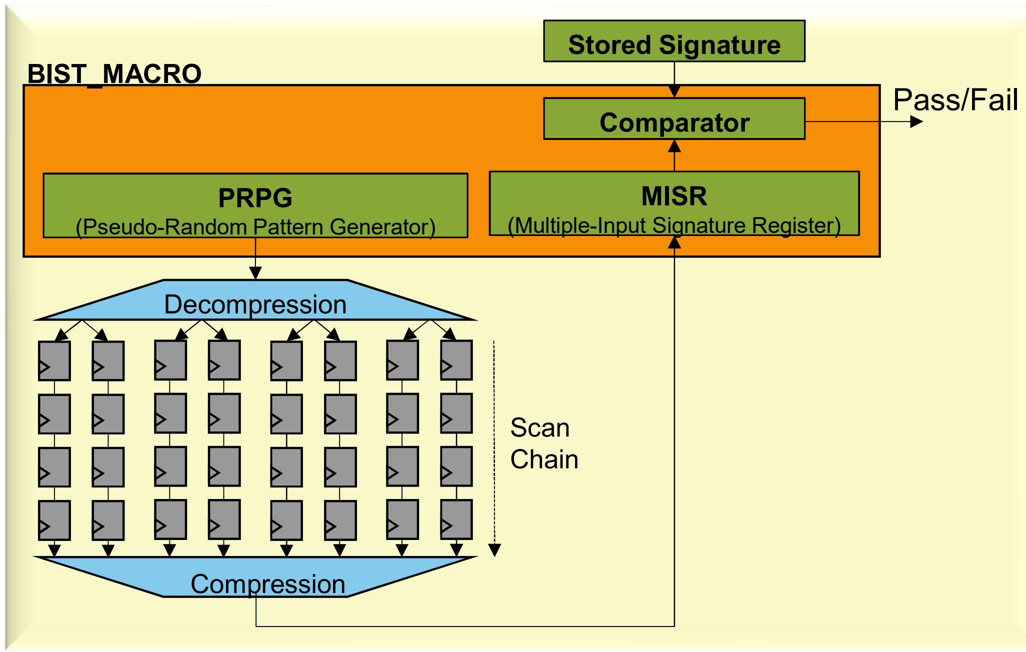
Built-In-Self-Test (BIST) for Functional Safety

An example of safety mechanism application and requirements

- BIST is used for automotive in-system/field testing for lifetime reliability to achieve desired ASIL
  - Power-On-Reset
  - Mission-Mode (which requires the system to be operational during the periodic in-field testing)

- Specific challenges and requirements:
  - High Coverage → meet ASIL requirements
  - Area overhead → cost
  - Short test-time → meet the Fault Tolerant Time Interval (DTI/FTTI) requirements
  - IEEE 1500: Isolate blocks for in-system LBIST

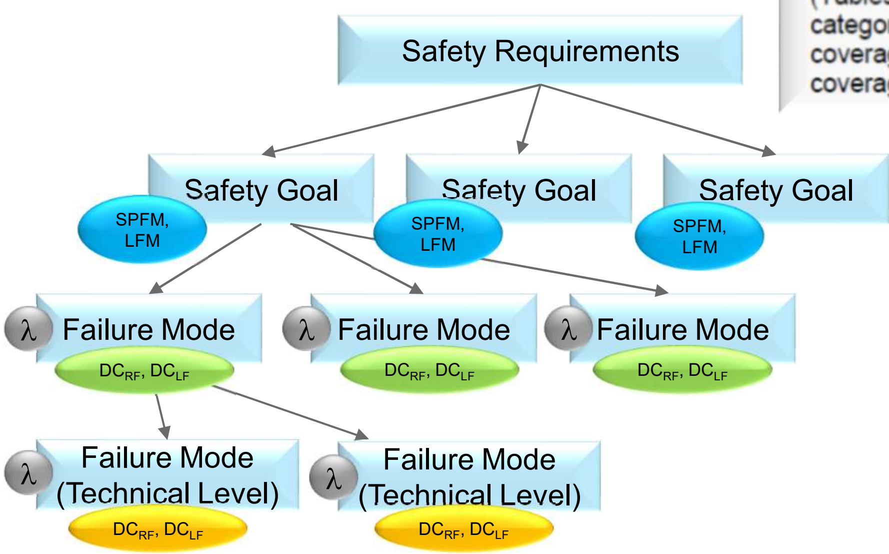
Note: Although correlated, test coverage estimated during BIST insertion is not exactly the DC required by the random failures HW metrics



