Analysis of Fault Types and Common-Cause Faults

<table>
<thead>
<tr>
<th>Document Type</th>
<th>Deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Number</td>
<td>D5.2</td>
</tr>
<tr>
<td>Document Author(s)</td>
<td>David Brès / IFAG</td>
</tr>
<tr>
<td>Document Version / Status</td>
<td>1.3 / Release</td>
</tr>
<tr>
<td>Distribution Level</td>
<td>PU (Public)</td>
</tr>
</tbody>
</table>

Project Acronym       | VeTeSS               |
Project Title         | Verification and Testing to Support Functional Safety Standards |
Project Website       | www.vetess.eu        |
Project Coordinator   | Steve Neill / IFX-UK / steve.neill@infineon.com |
JU Grant Agreement Number | 295311              |
Date of latest version of Annex I against which the assessment will be made | 2012-02-15 |
Analysis of Fault Types and Common-Cause Faults

DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>VeTeSS Consortium</td>
</tr>
<tr>
<td>Georgi Kuzmanov</td>
<td>ARTEMIS-JU</td>
</tr>
</tbody>
</table>

CONTRIBUTORS

<table>
<thead>
<tr>
<th>Organization</th>
<th>Name</th>
<th>Organization</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFAG</td>
<td>David Brès</td>
<td>QRT</td>
<td>Gustav Söderström</td>
</tr>
<tr>
<td>BSC</td>
<td>Jaume Abella</td>
<td>IFAT</td>
<td>Klaus Hörmaier</td>
</tr>
<tr>
<td>BSC</td>
<td>Carles Hernandez</td>
<td>IFAT</td>
<td>Claudia Schett</td>
</tr>
<tr>
<td>NXP-NL</td>
<td>Steef Grimbergen</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DOCUMENT HISTORY

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Author / Organization</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>31.07.2013</td>
<td>David Bres / IFAG</td>
<td>Initial draft</td>
</tr>
<tr>
<td>0.2</td>
<td>09.08.2013</td>
<td>David Bres / IFAG</td>
<td>Refined Section 3.2 content</td>
</tr>
<tr>
<td>0.3</td>
<td>06.09.2013</td>
<td>Carles Hernández /BSC</td>
<td>Literature research content added</td>
</tr>
<tr>
<td>0.4</td>
<td>11.09.2013</td>
<td>Gustav Söderström / QRT</td>
<td>Added Section 3.3 with content</td>
</tr>
<tr>
<td>0.5</td>
<td>02.10.2013</td>
<td>Steef Grimbergen (NXP)</td>
<td>Added section 3.4 Reliability Knowledge Matrix</td>
</tr>
<tr>
<td>0.6</td>
<td>09.10.2013</td>
<td>David Bres / IFAG</td>
<td>Added content from NXP in section 3.4 Added more content in section 3.2 Added list of references Draft for sections Objectives / Dissemination and Exploitation / Conclusion Added link to IFAG MC fault catalogue in iShare</td>
</tr>
<tr>
<td>0.7</td>
<td>22.10.2013</td>
<td>David Bres / IFAG</td>
<td>Included internal review comments Spelling/wording corrections</td>
</tr>
<tr>
<td>1.0</td>
<td>12.11.2013</td>
<td>David Bres / IFAG</td>
<td>Clean-up Document ready for formal review</td>
</tr>
<tr>
<td>1.1</td>
<td>12.11.2013</td>
<td>David Bres / IFAG</td>
<td>Corrections after review Release</td>
</tr>
<tr>
<td>1.3</td>
<td>06.10.2014</td>
<td>J Packer / IFX-UK</td>
<td>Changed to CO (set in error) to PU.</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

1. **EXECUTIVE SUMMARY**
   - 1.1 Relationship to other project activities  
   - 1.2 Document structure  

2. **OBJECTIVES**

3. **DESCRIPTION OF WORK**
   - 3.1 Literature research
     - 3.1.1 Objectives
     - 3.1.2 Working method
     - 3.1.3 Literature sources
     - 3.1.4 Fault classification
     - 3.1.5 Results
     - 3.1.6 References
   - 3.2 Microcontroller fault catalogue
     - 3.2.1 Objectives
     - 3.2.2 Data collection process
     - 3.2.3 Innovative concepts
     - 3.2.4 Structure of the fault catalogue
     - 3.2.5 Results
   - 3.3 Fault model catalogue
     - 3.3.1 Objectives
     - 3.3.2 Working method
     - 3.3.3 Innovative concepts
     - 3.3.4 Fault modelling approach
     - 3.3.5 Structure of the fault catalogue
     - 3.3.6 Results
   - 3.4 Reliability knowledge matrix
     - 3.4.1 Objective
     - 3.4.2 Description
     - 3.4.3 Comparison of the rkm with the other fault catalogues

4. **DISSEMINATION, EXPLOITATION AND STANDARDISATION [ALL]**

5. **CONCLUSIONS [ALL]**

6. **REFERENCES**

A. **ABBREVIATIONS AND DEFINITIONS**
LIST OF FIGURES

Figure 1 Short circuit model, before and after fault injection ......................................................... 17
Figure 2 Open circuit model, before and after fault injection ........................................................... 18
Figure 3 Degraded circuit model, before and after fault injection .................................................... 19
Figure 4 Short circuit model, before and after fault injection ........................................................... 19
Figure 5 Multi-node Short circuit, one possible fault model .............................................................. 21

LIST OF TABLES

Table 1. Permanent faults .................................................................................................................. 8
Table 2. Transient faults .................................................................................................................. 9
Table 3 Source of component data in fault model catalogue ........................................................... 22
Table 4 Fault model description table ............................................................................................ 22
1. EXECUTIVE SUMMARY

This report documents the activities and results of Task 5.2.1 Analysis of fault types, common mode faults and common cause initiators.

