Use of Formal Methods for verification and optimization of Fault Lists in the scope of ISO26262

Felipe A. da Silva, Ahmet C. Bagbaba, Said Hamdioui and Christian Sauer

Agenda

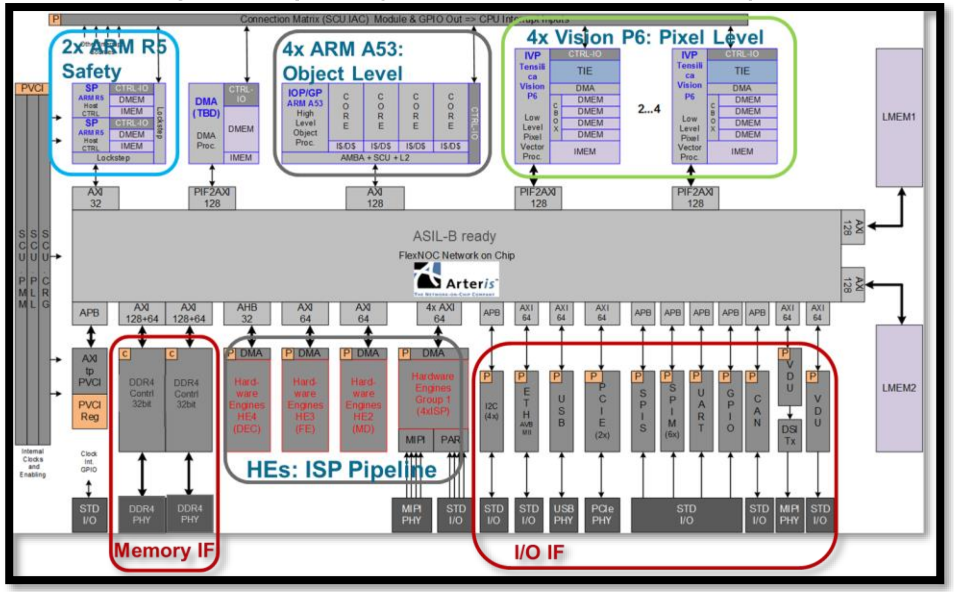
- Introduction
- ISO26262 – Functional Safety Verification and Tool Qualification
- Fault Analysis Methodologies
- Verification and Optimization Flow
- Results
- Conclusions