cādence

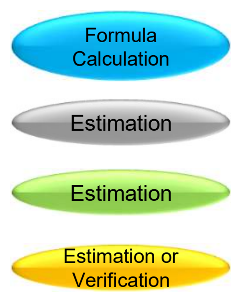
Evaluation of the diagnostic coverage through FS verification

					Total SPFM		94.78%
Part	Sub-part	Failure mode	Safety Goal	Failure Rate	Safety Mechanisms	DC	SPFM
CPU	Decoder	Incorrect Instruction Flow caused by a fault the decode logic	SG1	3.92E-03	SM1	90%	90%
	Multiplier	Incorrect Instruction Execution caused by a fault in the multiplier	SG1	9.09E-03			0%
	Adder	Incorrect Instruction Execution caused by a fault in the adder	SG1	2.25E-03	SM2	90%	90%
	Divider	Incorrect Instruction Execution caused by a fault in the divider	SG1	1.60E-03			0%
	Fetch	Incorrect Instruction Flow caused by a fault the fetch logic	SG1	1.83E-02	SM3	60%	60%
	Cache	Wrong data cell caused by a fault in the cache	SG1	3.98E-01	SM4	99%	99%



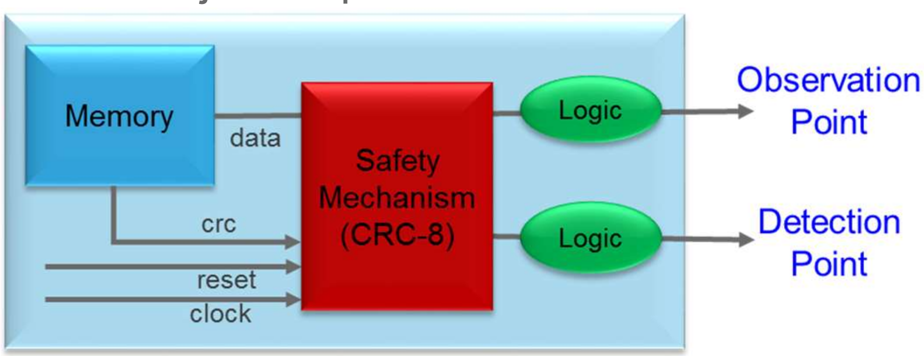
Additional detail on the safety mechanisms associated with these element faults are referenced in each row (Tables D.2 to D.14). The effectiveness of these typical safety mechanisms for the given elements is categorized according to their ability to cover the listed faults to achieve low, medium or high diagnostic coverage of the element. These low, medium and high diagnostic coverage rankings correspond to typical coverage levels at 60 %, 90 % or 99 % respectively.

Excerpt from ISO 26262-5:2011(E) – Annex D (Evaluation of Diagnostic Coverage)



Functional Safety Verification

- For some safety mechanisms (SM), DC can be analytically calculated but might still need to be verified for ASIL D applications
- In the case of custom or SW SM, fault injection simulation can be used for a more accurate verification of the DC value
- A fault injection campaign requires:
  - Description of the workload
  - Observation and detection points
  - Injection points