The activities described in this document have been carried out between August 2012 and September 2013.

1.1 Relationship to other project activities

The work outputs mentioned in this report are inputs for fault injection activities in task T5.3 Simulation and Modelling and task T5.6 Physical Testing.

1.2 Document structure

This document describes the objectives of task T5.2.1, and the work done by the partners that contributed to the deliverable. In section 3 Description of Work, each partner describes his contribution to the Analysis of Fault Types and Common-Cause Faults. The task activities started with a review of the existing literature (3.1), then fault catalogue were compiled for analog (3.3) and digital (3.2) components separately. Finally the fault identified in the faults catalogues were compared to NXP reliability knowledge matrix (3.4). Section 4 lists the dissemination, exploitation and standardisation activities related to the work described in this document.
2. **OBJECTIVES**

The global objective of the task T5.2.1 is to improve the knowledge about the fault models of hardware elements to allow more efficient hardware verification.

This will support the work in WP5 by providing the necessary information to model and inject faults in simulations at different abstraction levels and help select the adapted validation and testing methods for physical verification.

Moreover, to go further than the current state of the art and the requirements from the ISO 26262 standard, it is an objective of this task to identify potential common cause fault initiators and to analyse their functional impact on hardware.

Based on the identified fault models, some will be selected to be modelled and injected in simulation. For these fault models, the description shall enable us to model the fault at different abstraction levels, and provide information such as the fault probability distribution and the failure rate.

It was the objective to consider, for the above mentioned study, the following type of components: Digital components (such as microcontroller parts), transceiver, and integrated power electronics elements. As QRT focus slightly changed since the writing of the technical annex, QRT contribution was shifted from analysis of faults related to integrated power electronics elements to study ways of modelling typical faults in analogue electronic circuits.
3. **DESCRIPTION OF WORK**

3.1 **Literature research**

BSC made the main contribution to the work presented in this section.

3.1.1 **Objectives**

The goal of this activity was to review existing literature in fault modelling, and to provide a list of faults that are relevant for the current and upcoming manufacturing technologies: technology nodes of 65nm and below are considered.

3.1.2 **Working method**

References have been collected by reviewing the main conferences and journals in the area of fault tolerance and fault modelling in electronic circuits. References were chosen by identifying the most significant papers (the most cited ones) considering at least one reference for any of the relevant faults found in the literature.

3.1.3 **Literature sources**

**Conferences**

- International Conference on Dependable Systems and Networks (DSN)
- International On-Line Testing Symposium (IOLTS)
- IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems (DFT)
- IEEE VLSI Test Symposium (VTS)
- International Conference on Computer-Aided Design (ICCAD)

**Journals**

- IEEE Transactions on Semiconductor Manufacturing (TSM)
- IEEE Transactions on Computer-Aided Design (TCAD)
- IEEE Transactions on Computers (TCOM)

3.1.4 **Fault classification**

Faults have been classified according to their nature to efficiently manage all the relevant references. In particular, faults have been classified into two categories: permanent and transient faults. References of faults caused by aging effects have been included in the permanent faults category as they are not reversible. References related to intermittent faults have been included in the transient faults category.

Faults have also been labelled to distinguish between faults caused by the environment and faults related to defects introduced during the manufacturing process (being visible or not after
production test). This classification and its corresponding tagging, improves the way references are handled, making the process of searching for a given reference easier.

### 3.1.5 Results

The main result of this activity is a list of fault sources that, according to literature, should be considered in the verification and test process of current and future microcontroller designs. Associated to each of the sources, a fault model and its corresponding reference are provided. Table 1 lists references covering permanent faults and Table 1. Permanent faults shows the references collected for transient faults.

<table>
<thead>
<tr>
<th>Fault-origin</th>
<th>Effect</th>
<th>Fault-Model</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device level defects: Contact opens, mask misalignments... Chip level defects: Metallizations open/short, leakage or short between package leads, surface contamination...</td>
<td>Individual signals and pins are assumed to be stuck at a given value</td>
<td>stuck-at</td>
<td>[13]</td>
</tr>
<tr>
<td>NBTI affects PMOS transistors when a negative voltage is applied at the gate causing an increase in the threshold voltage</td>
<td>Variation in the nominal value of Vth causing timing violation and/or delay mismatch</td>
<td>timing-delay</td>
<td>[4]</td>
</tr>
<tr>
<td>Random-dopant-fluctuations caused by extra conducting material being deposited or not etched away between two regions of conducting material or by a defect in the insulating layer allowing two conducting layers to come into contact with each other</td>
<td>short-between two nodes</td>
<td>Resistive bridging</td>
<td>[2][13]</td>
</tr>
<tr>
<td>Electromigration</td>
<td>Voids and hillouts caused by the increased current density</td>
<td>Time dependent modelling: from delay faults to stuck-at open/short</td>
<td>[16][15]</td>
</tr>
<tr>
<td>Hot-Carrier Injection</td>
<td>Electrons trapped into the gate oxide causing Vth shifts</td>
<td></td>
<td>[15]</td>
</tr>
<tr>
<td>Time-dependent-dielectric breakdown (TDDB)</td>
<td>formation of conductive paths in the transistor dielectric gate which may short the anode and cathode</td>
<td></td>
<td>[15]</td>
</tr>
</tbody>
</table>