Introduction

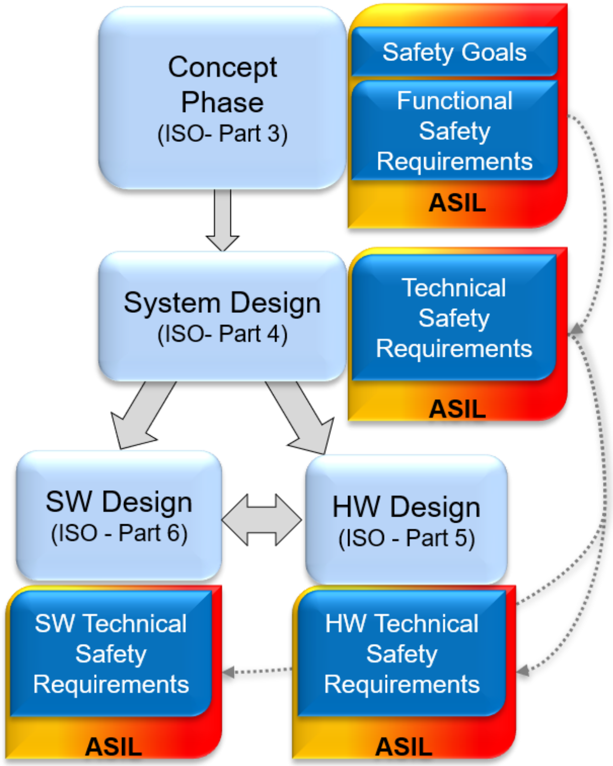
- Autonomous Vehicle

Vehicle capable of sensing its environment and navigating without human input

High complexity SoCs – ADAS Example



Safety Development Process

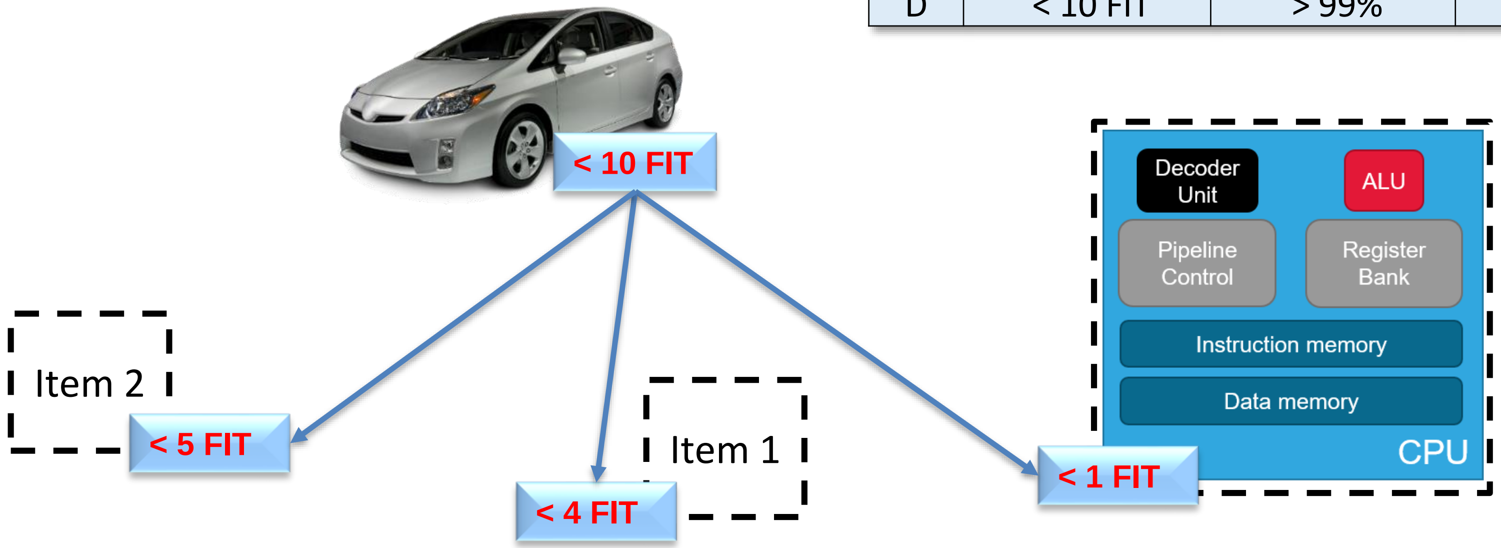


ISO26262 - Functional Safety Verification

- Hazard Analysis and Risk Assessment

- ISO26262 Metrics

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



FIT: Failure In Time (1 Failure / 10^9 hours)
SPFM: Single Point Fault Metric
LFM: Latent Fault Metric
PMHF: Probabilistic Metric for random Hardware Failures

ISO26262 - Tool Qualification

- The development process of the safety-related component shall consider the evaluation of tool outputs

- Tool Impact (TI)
  - tool can introduce or fail to detect errors
- Tool error Detection (TD)
  - confidence in measures to detect tool malfunctions