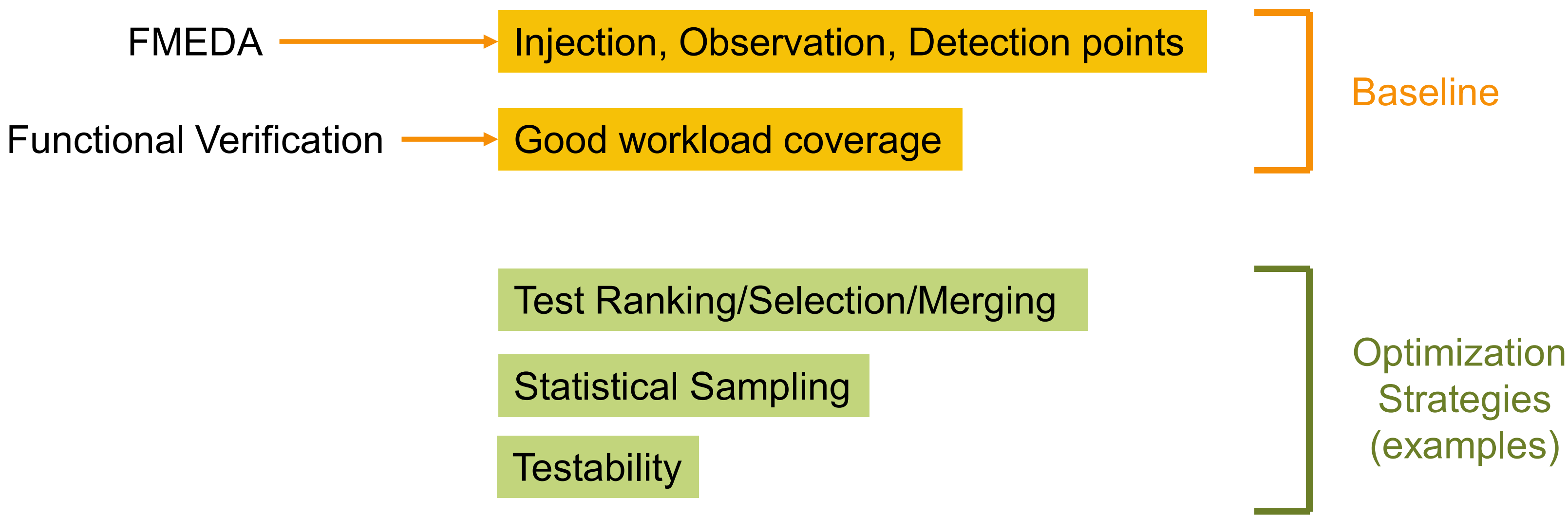
Fault categorization (used to measure the DC):
- Safe: the functional output is not affected by the injected fault (*)
- Dangerous Detected: functional output is affected, but the SM has detected it
- Dangerous Undetected: functional output is affected, and the SM has not detected it

(*) Assuming good coverage from workload

cadence

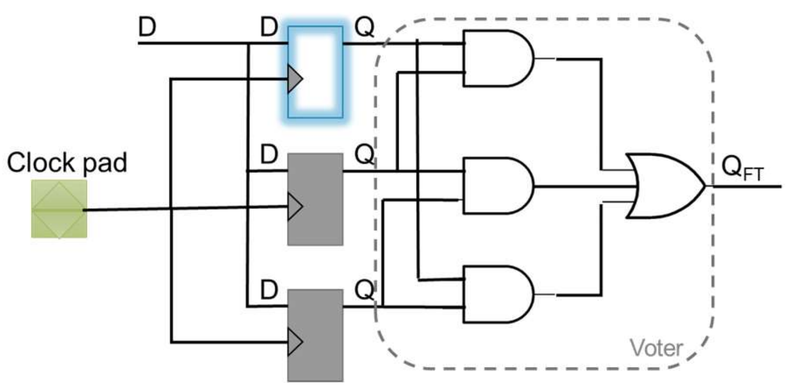
Additional Considerations for FS Verification