Table 1. Permanent faults
### Analysis of Fault Types and Common-Cause Faults

<table>
<thead>
<tr>
<th>Fault-origin</th>
<th>Effect</th>
<th>Fault-Model</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha particles generated during radioactive decay of the package</td>
<td>Flip of a bit for both storage elements and combinational logic</td>
<td>Single Event Upset</td>
<td>[1][3][11]</td>
</tr>
<tr>
<td>Neutrons, protons...by cosmic rays penetrating the atmosphere</td>
<td>Flip of several bits for both storage elements and combinational logic</td>
<td>Multi-Event Upset</td>
<td>[3][11]</td>
</tr>
<tr>
<td>Alpha particles generated during radioactive decay of the package</td>
<td>Undesired transition that occurs before the signal settles to its intended value.</td>
<td>glitch</td>
<td>[10]</td>
</tr>
<tr>
<td>Neutrons, protons...by cosmic rays penetrating the atmosphere</td>
<td>Timing violation, delay mismatch</td>
<td>Timing-delay</td>
<td>[10]</td>
</tr>
<tr>
<td>EMI (power supply noise, bit oscillation ...)</td>
<td>Crosstalk: Interference caused due to the cross-coupling capacitance and inductance between interconnects</td>
<td></td>
<td>[12]</td>
</tr>
<tr>
<td>EMI (power supply noise, ...)</td>
<td>Crosstalk: Interference caused due to the cross-coupling capacitance and inductance between interconnects</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random telegraph signal noise in the soft breakdown gate leakage</td>
<td>Erratic bit fluctuations</td>
<td>Bit-flip of several cycles duration</td>
<td>[7]</td>
</tr>
</tbody>
</table>

Table 2. Transient faults

### 3.1.6 References


In this paper the increase in the soft error probability across 65nm and 45nm technology nodes at different supply voltages using the Qcrit based simulation methodology is quantified. Qcrit stand for critical charge, this is the minimum electron charge disturbance needed to change the logic level. The Qcrit for both bit cells and latches decreases by 30% as the designs are scaled from 65nm to 45nm. This decrease is expected to continue with further technology scaling as well. This work shows that in sub-65nm technology nodes with aggressive voltage scaling is equally critical to solve the soft error problems in logic (latches, flip-flops) as it is in SRAMs.
This paper presents three resistive bridging fault models valid for different CMOS technologies. The first model is based on Shockley equations and is valid for conventional but not deep submicron CMOS. The second model is obtained by fitting SPICE data. The third resistive bridging fault model uses Berkeley Predictive Technology Model and BSIM4; it is valid for CMOS technologies with feature sizes of 90nm and below, accurately describing non-trivial electrical behaviour in that technologies.

Embedded static random access memory (SRAM) instances are critical contributors to the overall Soft Error Rate (SER) of the system, requiring a careful consideration of the reliability aspects and adequate sizing of the error mitigation capabilities. While error detecting and correcting codes are widely available and particularly effective against most types of Single Event Effects, Multiple Bit Upsets and progressive error accumulation may defeat the error correction capabilities of standard SECDED (Single error correction double error detection) codes.

Accordingly, the paper presents an overall approach to the structural and functional SER analysis of the memory instances in addition to error mitigation efficiency estimation. Moreover, intrinsic, nominal, SER figures are not a realistic indicator of the memory behavior for a given application. In the paper it is proposed instead, an opportunity window metric, associated to the notion of data lifetime in the memory, as extracted from functional simulations. Lastly, based on the opportunity window figures, targeted and efficient fault simulation campaigns can be prepared to estimate high-level functional failures induced by Single Events. The overall memory SER evaluation aims at assisting the designers to improve the performances of the design and to document the reliability figures of the system.

Negative bias temperature instability (NBTI) in MOSFETs is one of the major reliability concerns in sub-100 nm technologies. So far, studies of NBTI and its impact on circuit performance have assumed an average behaviour of the degradation process. However, in very short channel devices, finite number of Si-H bonds in the channel can induce a statistical random variation of the degradation process. This results in significant random Vt variations in PMOS transistor. The NBTI induced variation depends on operating temperature and the effective stress period for the specific device. In this paper, we analyse the impact of stochastic temporal NBTI variations and propose a compact circuit level Vt model. Using the proposed model, we show how temporal Vt variations can affect the lifetime performance of different circuit topologies including 6T SRAM cell and random combinational logic circuits.
In this paper the effect of technology scaling on the soft error rate (SER) is analysed. This paper also presents a detailed modelling of soft errors in SRAM, latches and combinational circuits. Thanks to this modelling the authors are able to characterize the SER for future technologies. More concretely the authors show that for technologies below 50nm and for processor architectures presenting a deep pipeline the SER rate of combinational logic overcomes the SER rate of SRAM circuits in the Chip. Note also that SRAM circuits are easily protected against soft-errors at low cost by means of ECC. Therefore, the effect of soft-errors in combinational logic circuits is presented as an important challenge to be faced in future CPU designs.

Advances in semiconductor technology have led to impressive performance gains of VLSI circuits, in general, and microprocessors, in particular. However, smaller transistor and interconnect dimensions, lower power voltages, and higher operating frequencies have contributed to increased rates of occurrence of transient and intermittent faults. In this paper we address the impact of deep submicron technology on permanent, transient and intermittent classes of faults, and discuss the main trends in circuit dependability. Two case studies exemplify this analysis. The first one deals with intermittent faults induced by manufacturing residuals. The second case study shows that transients generated by timing violations are capable of silently corrupting data. It is concluded that semiconductor industry is approaching a new stage in the design and manufacturing of VLSI circuits. Fault tolerance features, specific to custom designed computers, have to be integrated into commercial-off-the-shelf (COTS) VLSI systems in the future, in order to preserve data integrity and limit the impact of transient and intermittent faults.