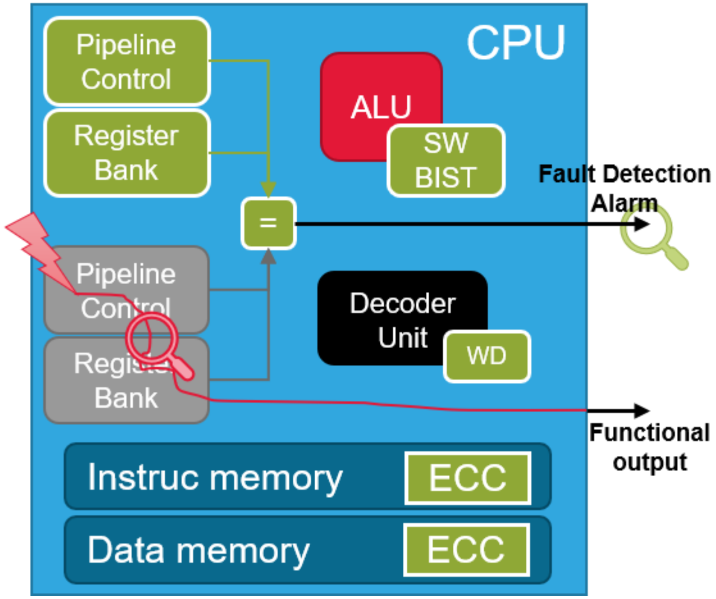
Table 3 — Determination of the tool confidence level (TCL)

		Tool error detection		
		TD1	**TD2**	**TD3**
Tool impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

ISO26262:
Prevention or detection can be accomplished through process steps, redundancy in tasks or software tools or by rationality checks within the software tool itself.

Fault Injection Campaigns

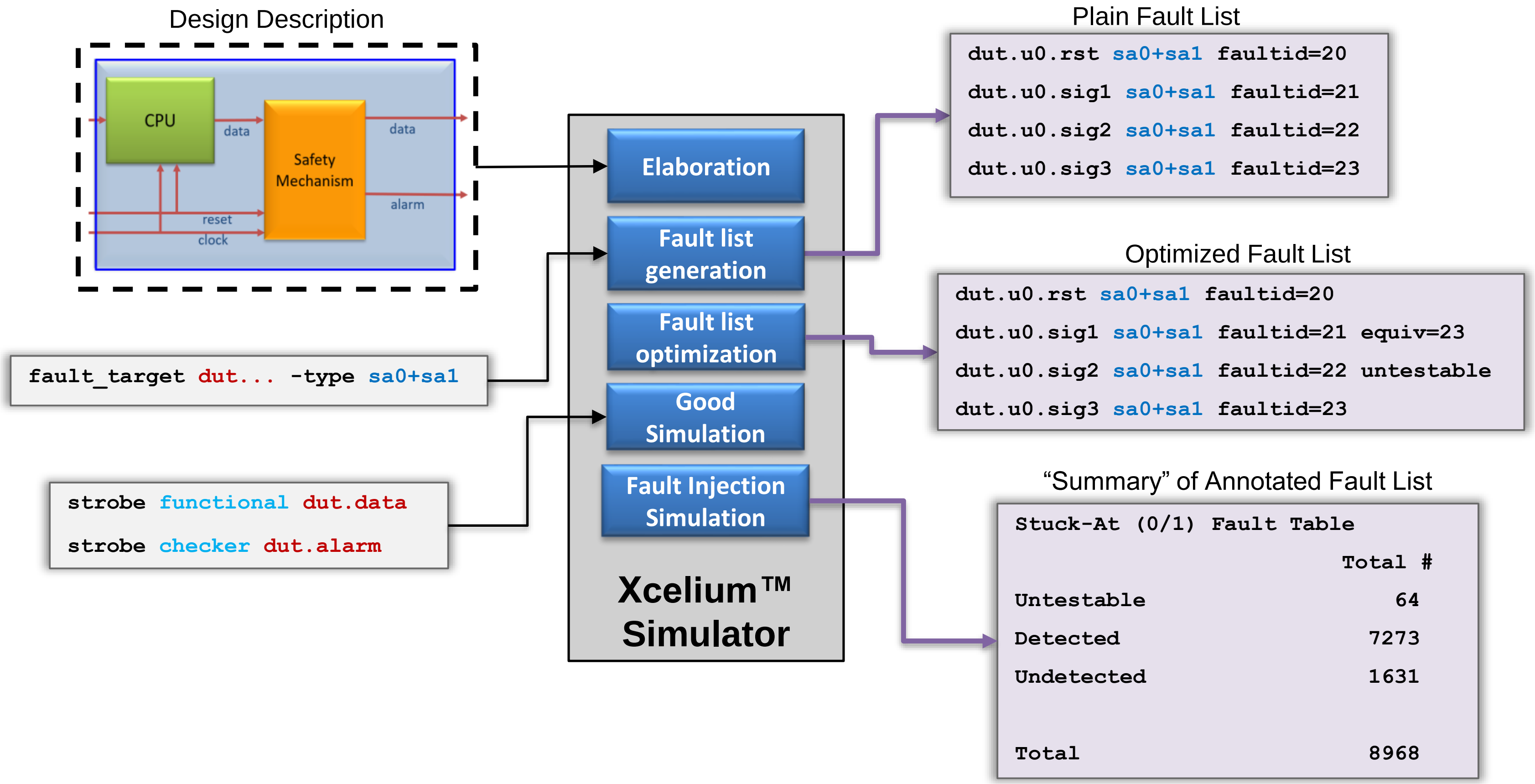
- Behavior analysis of a design under the effect of a random faults
  - Show rationality on Failure Modes (FMs) and Safety Mechanisms (SMs) selection
  - Evaluation of Safety Mechanism capability (Diagnostic Coverage)
  - All possible fault targets should be analyzed



