Leveraging Several Functional Safety Methodologies (Full Faults and SRF) to Enhance Design Quality in Automotive IC

Gulshan Kumar Sharma
Samsung Semiconductor India Research (SSIR)
India
gks.92@samsung.com

Sougata Bhattacharjee
Samsung Semiconductor India Research (SSIR)
India
sougata.b@samsung.com

Wonil Cho
Samsung Electronics
South Korea
wonil.cho@samsung.com

Akshaya Kumar Jain
Samsung Semiconductor India Research (SSIR)
India
akshaya.jain@samsung.com

James Kim
Siemens
South Korea
james.kim@siemens.com

Sangkyu Park
Samsung Electronics
South Korea
sangq.park@samsung.com

Hyeonuk Noh
Samsung Electronics
South Korea
hyeonuk.noh@samsung.com

Andrey Likhopoy
Samsung Electronics
South Korea
a.likhopoy@samsung.com

Ann Keffer
Siemens
USA
ann.keffer@siemens.com

Arun Gogineni
Siemens
USA
arun.gogineni@siemens.com

***Abstract*- Functional verification efforts are concentrated towards making sure that the design is meeting the expectation as per the specification and all the functionality has been verified. It will never look for design's capability to detect or correct itself from random hardware failures. The ability to recover from hazardous and random failure is very important for functional safety. The motivation for this paper is to introduce functional safety-related flows and observe their affect on design correctness. We also present several comparisons that are derived out of results from using different optimization techniques while performing fault simulation either with full fault list generation or with SRF. The paper also pointed out different techniques so that maximum Diagnostic Coverage (DC) can be achieved in minimum time.**

I. INTRODUCTION

Safety features in today's vehicles are more important than ever before. Newer trends for autonomous vehicles demands car electronics to be extremely safe and more reliable. Functional safety has become a part of the overall safety of the product, which ensures the design behavior. To ensure the quality of chips used in the automotive market, the industry has come up with Industrial (IEC 62380) and Automotive standard (ISO 26262). All automotive chip manufacturers/providers must be compliant with ISO 26262 before being used inside the automobile. The intent of functional safety (FuSa) with simulation or emulation at the SOC or IP level is to inject faults at the safety logic to observe and analyze the effect of those faults.

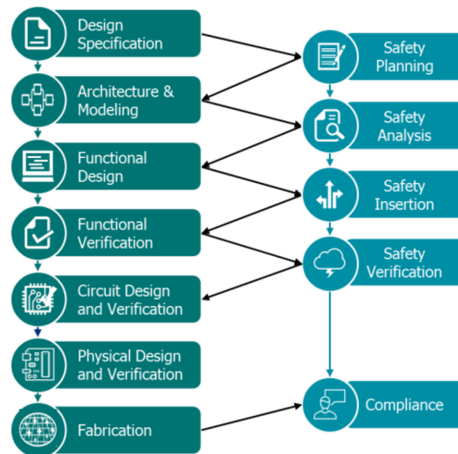

Fig. 1. Traditional flow v/s FuSa flow

The traditional FuSa flow starts with the availability of gate-level netlist. Analyzing the design with safety mechanisms and generating the FMEDA report [1] is the work of a safety engineer. If the diagnostic coverage (DC) is below the required metric, then restarting the whole lifecycle from design to verification and then safety analysis is very time-consuming and will increase the project cost significantly. To reduce these time consuming and expensive iterations, safety engineers should start working at the architecture level. Safety planning and assessment of safety mechanisms used in the design should be considered at an architectural level before the RTL is created. This safety analysis will help in predicting a near to accurate diagnostic coverage (DC) [1][2] value which can help in reaching the desired Automotive Safety Integrity Level (ASIL) of the integrated chip.

Failure rate or Base failure rate (BFR) is the frequency or rate at which a component or device fails and malfunctions. Failure rate is represented by λ and the unit used to measure the failure rate is FIT (Failure in-time). FIT is failure rate of device/component per billion hours. 1 FIT = 10^{-9} per hour.

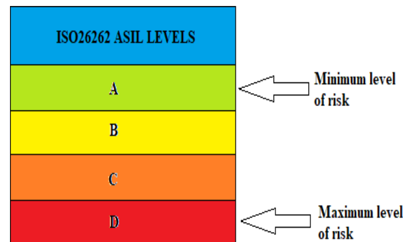

Fig. 2. ASIL Levels

ASIL A corresponds to the minimum level of risk and damage is minimal, whereas, ASIL D corresponds to the maximum level of risk and greater damage to human life.