This paper describes the observance of erratic behaviour in minimum voltage SRAM designs. The authors show that erratic bits behaviour gets worse with smaller cell sizes and represents another constraint on the scaling of SRAM cells and on the minimum operational voltage of SRAM arrays.

Memory blocks are important features of any design, in terms of functionality, silicon area and reliability. Embedded SRAM instances are critical contributors to the overall Soft Error Rate of the system, requiring a careful consideration of the reliability aspects and adequate sizing of the error mitigation capabilities. While error detecting and correcting codes are widely available and
particularly effective against most types of Single Event Effects, Multiple Bit Upsets and progressive errors, accumulation may defeat the error correction capabilities of standard SECDED codes. Accordingly, the paper presents an overall approach to the structural and functional SER analysis of the memory instances in addition to error mitigation efficiency estimation. Moreover, intrinsic, nominal SER figures are not a realistic indicator of the memory behaviour for a given application. We propose instead, an opportunity window metric, associated to the notion of data lifetime in the memory, as extracted from functional simulations. Lastly, based on the opportunity window figures, targeted and efficient fault simulation campaigns can be prepared to estimate high-level functional failures induced by Single Events. The overall memory SER evaluation aims at assisting the designers in improving the performances of the design and to document the reliability figures of the system.


In this paper a microarchitecture-aware model for within-die process variations is presented. The model is specified using a small number of highly intuitive parameters. Using the variation model, this paper also proposes a framework to model timing errors caused by parameter variation. The model yields the failure rate of microarchitectural blocks as a function of clock frequency and the amount of variation. The developed tool is capable of producing detailed statistics of timing errors as a function of different process parameters and operating conditions.


This paper addresses the soft-error problem in ultra-deep-submicron circuits by presenting on-line fault-tolerant circuit design techniques. The proposed scheme, separate dual transistor (SDT) structure is introduced into the register design as a key component to increase the input-signal stability as well as the robustness of the circuit against the effects of ionizing particles. The work presented in this paper not only demonstrates the feasibility of its physical implementation, but also shows the cost effectiveness. To compare with other fault-tolerant techniques, ISCAS89 circuits have been synthesized with the SDT standard cells to investigate its cost/timing overheads. Additionally, two main sources of transient errors in UDSM circuits have been modelled; EM noise induced transient errors and particle strike soft errors. The benchmark comparison reveals its better applicability over two representative techniques (TMR and ECC) for the logic circuits in digital systems


Soft error rates measured on embedded SRAMs in a 65nm CMOS technology show a significant increase of the error rate induced by neutron radiation (NSER), while the number of soft errors due to alpha radiation (ASER) is within the expected range. In this paper it is shown that the increase of the NSER values is caused by an unexpected high number of single event upsets (SEU) that flip multiple SRAM cells simultaneously (multi-bit upset). As root cause radiation induced switching of parasitic bipolar transistors was found.

System-on-chips (SOCs) using ultra deep sub-micron (DSM) technologies and GHz clock frequencies have been predicted by the 1997 SIA Road Map. Experiments reported in this paper, show significant crosstalk effects in long on-chip interconnects of GHz DSM chips. Recognizing the importance of high-speed, reliable interconnects in GHz SOCs, we address in this paper the problem of testing for glitch and delay errors caused by crosstalk in buses and interconnects between components of a SOC. Since it is not possible to explicitly test for all the possible process variations and defects that can lead to crosstalk errors in SOC interconnects, we present an abstract model, Maximum Aggressor (MA) fault model, and its test requirements. The attractiveness of the model is that it can abstract crosstalk defects in interconnects with a linear number of faults, while the corresponding MA tests provide complete coverage for all level defects related to cross-coupling capacitance the interconnects. A SPICE-level fault simulation methodology is presented which allows simulation of a small subset of the potentially exponential number of defects. The simulation methodology also enables validation of the proposed fault model and the resulting test set.


The authors present a simulator for resistive-bridging and stuck-at faults. In contrast to earlier work, it is based on electrical equations rather than table look up, thus, exposing more flexibility. For the first time, simulation of sequential circuits is dealt with; interaction of fault effects in current time frame and earlier time frames is elaborated on for different bridge resistances. Experimental results are given for resistive-bridging and stuck-at faults in combinational and sequential circuits. Different definitions of fault coverage are listed, and quantitative results with respect to all these definitions are given for the first time.


As technology feature sizes decrease, single event upset (SEU), digital single event transient (DSET), and multiple bit upset (MBU) effects dominate the radiation response of microcircuits. Recent test circuits and test methods have quantified the pulse widths of DSETs generated from heavy-ion strikes on critical microcircuit nodes. These pulse widths have proven to be much larger than previously thought, which substantiates the importance of DSET induced errors to the soft error rate (SER) of modern microcircuits. New DSET circuit modelling approaches are presented which couple the circuit response to the charge collection mechanisms responsible for forming the DSET. These new circuit charge collection models successfully account for the experimentally observed heavy-ion induced transient widths and are supported by fully-coupled 3-d device physics simulations. Novel hardening and mitigation approaches are proposed based on our new understanding of the circuit response mechanisms. These new techniques are non-invasive to existing fabrication processes and can be transparently applied to existing bulk CMOS microcircuit layouts.