Fault Propagation Analysis

Fault Detection Analysis

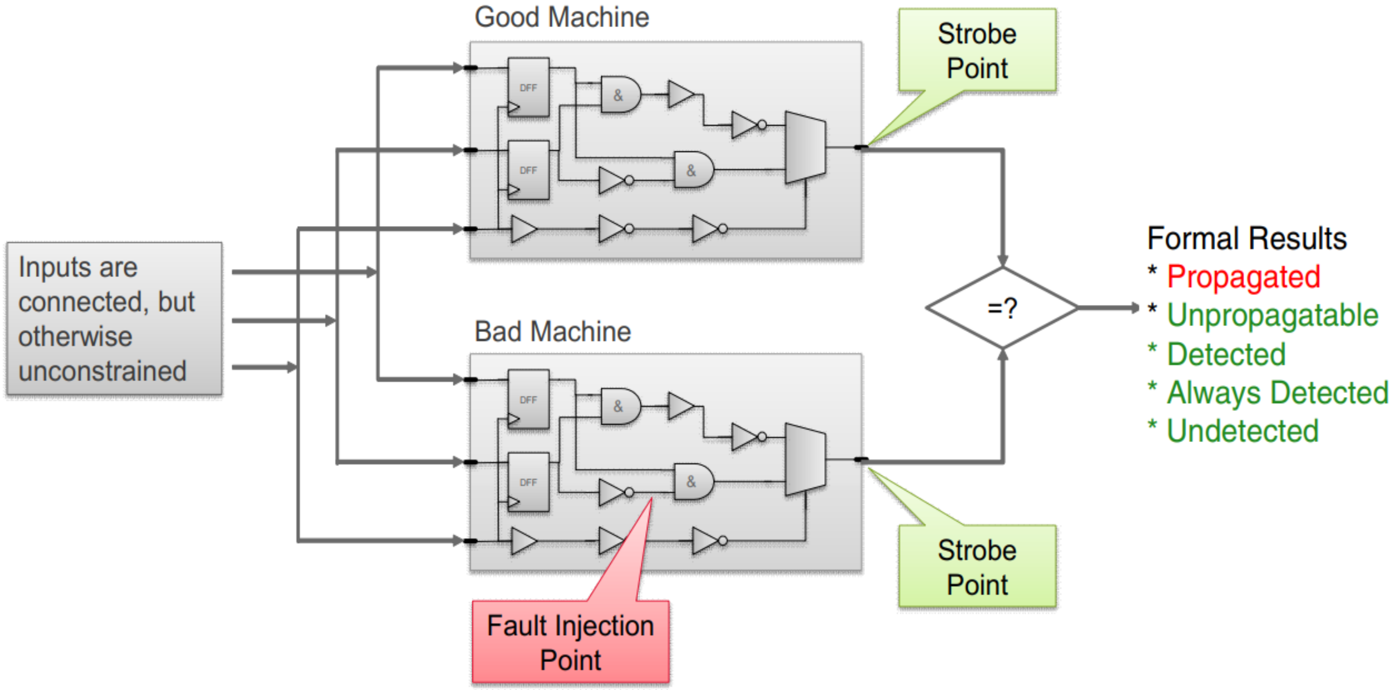
Fault Analysis by Fault Simulator



Fault Analysis by Formal Methods

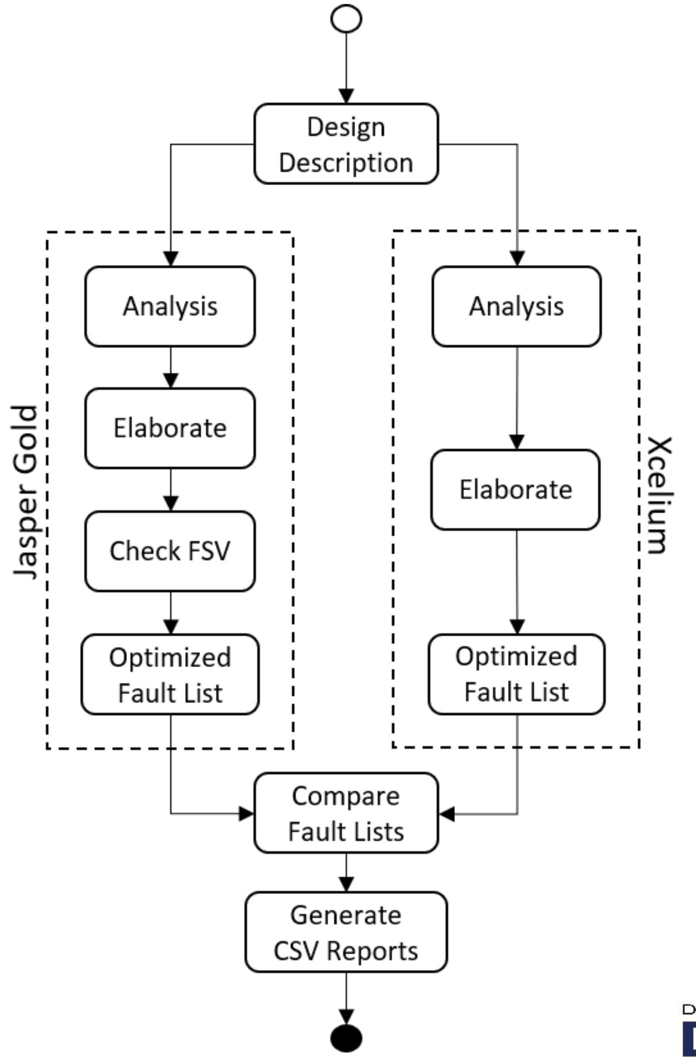
- Standard Analysis
  - Safe (Untestable) Faults
  - Collapsing Groups

- Advanced Analysis
  - Activation
  - Propagation



Verification and Optimization Flow

- Automated application for verification of Fault Lists
- Independent flow execution
- Comparison of Fault Lists
- Detailed Report



Detailed Report

- Compare Rules
  - Example:
    - XFS: Dangerous == JG FSV: Propagated
    - XFS: Detected != JG FSV: Safe

- Report Example:

	XFS				JG FSV				Result
Fault ID	Signal Name	Fault Type	Annotation	Collapsing	Signal Name	Fault Type	Annotation	Collapsing	
0	dut.u0.rst	sa0	Dangerous		dut.u0.rst	SA0	Propagated		PASS
1	dut.u0.rst	sa1	Untestable		dut.u0.rst	SA1	Safe		PASS
2	dut.u0.sig1	sa0	Detected		dut.u0.sig1	SA0	Detected		PASS
3	**dut.u0.sig1**	**sa1**	**Detected**		**dut.u0.sig1**	**SA1**	**Safe**		**WARNING**
4	dut.u0.sig2	sa0	Dangerous	equiv=2	dut.u0.sig2	SA0	Propagated	2	PASS
5	dut.u0.sig2	sa1	Detected		dut.u0.sig2	SA1	Detected		PASS

Example Designs

- International Workshop on Logic and Synthesis (IWLS) 2005 Benchmarks
  - Collection of open designs put together by Cadence Berkeley Labs
  - Verilog RTL design description

- Selected Designs:

DMA	Direct Memory Access (DMA) Controller	vga_lcd	WISHBONE Enhanced VGA/LCD Controller
ss_pcm	Single Slot PCM Interface	tv80	TV80 8-Bit Microprocessor Core
usb_phy	USB 1.1 PHY	systemcaes	SystemC AES
sasc	Simple Asynchronous Serial Controller	mem_ctrl	WISHBONE Memory Controller
simple_spi	MC68HC11E based SPI interface	ac97	WISHBONE AC 97 Controller
i2c	WISHBONE compliant I2C Master controller	usb_funct	USB function core
spi	SPI IP	aes	AES Cipher
systemcdes	SystemC DES	wb_conmax	WISHBONE Conmax IP Core

Results

- Same fault targets for SA0/SA1 were selected by both tools
- Jasper Gold FSV Standard Analysis
  - Average Fault List reduction of 29,5%
  - Average Run Time of 151 seconds

Design	Xcelium		Jasper Gold				Fault List Reduction	
	Nº of Faults	Safe Faults	Nº of Faults	Safe Faults	Collapsed Faults	Run Time (seconds)	By Safe Faults	By Collapsed Faults
DMA	33428	106	33428	4921	8734	186	14,40 %	26,13 %
ac97	11192	134	11192	1401	2326	674	9,88 %	20,78 %
aes	4266	0	4266	49	1408	168	1,15 %	33,01 %
i2c	528	0	528	14	86	9	2,65 %	16,29 %
mem_ctrl	11044	8	11044	3933	2246	346	34,75 %	22,11 %
sasc	86	0	86	1	0	7	1,16 %	0,00 %
simple_spi	534	28	534	35	54	9	1,31 %	10,11 %
spi	1396	0	1396	12	324	13	0,86 %	23,21 %
ss_pcm	242	2	242	3	1	7	0,41 %	0,41 %
systemcaes	9302	0	9302	425	2664	40	3,37 %	47,38 %
systemdes	4104	64	4104	98	1806	41	0,77 %	47,84 %
tv80	1942	36	1942	51	206	49	0,73 %	15,48 %
usb_funct	20386	56	20386	8128	6483	665	39,38 %	32,17 %
usb_phy	364	0	364	3	58	8	0,80 %	18,62 %
vga_lcd	762	0	762	4	0	9	0,52 %	0,00 %
wb_conmax	106666	0	106666	2794	65216	186	2,61 %	61,31 %

Conclusions

- ISO 26262 tool confidence level can be improved by use of redundant methodologies to detect errors in the tool outputs
- Different technologies capable of generating fault lists can be combined
- Formal methods from the JasperGold® platform can work together with the Xcelium™ simulator to create robust and optimized fault injection campaign

Acknowledgment

RESCUE
European Training Network

H2020–MSCA–ITN–2016 / MSCA–ITN–ETN

HORIZON 2020 MARIE CURIE ACTIONS

The RESCUE ETN project has received funding from the European Union's Horizon 2020 Programme under the Marie Curie-Skłodowska actions for research, technological development and demonstration, under grant n. 722325.

Thank You

Any Questions?