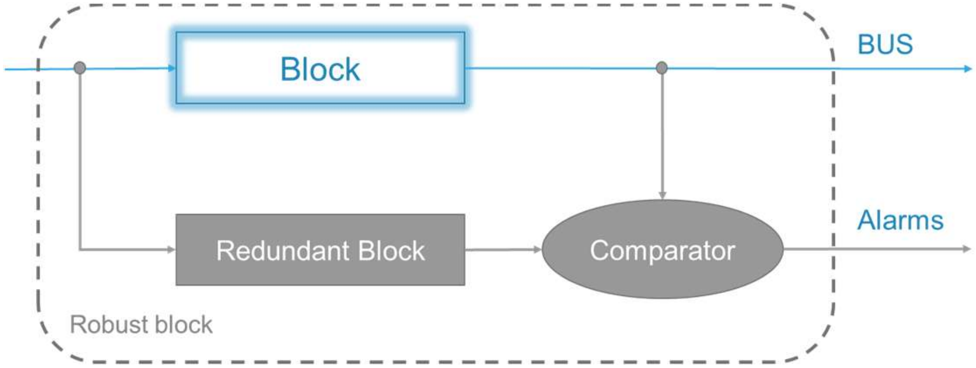
- Fault Injection simulation can be an expensive step and requires optimized setup



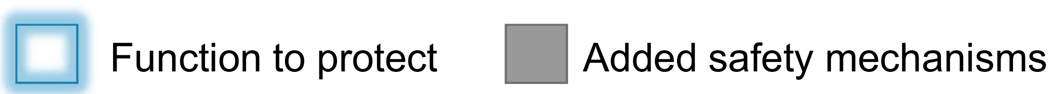
Examples of Safety Mechanisms

Triple Modular Redundancy (TMR)





Dual Core Lock Step (DCLS)



Physical Implementation of Safety Mechanisms

Dependent Failure Analysis requirements

- Safety mechanisms are used to improve FS by increasing the diagnostic coverage (ability to detect a failure and bring the system into a safe state)
- Redundancy only helps when there is true independence of the redundant logic
- Physical Implementation needs to support true independence by avoiding common cause failures

Table D.4 — Processing units

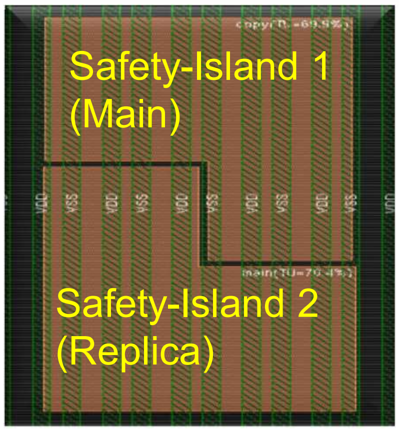
Safety mechanism/measure	See overview of techniques	Typical diagnostic coverage considered achievable	Notes
Self-test by software: limited number of patterns (one channel)	D.2.3.1	Medium	Depends on the quality of the self test
Software diversified redundancy (one hardware channel)	D.2.3.4	High	Depends on the quality of the diversification. Common mode failures can reduce diagnostic coverage
HW redundancy (e.g. Dual Core Lockstep, asymmetric redundancy, coded processing)	D.2.3.6	High	It depends on the quality of redundancy. Common mode failures can reduce diagnostic coverage

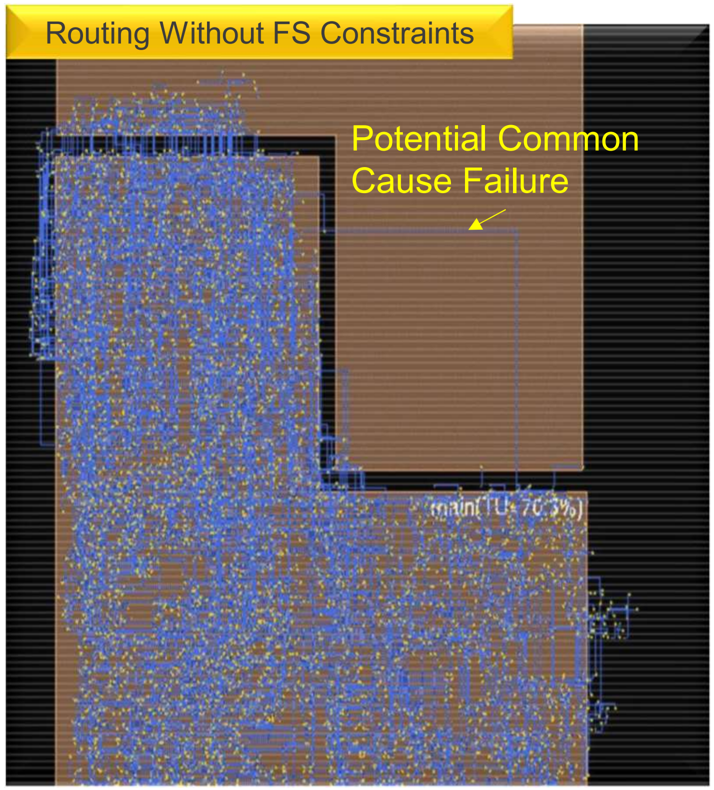
Excerpt from ISO 26262-5:2011(E) – Annex D (Evaluation of Diagnostic Coverage)