The intrinsic failure mechanisms and reliability models of state-of-the-art MOSFETs are reviewed. Specifically, the contribution of Electromigration, Hot-carrier Injection, Negative-Bias-Temperature Instability, and Time-dependent-dielectric-breakdown in the MTTF is considered. The review includes historical background as well as a new approach for accurately predicting circuit reliability and failure rate from the system point of view.

http://doi.acm.org/10.1145/1667062.1667066

Degradation of devices has become a major issue for processor design due to continuous device shrinkage and current density increase. Transistors and wires suffer high stress, and failures may appear in the field. In particular, wires degrade mainly due to electromigration when driving current. Techniques to mitigate electromigration to some extent have been proposed from the circuit point of view, but much effort is still required from the microarchitecture side to enable wire scaling in future technologies.

3.2 Microcontroller fault catalogue

IFAG made the main contribution to the work presented in this section.

3.2.1 Objectives

The objective of this activity was to compile information on failure modes of a microcontroller hardware unit, allowing the identification of potential common cause initiators. The main focus is on digital parts.

3.2.2 Data collection process

The collection work consisted of several face to face meetings of an expert group composed of:

- A functional safety expert as moderator
- A microcontroller safety architect
- A microcontroller hardware reliability expert
- A radiation expert
- An electromagnetic interferences expert
- An electrostatic discharge expert
- A package reliability expert
3.2.3 Innovative concepts

To meet the objective of identifying the most important potential common cause failure initiators, the expert group defined the following concepts:

**Locality**

The locality indicates to which extent a given root cause can induce a fault. Some causes have very local effects, for example a gate oxide breakdown in a transistor due to electrical overstress. Others, such as energy pulses, may affect a wider area. The locality is indicated as local / mid-range / global.

**On-chip propagation**

The propagation indicates the way a fault due to a given cause may propagate in the hardware, for example through power lines or substrate.

**Locality after propagation**

A fault has an effect which is propagated to the hardware, for example leakage current caused by temperature stress and propagated through a power line, could propagate through the whole component, this is indicated by a locality after propagation “global”. In a similar way to the locality concept described above, the locality after propagation is indicated as local / mid-range / global.

The causes creating faults with a global effects after propagation are very likely common cause initiators and have been considered carefully when we have selected the faults we would be considering for verifying the robustness of our design against common cause faults.

3.2.4 Structure of the fault catalogue

In a first iteration, the following structure was defined for the fault catalogue:

<table>
<thead>
<tr>
<th>Causes</th>
<th>Physical Effect</th>
<th>Physical Effect Type</th>
<th>Consequence</th>
<th>Locality</th>
<th>On-Chip Propagation</th>
<th>Locality after propagation</th>
</tr>
</thead>
</table>

In a second iteration, effects at basic components and/or functional part level have been defined, for each cause:

<table>
<thead>
<tr>
<th>SRAM</th>
<th>Flip-Flops</th>
<th>Logic</th>
<th>Flash</th>
<th>Core voltage / EVR</th>
<th>Clock system</th>
<th>ADC</th>
<th>I/Os</th>
<th>reset system</th>
</tr>
</thead>
</table>

For some causes with a very local effect, the effect on basic structures (SRAM, flip-flops and logic) is more relevant and well known. This is the case for example for soft errors caused by alpha particles of the package or cosmic radiations.

For other causes, affecting the hardware on a larger scale, such as electromagnetic interference, the effect on functional parts is more relevant.

3.2.5 Results

The complete microcontroller fault catalogue can be found on the VeTeSS iShare at:
Based on this activity, IFAG decided to particularly study the effect of high frequency pulses on power and signal lines as it has a high potential for common cause faults between functional parts and their safety mechanism. Particularly it might have a significant influence for lockstep CPUs.

Effects of alpha particles and cosmic neutrons (soft errors) are also of particular interest as soft errors are the dominant type of faults in modern technologies for microcontrollers.

### 3.3 Fault model catalogue

QRT made the main contribution to the work presented in this section.

#### 3.3.1 Objectives

The objectives of the fault model catalogue were to make a complete catalogue to cover as many electronic components as possible and to include a fault model for every fault, of every component type, so that almost any single point fault in a basic electronic system could be set up for further simulation and evaluation.

The component types should have all the data needed for a basic FMEA. This FMEA work should also be aided by an included Fault Model for each of the components, and each of its most common faults. This Fault Model should be described well enough so that a fault injection simulation can be set up for assessing safety goal violations in any system.

#### 3.3.2 Working method

The literature study for the Fault Model Catalogue was based on earlier FMEA work at QRT and the references used in there. Among the references used, a few publicly available resources stand for a majority of the statistics. Among the most useful resources were the “Failure Mode/Mechanism Distributions” report by RAC in 1991 [17], “Siemens Norm SN29500-1” from 2005 [19] and “Military Handbook, Reliability Prediction of Electronic Equipment” by the US Department of Defense in 1990 [18].

It was decided to first use [17] as a base list of components, with their most common faults and their normalized distributions. To add the FIT values needed for a basic FMEA, the list was compared to the component types in [18] and the combined information was compiled. The resulting list was then analysed and a set of general fault models to cover the faults was constructed to complete the Fault Model Catalogue.