TABLE I. FAILURE IN TIME (FIT) RATE

Safety Integrity Level	Failure per billion hour (FIT Rate)
A	100,000 to 10,000
B	10,000 to 1,000
C	1,000 to 100
D	100 to 10

The probability of failures per hour is reduced and the risk reduction factor also goes down from ASIL A to ASIL D. The faults expected to occur to achieve ASIL D level is least per billion hours i.e. in the range of 10 to 100 faults.

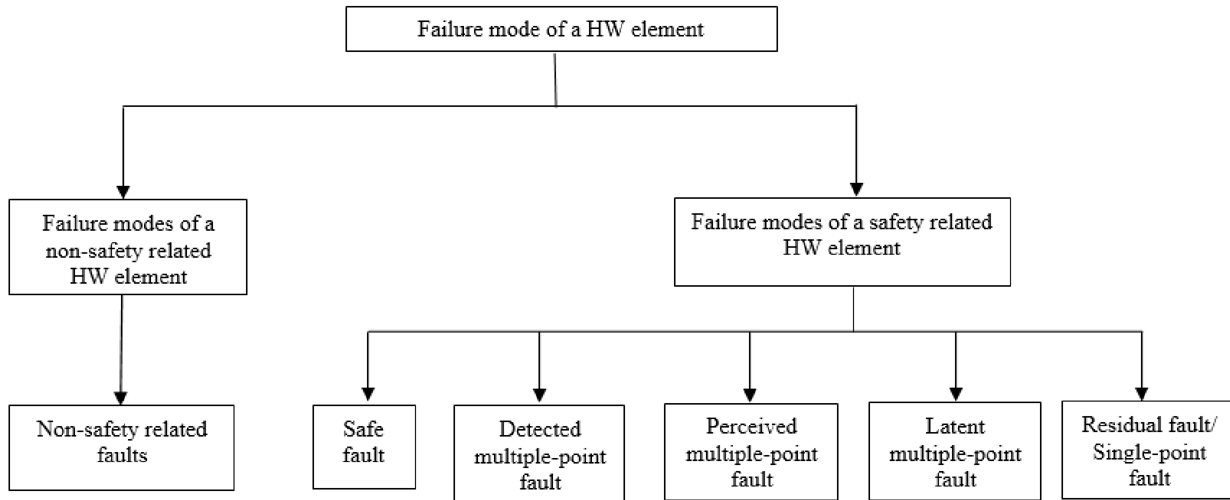

Fig. 3. Failure mode classifications of a hardware element

Fault classification of hardware element is as mentioned below:

- Safe faults are not impacting the safety critical logic and the faults are masked with no effect on output.
- Detected multiple-point fault are the ones that are corrected and detected by safety mechanism.
- Perceived multiple-point faults are faults which are not detected by safety mechanism, but have some noticeable impact on driving experience.
- Latent multiple-point faults are the ones that are corrected but there is no indication that they existed.
- Residual faults are dangerous faults for which safety mechanism is not able to detect the fault.

There is one more way through which the faults are classified and that is with the help of a formal engine embedded within the safety scope called a Cone of Influence (COI). COI is a technique that has been used in the generation point and is observed at checkpoints and between these two points there are several areas where COI overlaps with each other which are helpful in fault classification.

II. SAFETY ANALYSIS FLOW

A. Base Failure Rate Calculation

IEC 62380 standard is commonly used while estimating the Base Failure rate (BFR) of a device or component in functional safety analysis. IEC 62380 IC failure rate can be modelled as sum of die and package failure rates.

$$\lambda = \lambda_{die} + \lambda_{package} \qquad [1]$$

in which

$\lambda_{die} = \lambda_{thermal\ effects} + \lambda_{EOS\ effects}$

and

$\lambda_{package} = \lambda_{thermomechanical\ effects}$

$$\lambda = \left(\underbrace{\left\{\lambda_1 \times N \times e^{-0.35 \times \alpha} + \lambda_2\right\} \times \left\{ \frac{\sum_{i=1}^{y} (\pi_t)_i \times \tau_i}{\tau_{on} + \tau_{off}} \right\}}_{\lambda_{die}} + \left\{ \underbrace{2.75 \times 10^{-3} \times \pi_\alpha \times \left(\sum_{i=1}^{z} (\pi_n)_i \times (\Delta T_i)^{0.68} \right) \times \lambda_3}_{\lambda_{package}} \right\} + \left\{ \underbrace{\pi_I \times \lambda_{EOS}}_{\lambda_{overstress}} \right\} \right) \times 10^{-9}/h$$