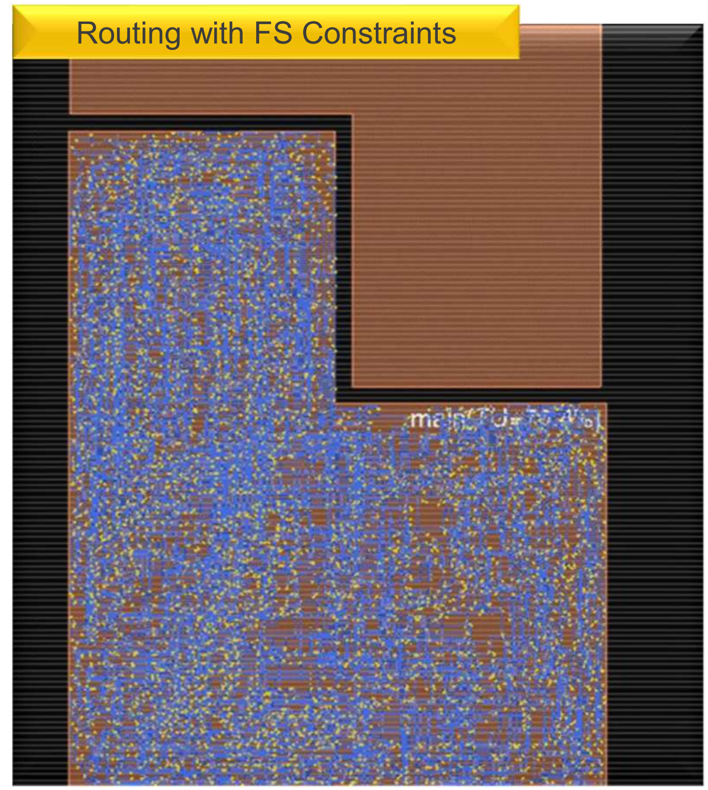
FS-Aware Place&Route

Implementing redundant HW according to DFA requirements

Example of HW redundancy



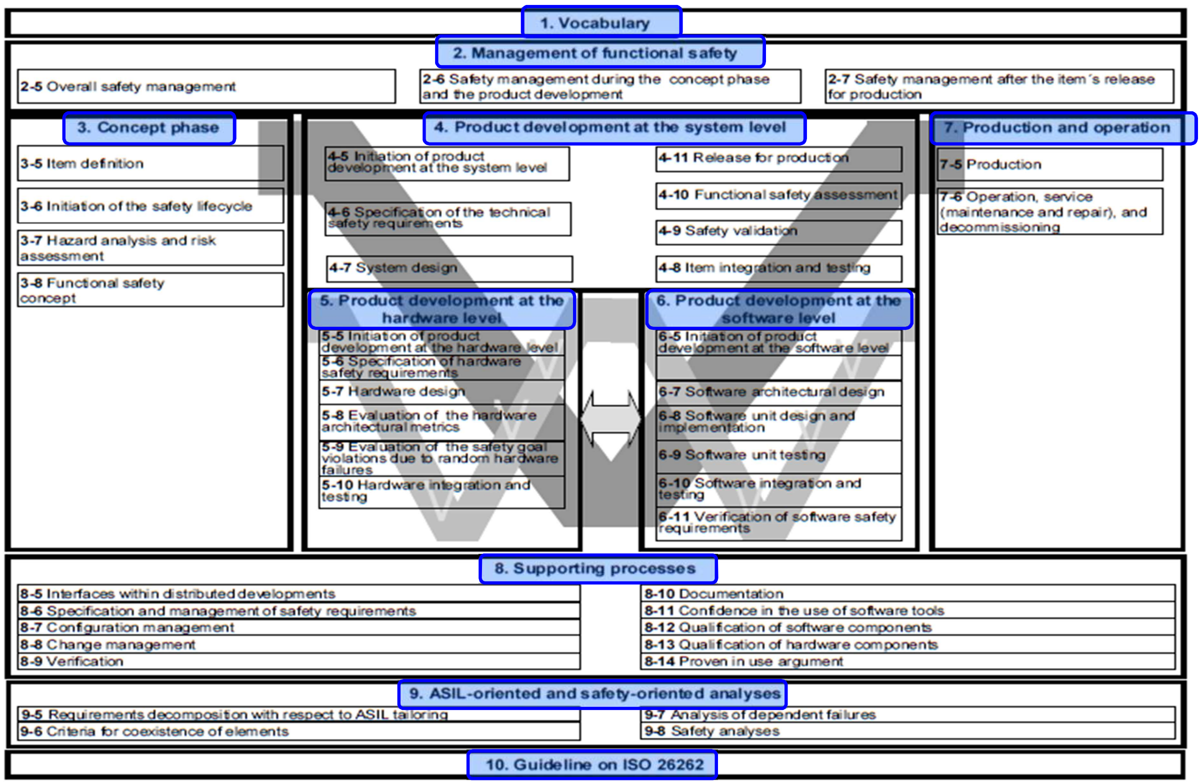




- Same value register spacing – special placement
- Logic isolation - safety islands
- Power-domain routing - specific safety coloring
- Reliability - 100% multi-cut via coverage

Functional Safety Process Compliance