#### 3.3.3 Innovative concepts

The main concept and novel idea in this study that motivates a good fault model catalogue is Netlist Fault Injection. This differs from regular software fault injection by actually injecting hardware faults in a virtual copy of the system and then simulating its new, altered behaviour. To simulate faulty or slightly modified hardware systems is, by itself, nothing new. But to combine this with standardized rules for fault modelling and automated simulation programs to directly produce an FMEA with almost complete fault coverage of a system is an approach that the authors of this catalogue have not yet seen anywhere else.
3.3.4 Fault modelling approach

To find a general rule for modelling faults for simulating any common fault of any electronic component, the syntax of describing the functional component for the simulator was used as a starting point. The common way of describing all the components in a system for any simulator is to use netlists. The Netlist syntax may of course vary depending on the simulator, but the information it holds is at some level always the same; every system has components, every component has properties and all components are linked to each other. For a basic component, the information needed for the simulator can be contained in a single line. The line usually contains the component name, which nodes it is attached to, and the properties of the component. If this should be used to model a faulty component, it is reasonable to alter this line, and/or add lines to the Netlist in connection to this component. The resulting fault model does not have to describe the faulty model internal details in order to be used for fault injection; it only has to give rise to a similar response to that of a faulty component. Here it becomes obvious that the most basic and also most common types of faults, namely short circuits and open circuits, also have the most basic kind of fault models. Below is proposed how the three most common faults can be modelled for any two-node components and how this can be transferred to three-node components and even Multi-node components.

It should be noted that some faults are not covered by this approach, namely software faults and faults where all hardware is intact, i.e. bit-flips and signal faults. These faults can be incorporated similarly in the test procedure by altering the software component of the simulated system, but that is out of the scope of this section.

Short, open & degraded components

Short circuited, two-node components are proposed to be modelled by introducing a low-resistive element in parallel with the original model, typically a resistor. The value of the resistor may be as low as possible to simulate an error-prone fault but still preserving a low simulation time, 10 nOhms has been found to be a reasonable value in many cases.

Since short circuits may appear within a short period of time and this, in general, may be seen as a worse case than if introduced over a longer period of time, it is therefore proposed that short circuit models are simulated by first using the original, non-faulty model until a fault-injection time $T_{fi}$, and then simulate the faulty model with the ending state of the first simulation as the initial condition.

![Figure 1 Short circuit model, before and after fault injection](image)

For simulation of fault injection on Netlist level, to model a short-circuited two-node component, one may substitute the following, typical component definition:

```
C_name n1 n2 v1
```
With this:

```
C_name n1 n2 v1
R_name_short n1 n2 10e-9
```

The `C_name` and `R_name_short` are of course arbitrary, only the first letter is usually defined by simulator syntax if there is a resistor, capacitor, inductor, diode etc. that is modelled.

The node numbers `n1` and `n2` are the same in both lines since the short-circuiting element is placed in parallel.

The parameter value `v1` of the component is kept the same since most of the component’s behaviour will be dependent on the newly introduced short-circuiting element. It may, however, be useful to further study if this value can be altered to better simulate a short circuit.

**Open circuited**, two-node components are proposed to be modelled by introducing a highly resistive element in series with the original model, typically a resistor. The value of the resistor may be as high as possible to simulate an error-prone fault, 1 GOhm may be sufficient for most circuits, even higher values may be taken if necessary - the impact on simulation time for extreme values is often not as severe as in the case of modelling short circuits.

Since open circuits may also appear within a short period of time and this, in general, also may be seen as a worse case than if introduced over a longer period of time, it is proposed that open circuit models are simulated just like short circuits, by first using the original non-faulty model until a fault-injection time $T_{fi}$, and then simulate the faulty model with the ending state of the first simulation as the initial condition.

```
Figure 2 Open circuit model, before and after fault injection
```

For simulation of fault injection at Netlist level, to model an open-circuited two-node component it is proposed to substitute the following, typical component definition:

```
C_name n1 n2 v1
```

With this:

```
C_name n1 n3 v1
R_name_open n3 n2 1e9
```

The node number `n3` is here introduced to attach the new element to the node `n2`, and this replaces the old entry of `n2` in the definition of the original component.

**Degraded**, two-node components are proposed to be modelled by simply changing the value of the original model’s property, for example decreasing a resistor’s resistance with 10%.
Since degradation may be assumed to occur over long time and several duty cycles, the fault injection simulation may start in any start-up initial condition and include the fault injected model directly.

![Degraded Circuit Model](image1)

**Figure 3 Degraded circuit model, before and after fault injection**

For simulation of fault injection on netlist level, to model a degraded two-node component it is proposed to substitute the following, typical component definition:

```
R_name n1 n2 v1
```

With this:

```
R_name n1 n2 v1*0.9
```

**Two-node, three-node & multi-node components**

**Two-node components and three-node components** that are Short circuited or Open circuited could in most ways be simulated similarly, however, the three possible nodes that may be short circuited or opened should all be simulated, and in the final evaluation, each result should be weighed in with its corresponding probability.

![Short Circuit Model](image2)

**Figure 4 Short circuit model, before and after fault injection**

For simulation of fault injection at netlist level, to model a short-circuited three-node component it is proposed to substitute the following, typical component definition:

```
Q_name n1 n2 n3 v1 v2 v3
```

With this:

```
Q_name n1 n2 n3 v1 v2 v3
R_name_short n1 n2 10e-9
```
Or:

\[
\begin{align*}
Q_{\text{name}} & n1 \ n2 \ n3 \ v1 \ v2 \ v3 \\
R_{\text{name\_short}} & n1 \ n3 \ 10e^{-9}
\end{align*}
\]

Or:

\[
\begin{align*}
Q_{\text{name}} & n1 \ n2 \ n3 \ v1 \ v2 \ v3 \\
R_{\text{name\_short}} & n2 \ n3 \ 10e^{-9}
\end{align*}
\]

depending on which particular fault is to be modelled. It is however suggested that all three faults are to be simulated since they may give rise to completely different effects within a circuit. If no statistical data is available on the probability distribution within these three, it is suggested that their results are weighted into the final evaluation with one third of the probability of a short circuit.