Total FIT = Die FIT Package FIT + EOS FIT

Fig. 4. BFR calculation equation from IEC 62380 standard

NECESSARY INFORMATION:

$(t_{ac})_i$	: average outside ambient temperature surrounding the equipment, during the ith phase of the mission profile.
$(t_{ac})_i$	: average ambient temperature of the printed circuit board (PCB) near the components, where the temperature gradient is cancelled.
λ_1	: per transistor base failure rate of the integrated circuit family. See Table 16.
λ_2	: failure rate related to the technology mastering of the integrated circuit. See Table 16.
N	: number of transistors of the integrated circuit.
a	: [(year of manufacturing) – 1998].
$(\pi_t)_i$	: i[th] temperature factor related to the ith junction temperature of the integrated circuit mission profile.
τ_i	: i[th] working time ratio of the integrated circuit for the ith junction temperature of the mission profile.
τ_{on}	: total working time ratio of the integrated circuit. With: $\tau_{on} = \sum_{i=1}^{y} \tau_i$
τ_{off}	: time ratio for the integrated circuit being in storage (or dormant). With $\tau_{on} + \tau_{off} = 1$
π_α	: influence factor related to the thermal expansion coefficients difference, between the mounting substrate and the package material.
$(\pi_n)_i$	: i[th] influence factor related to the annual cycles number of thermal variations seen by the package, with the amplitude ΔT_i .
ΔT_i	: i[th] thermal amplitude variation of the mission profile.
λ_3	: base failure rate of the integrated circuit package. See Table 17a and 17b
π_I	: influence factor related to the use of the integrated circuit (interface or not).
λ_{EOS}	: failure rate related to the electrical overstress in the considered application..

B. Safety Mechanism

Safety mechanism refers to the design and implementation of safety measures in the vehicle hardware to protect occupants and prevent accidents. This can include measures such as sensors and systems to detect and respond to potential hazards, as well as fail-safe systems that can take over control when any malfunction or failure happens. The goal of the safety mechanism is to minimize the risk of accidents and injuries caused by hardware failure.

An example of the implementation of a safety mechanism is shown in Fig. 5. Some examples of safety mechanism implemented in designs include Parity checkers, Error correcting code (ECC), Cyclic redundancy check (CRC), Memory-ECC (MECC), etc.

The details of the Safety Mechanism deployed in the NPU subsystem are given below:

Endpoint ECC (ATD-EECC):

a. ECC is a single bit correct and double bit detect (SECDED)

b. Performs ECC compare among input and output of each Flip Flop and performs following operations

0) Input of each FF is used for ECC generate

1) O/P of each FF is cut and applied to checker

2) ECC checker o/p reconnect the circuit

	EP	SP	Cone	Tolerance
Perm DC	99	0	0	1

Memory ECC (ATD-MECC):

a. ECC is a single bit correct and double bit detect (SECDED)

b. Performs ECC across all the memory instances for SP, EP and Cones.

	EP	SP	Cone	Tolerance
Perm DC	99	99	99	1

Endpoint Parity (ATD-EPAR):

a. Performs parity compare across input and output of each flip flop

b. All the transistors within the endpoints are covered by SM

	EP	SP	Cone	Tolerance
Perm DC	99	0	0	0

Endpoint Cone Duplication (ATD-ECDUP):

a. Performs lockstep compare.

b. All the transistors within the EP and Cones are covered.