Addressing systematic errors



Both processes and metrics are ASIL-dependent

<u>Part 2 (Management of Functional Safety)</u>
- Functional Safety Manager
- Evidence of competence
- Quality Management

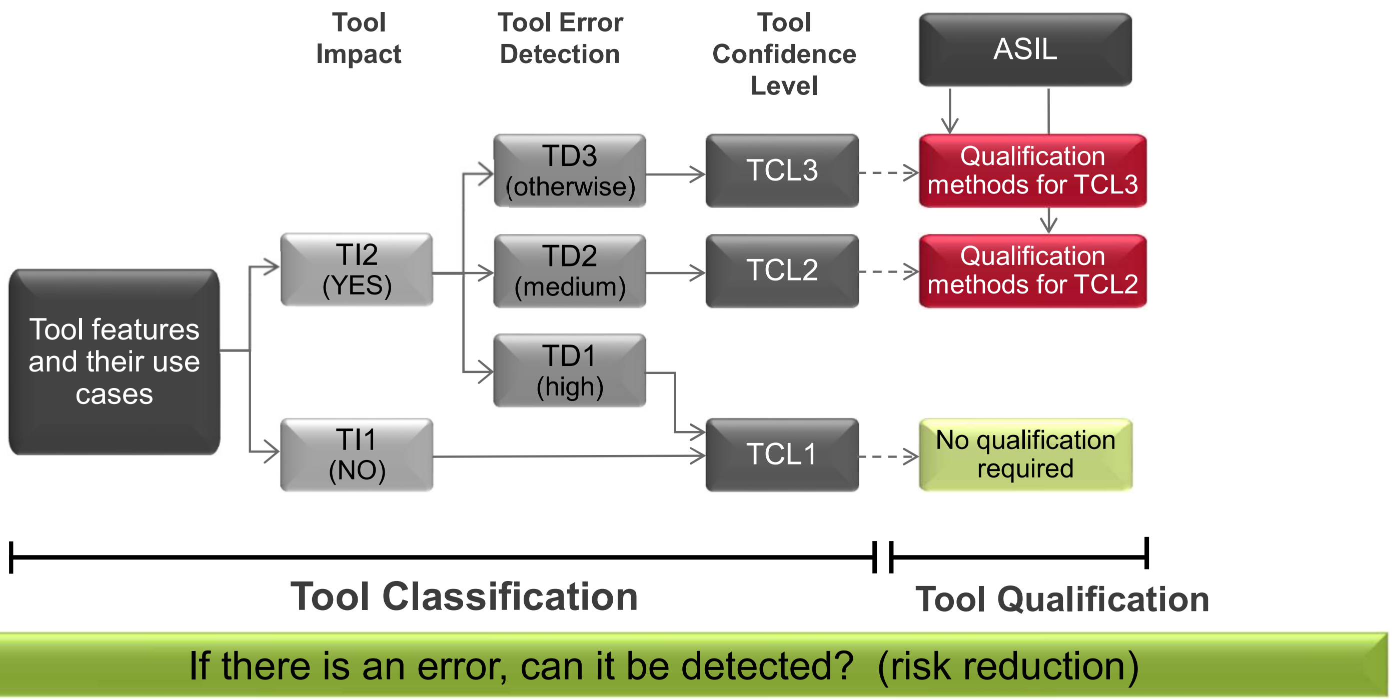
<u>Part 5 (Hardware Development)</u>
- Hardware design and requirement specification
- Hardware integration and testing
- Hardware verification