In the case of an open circuit in a three-node component, the method for deriving the suggested fault model is the same as for short circuits, which is described in more detail in the fault model catalogue.

Degraded two-node components often have a single ideal property such as resistance, capacitance or inductance. With a three-node component or any component model above the most basic, there may be two or more parameters that describe the model. If these parameters change the behaviour in different ways, and they are all subject to degradation, it is reasonable that they all need to be changed and each new degraded component model version that arises from this should be simulated in a separate simulation.

For example, for simulation of fault injection on Netlist level, here with a three-node component, if there are three parameter values \(v1\), \(v2\) and \(v3\) that describe the component it is proposed to substitute the following, definition:

\[
\begin{align*}
Q_{\text{name}} & n1 \ n2 \ n3 \ v1 \ v2 \ v3
\end{align*}
\]

With this:

\[
\begin{align*}
Q_{\text{name}} & n1 \ n2 \ n3 \ v1 \times 0.9 \ v2 \ v3
\end{align*}
\]

Or:

\[
\begin{align*}
Q_{\text{name}} & n1 \ n2 \ n3 \ v1 \ v2 \times 0.9 \ v3
\end{align*}
\]

Or:

\[
\begin{align*}
Q_{\text{name}} & n1 \ n2 \ n3 \ v1 \ v2 \ v3 \times 0.9
\end{align*}
\]

depending on which degradation behaviour is to be modelled.

It is in some cases however possible to argue that, in a worst-case scenario, all three values have degraded, and then the fault model could simply be set to:

\[
\begin{align*}
Q_{\text{name}} & n1 \ n2 \ n3 \ v1 \times 0.9 \ v2 \times 0.9 \ v3 \times 0.9
\end{align*}
\]

But if several fault models are used and simulated, they have to be weighted into the result depending on their distributed probability, which, in lack of statistics, may be set to equal.

It might also be worth mentioning that if the typical ageing behaviour of a components parameter value is to increase, the degradation factor should rather be 1.1 instead of the 0.9 that is suggested above.
**Multi-node components** with *i* inputs and *j* outputs, short circuits could be modelled as with three-node components but there may be up to a total of *i+j* I/O nodes where an open circuit would need to be simulated, and up to (*i+j*)² combinations of nodes to short-circuit, a random selection of combinations can be taken to shorten simulation time but this will of course decrease fault coverage.

![Figure 5 Multi-node Short circuit, one possible fault model](image)

For example, for simulation of fault injection on Netlist level, a Short-circuited, Multi-node component with *i+j* I/O nodes and *k* parameter values, it is proposed to substitute the following, typical component definition:

```
M_name n1 n2 ... ni ni+1 ni+2 ... ni+j v1 v2 ... vk
```

With this:

```
M_name n1 n2 ... ni ni+1 ni+2 ... ni+j v1 v2 ... vi
R_name_short n1 n2 10e-9
```

Or:

```
M_name n1 n2 ... ni ni+1 ni+2 ... ni+j v1 v2 ... vi vi
R_name_short n1 n3 10e-9
```

... and so on until:

```
M_name n1 n2 ... ni ni+1 ni+2 ... ni+j v1 v2 ... vi
R_name_short n1 ni+j 10e-9
```

Or:

```
M_name n1 n2 ... ni ni+1 ni+2 ... ni+j v1 v2 ... vi
R_name_short n2 n3 10e-9
```

... and so on until:

```
M_name n1 n2 ... ni ni+1 ni+2 ... ni+j v1 v2 ... vi
R_name_short ni+j-1 ni+j 10e-9
```

### 3.3.5 Structure of the fault catalogue

The catalogue is divided into four sheets. The general descriptions of fault models reside in the last sheet while the others contain lists of components with their respective failure mode distributions, failure-in-time values and a link to one of the general fault models.

All component types and their sub-categories have their values listed in the same way as Table 3 below and all the fault models have their description ordered according to Table 4.
<table>
<thead>
<tr>
<th>Component type, sub-categories</th>
<th>Fault mode 1</th>
<th>probability 1</th>
<th>(reference for probability value)</th>
<th>FIT-value 1</th>
<th>(reference for FIT value)</th>
<th>Fault model link 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault mode 2</td>
<td>probability 2</td>
<td>(reference for probability value)</td>
<td>FIT-value 2</td>
<td>(reference for FIT value)</td>
<td>Fault model link 2</td>
<td></td>
</tr>
<tr>
<td>... Etc. ...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Fault mode N</td>
<td>probability N</td>
<td>(reference for probability value)</td>
<td>FIT-value N</td>
<td>(reference for FIT value)</td>
<td>Fault model link N</td>
<td></td>
</tr>
</tbody>
</table>

Table 3 Source of component data in fault model catalogue

Table 4 Fault model description table
3.3.6 Results
The catalogue can be found on the VeTeSS iShare at:


3.4 Reliability knowledge matrix
NXP made the main contribution to the work presented in this section.

3.4.1 Objective
The objective of NXP was to compare the information collected in IFAG microcontroller fault catalogue and BSC literature research with NXP reliability knowledge, based on the Reliability Knowledge Matrix.

3.4.2 Description
The Reliability Knowledge Matrix (RKM) intends to provide a concise summary of accessible and acknowledged information on failure mechanisms and failure causes with further details related to those failure mechanisms. It therewith provides a database to support risk assessments and definitions of qualification strategies. It is based on the ZVEI Robustness Validation Knowledge Matrix [20], which is an add-on to JEP122.