	EP	SP	Cone	Tolerance
Perm DC	99	0	99	0

Cyclic Redundancy Check

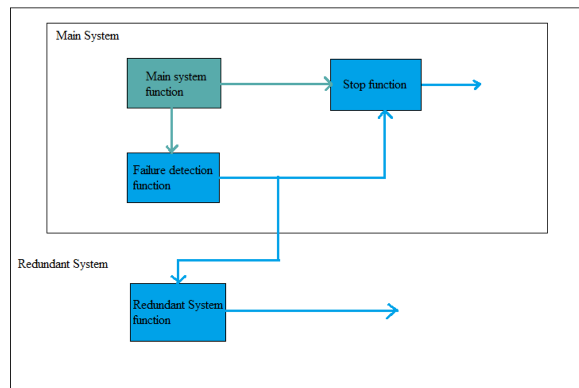

Fig. 5. Example of safety mechanism implementation in design

C. Safety Analysis without Safety Mechanism

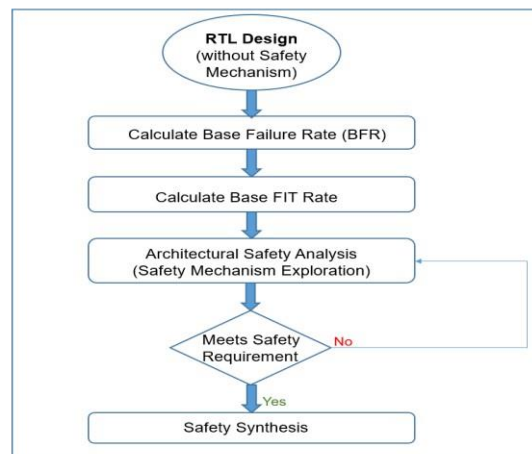

Fig. 6. Safety Analysis flow without a safety mechanism

The first step, for safety analysis of a design is to calculate its Base Failure Rate (BFR). BFR provides the information regarding failure rate per billion hours in the design. BFR will help in predicting the FIT rate and also the diagnostic coverage of the design. Based on the DC value, safety mechanism (SM) exploration is done to find the suitable mechanism that will help in achieving ASIL targets for automotive standards. If the safety requirement is met, then the final step is to go for safety synthesis.

D. Safety Analysis with Safety Mechanism

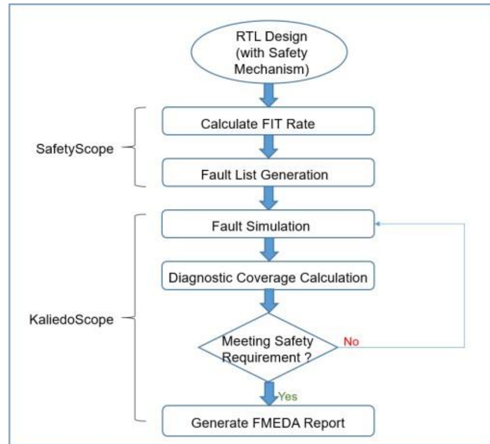

Fig. 7. Safety Analysis flow with the safety mechanism

When a safety mechanism is already added to the design, then the safety flow will start with calculating the FIT rate. After calculating the initial metrics of DC and FIT (λ), a fault list is created for the design nodes. Faults injected can be permanent faults or transient faults. After the Fault simulation, the DC value is calculated which provides information about the fault detection coverage. For achieving ASIL B minimum of 90% of DC value is required and more for ASIL C (> 97%) and ASIL D (> 99%). Till the safety requirements for FIT and DC are met, analyzing the scenarios and creating more scenarios is done to increase the DC value and reduce the FIT rate. Once safety requirements are met, the final functional safety FMEDA report is generated for ISO26262 compliance.

E. Statistical Random Faults (SRF)

Recent increase in the functionality of the automotive semiconductors lead to increase in the number of gates in a logic circuit, the complexity and the number of faults has increased exponentially in nature. Because of the large number of faults, the computational time for full fault campaign also increased multifold. To overcome this high computational requirement, statistical random sampling methods are proposed. SRF [3] contains the subset of actual fault samples in the design on which the fault analysis will be performed. Fault coverage [4] obtained from SRF simulation will be used to estimate the fault coverage of the complete fault list within a small error range. Since the SRF fault list is very small compared to the actual fault list, the overall computational time will be reduced by a great margin.

The equation that has been incorporated for SRF is given below [2]

$$n = \frac{N}{1 + e^2 \times \frac{N-1}{t^2 \times p \times (1-p)}}$$

Also, the SRF is dependent on the level of Confidence Interval of various levels for 90, 95, 99, and 99.9.