<u>Part 8 (Supporting Processes)</u>
- Confidence in the use of software tools
- Configuration management
- Change management

cādence®

Tool Confidence Level (TCL) – ISO 26262-8:2011

EDA tools are supporting processes in the development environment

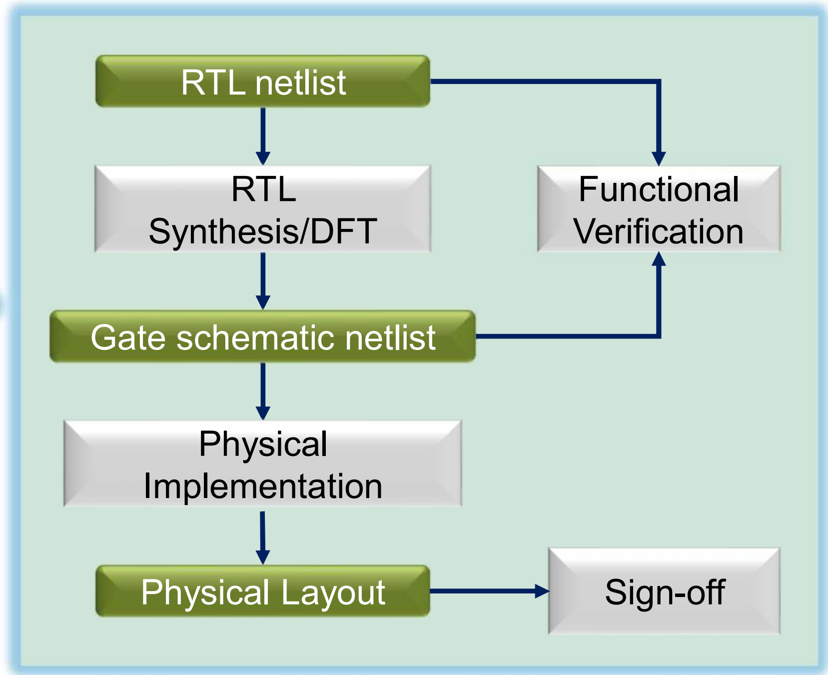


EDA providers deliver Safety Manuals for the tools/flows

Conclusions

- Basics of Functional Safety (Hardware Random Failure Metrics, ASIL)
- Functional Safety Analysis (FMEDA, Timing Analysis, DFA)

- Functional Safety requirements driving the traditional design flow



cadence