The RKM provides a comprehensive overview of reliability related definitions, terms and abbreviations. Also of main interest is the relation between failure mechanisms, causes, modes, effects, mitigation methods etc. For the analysis of Fault types and effects this is of high interest as many failure mechanisms result in quite similar failure modes (like shorts, open, leakage etc). This overview can be used to check which (limited) types of failure modes cover which (rather extended) types of failure mechanisms. This provides a reduction of the number of failures to be modelled.

The NXP internal version of the Reliability Knowledge Matrix (NXP-RKM), is a database of all relevant failure mechanisms that are known within NXP. The NXP-RKM is basically very similar to the ZVEI version. The main deviations are:

- More focus (limited) to NXP specific technologies
- Contains a sheet with ‘delta’s’ to the ZVEI RKM, mainly containing items that are commonly known failures in the industry which are not observed within NXP
- Contains some additional information on observation and detection methods
- Contains some links to tools and rules within NXP for Risk assessment and mitigation

For the analysis of the Fault types these deviations between the ZVEI RKM and the NXP-RKM are regarded as irrelevant.

3.4.3 Comparison of the rkm with the other fault catalogues
Some general observations on the differences between the RKM, the Microcontroller Fault catalogue presented in section 3.2 and the lists presented in section 3.1, resulting from the academic literature study:
The RKM has a wider scope in comparison with the already presented fault models with some additional information, such as:

- Contains package and assembly related faults
- Contains board level related faults (like solder joints)
- Takes faults into account that may lead to non-functional issues (like e.g. increased leakage current leading to battery drain)
- Wider scope of functions (like e.g. non-volatile memories, analogue circuits)

Additional permanent failure modes resulting from the RKM include:

- Increased resistance
- Retention fail of NVM (Non Volatile Memory) → form of bit-flip
- Change (increase as well as decrease) of Vth
- Increase in thermal resistance

Additional transient failure modes resulting from the RKM include:

- Intermittent open or increased resistance
- Intermittent shorts or decreased resistance
4. **DISSEMINATION, EXPLOITATION AND STANDARDISATION [ALL]**

The knowledge collected in this task is intended to be used in other tasks of the work packages (T5.3, T5.6), therefore the main dissemination and exploitation channel is through these tasks.

Nevertheless, IFAG has already started exploiting the work presented in this document in internal activities which are expected to lead to architectural decisions for the future generations of microcontrollers, improving the efficiency and reducing the costs of safety mechanisms.

Moreover, IFAG, is actively participating to ISO 26262 working groups where the knowledge gained from the work summarized in this document could have an impact.

QRT will use the fault catalogues reported here in the tools developed under task T5.3 and aim to use these tools to increase efficiency in the company’s and others daily work with safety analysis and verification of ISO 26262 compliance.

BSC, as a public research institution, will also make use of the material in this deliverable to research on new safety mechanisms adhering to ISO 26262 needs within the umbrella of VeTeSS. Also, BSC will use all non-confidential information (mainly section 3.1) in this deliverable for activities beyond VeTeSS such as other research projects and invited talks.
5. **CONCLUSIONS [ALL]**

The work in this task helps the VeTeSS partners to improve understanding of the faults encountered in electronic components and complex integrated circuits. It will also help to model faults at different levels of abstraction in T5.3 and to select the appropriate stimuli and test methods in T5.6.

When considering the IFAG fault catalogue, it was not possible to provide the required detailed information such as fault distribution and failure rate for all the types of faults. This is mainly due to the way the fault catalogue was built, classifying the faults by root causes instead by fault model. A workaround shall be discussed within T5.3 to allow modelling the fault regardless; reliability handbooks whose data is based on field return analysis are a possible source for such data.

We can also confirm that it makes sense to consider fault models listed in the ISO 26262 standard: stuck-at, bridging, opens and bit flips. On the other hand we foresee difficulties in modelling faults such as bridging and opens; this will definitely be a challenge for T5.3.
6. References


References


[17] Failure Mode/Mechanism Distributions report by RAC in 1991


[19] Siemens Norm SN29500-1 from 2005

# A. Abbreviations and Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSM</td>
<td>Deep submicron</td>
</tr>
<tr>
<td>ECC</td>
<td>Error correction code</td>
</tr>
<tr>
<td>EMI</td>
<td>Electromagnetic interference</td>
</tr>
<tr>
<td>FIT</td>
<td>Failure in time (1 FIT = 1 failure every $10^9$ hours of operation)</td>
</tr>
<tr>
<td>FMEA</td>
<td>Failure mode and effects analysis</td>
</tr>
<tr>
<td>MBU</td>
<td>Multiple-bit upset</td>
</tr>
<tr>
<td>MTTF</td>
<td>Mean time to failure</td>
</tr>
<tr>
<td>NBTI</td>
<td>Negative bias temperature instability</td>
</tr>
<tr>
<td>RKM</td>
<td>Reliability knowledge matrix</td>
</tr>
<tr>
<td>SECDED</td>
<td>Single error correction double error detection</td>
</tr>
<tr>
<td>SER</td>
<td>Soft error rate</td>
</tr>
<tr>
<td>SET</td>
<td>Single event transient</td>
</tr>
<tr>
<td>SEU</td>
<td>Single event upset</td>
</tr>
<tr>
<td>TDDB</td>
<td>Time-dependent dielectric breakdown</td>
</tr>
</tbody>
</table>