For a 99.9% Confidence interval, Statistical Random fault number is achieved by solving below equation

CI = +/- [3.291 * stdev + 1/2n]
Where stdev = sqrt[FPC * c(1 – c) / (n – 1)]

c = coverage goal
n = randomized fault subset
FPC = Finite Population Correction = (1 – n) / N

The above equation holds true for permanent faults and in the case of transient faults "Architecture Vulnerability Factor" is considered for which the following option needs to be employed
--ini avf = true
--ini rand_fault_select = "95:99.9"

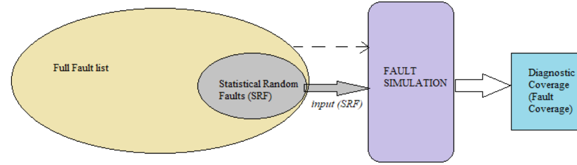

Fig. 8. SRF Fault Simulation

III. FAULT CAMPAIGN

A. Architectural Analysis on Design block under analysis

NPU block is considered for safety analysis because this block contributes for many safety applications related to video sensing like Driver monitoring system (DMS) and Occupant monitoring system (OMS). Safety goal includes safe video sensing application. Safety requirement is NPU should check integrity of NPU and its relevant parts or sub-parts.

Analysis includes breaking the design into sub-parts and analyzing the safety mechanism present for the sub-part. For example, ECC safety mechanism is used for TCM memory block, etc. Based on analysis and providing the information to the tool to generate the FMEA and FMEDA documents which provides the expected DC for the design under analysis.

B. Fault Campaign Flow

Austemper SafetyScope is used during Safety analysis and KaleidoScope is used for fault simulation. Fault campaign implementation steps are as follows:

1. RTL design is given as input to the tool along with the safety mechanism information implemented for each block under observation.
2. Tool generates the FIT values (λ) for both permanent and transient faults analysis.
3. The tool analyzes the design and safety mechanism information to generate a fault list for the block. The fault list generated is an optimized fault list.
4. The generated fault list along with observation points for the faults and alarm list is provided as input to tool for fault simulation. Faults are injected in fault simulation and output of fault simulation is observed.
5. The KaleidoScope will generate the diagnostic coverage (fault coverage) values for the faults.
6. Tool will also perform fault classification and the results can be analyzed to improve the DC coverage.
7. The final step is the generation of the FMEDA report for ISO26262 automotive standard compliance.

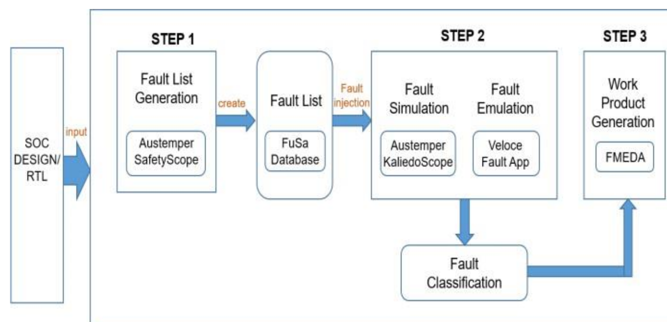

Fig. 9. Fault Campaign.

IV. RESULTS

Results obtained in the analysis are for different memory blocks inside NPU Subsystem block. Memory blocks under analysis were TCM Memory, Shared SRAM, LUTFIFO, DMA Memory. The faults under analysis are single point faults and SPFM (Single point fault metric) is calculated for these faults through fault campaign.
TCM block interacts with the CPU for quick access to certain pre-specified functions. The safety mechanism implemented on this block is MECC (Memory Error Correcting Code), EECC, and Parity. This safety mechanism is highly reliable and correction-capable on memory models. The Diagnostic coverage values for MECC are mentioned in Table II.

TABLE II. DIAGNOSTIC COVERAGE OF MECC SAFETY MECHANISM

Diagnostic Coverage Type		Diagnostic Coverage Value
Permanent	Endpoint	99
	Startpoint	99
	Cone	99
	Tolerance	1

The fault simulation results with full fault list are tabulated in Table III. The total number of faults in full fault list includes both Stuck-at-0 (SA0) and Stuck-at-1 (SA1) faults. For e.g., the SRF faults generated for the TCM block are 4800 faults (2400 SA0 + 2400 SA1 faults) and the full fault list of TCM has 593532 faults (296766 SA0 + 296766 SA1 faults).

Block	Full Fault Space	Alarms Detected
TCM Core 0	593,532	99.58%
SHARED SRAM	37,767,424	Simulation crashed
TCM Core 1	593,532	99.59%
L0 BUF Mem	66,496	99.98%
LUTFIFO	296,104	99.96%
DMA MEM	74,896	99.98%

TABLE III. ALARMS DETECTED WITH FULL FAULT LIST MEMORY SIMULATION

Table II and Table III provides alarms detected results for the different memory blocks with fault simulations. For SHARED SRAM memory block, which has a huge fault list, we observed a crash in simulation after covering around ~10% of the fault list. This is a significantly less number covered for the memory space. The randomized SRF fault list with confidence interval will ensure we are covering fault space and the faults.

The confidence interval (CI) used for generating the SRF fault space is 90%.

Block	SRF Fault Space	Alarms Detected
TCM Core 0	4800	99.92%
SHARED SRAM	4714	100%
TCM Core 1	4804	99.96%
L0 BUF Mem	4708	99.98%
LUTFIFO	4510	99.97%
DMA MEM	4438	99.95%

TABLE IV. ALARMS DETECTED IN SRF FAULT LIST MEMORY SIMULATION

Table V confirms that the computational time required for SRF fault list simulation is greatly reduced with fault coverage prediction accurate with a small error margin.

Block	Full Fault Space	SRF Faults	Full Fault Computation time	SRF Computation time
TCM Core 0	593532	4800	>48hrs	~7hrs
SHARED SRAM	1000000	4714	>72hrs	~6hrs
TCM Core 1	593532	4800	>48hrs	~7hrs
BUF Mem	66496	4708	>24hrs	~5hrs
LUTFIFO	296104	4510	~48hrs	~6hrs
DMA MEM	74896	4438	>24hrs	~5hrs

TABLE V. COMPUTATION TIME REDUCTION IN SRF FOR TCM MEMORY SIMULATION

Figure 10 provides computation time comparison for the full fault list and SRF fault list in fault simulation. The simulation time reduction is significant in terms of numbers and percentage for SRF fault list simulations.

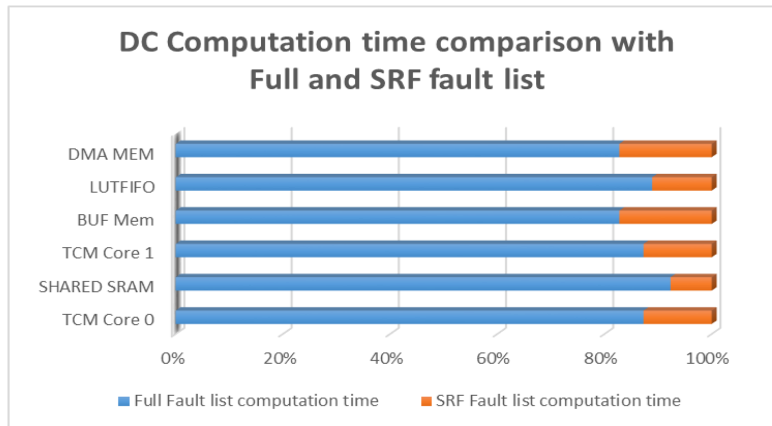

Fig. 10. Computation time comparison

V. CONCLUSION

One of the major motivation to implement SRF fault campaign is the huge fault space (fault list) generated for each block. The limitations occur in fault simulation when the fault space is huge as the time required to inject and cover the faults (both SA0 and SA1) is many times increased. The comparisons between full fault list alarms detected and the SRF fault list alarm detected numbers proves the implementation methodology. SRF provides a significant less computation time when compared with full fault list simulations.

REFERENCES

[1] ISO26262-11:2018, Clause 4.8.2, Page 56), Quantified error and confidence, 2009 Design, Automation & Test in Europe Conference & Exhibition. IEEE. April 2009, 502-506 (Reference of Page 56)

[2] Kevin Rich, Shekhar Mahatme, and Meirav Nitzen "Functional Safety Verification for ISO 26262, DVCON US 2018

[3] Michael G. Mcnamer, Subhash C. Roy and H. Troy Nagle , Fellow, IEEE, "Statistical Fault Sampling", IEEE Transactions on Industrial Electronics, Vol. 36, NO. 2, May 1989.

[4] Agrawal, V.D., "Sampling techniques for determining fault coverage in LSI circuits" , Journal of Digital Systems , vol. 5, no. 3, pp 189 - 202 , 1981.

[5] Austemper KaleidoScope User Guide – Safety Verification

[6] Austemper SafetyScope User Guide – Safety Analysis