Methodology and Algorithms for High-level Modelling of Cosmic Radiation Impacts on Electrical Systems

Hassan Anwar

Director : Claude Thibeault
Co-director : Yvon Savaria

École de Technologie Supérieure

Foreword

What we propose to research:

Algorithms and methodology for CR effects WHAT

aircraft, altitude/latitude of 55,000 ft, high-level fault model, low-level fault behavior

Signature HOW

i.e. Fault emulation, radiation-based experiment

Reliability of an FPGA based systems WHY

more exposure to radiation, phenomena causing faults, mitigation strategy

Foreword

What we propose to research:

Algorithms and methodology for CR effects WHAT

aircraft, altitude/latitude of 55,000 ft, high-level fault model, low-level fault behavior

Signature HOW

i.e. Fault emulation, radiation-based experiment

Reliability of an FPGA based systems WHY

more exposure to radiation, phenomena causing faults, mitigation strategy

Foreword

What we propose to research:

Algorithms and methodology for CR effects WHAT

aircraft, altitude/latitude of 55,000 ft, high-level fault model, low-level fault behavior

Signature HOW

i.e. Fault emulation, radiation-based experiment

Reliability of an FPGA based systems WHY

more exposure to radiation, phenomena causing faults, mitigation strategy

Foreword

What we propose to research:

Algorithms and methodology for CR effects WHAT

aircraft, altitude/latitude of 55,000 ft, high-level fault model, low-level fault behavior

Signature HOW

i.e. Fault emulation, radiation-based experiment

Reliability of an FPGA based systems WHY

more exposure to radiation, phenomena causing faults, mitigation strategy

Table of Contents

Context

Electromagnetic Platform for lightweight Integration/Installation of electrical system in Composite Electrical Aircraft - EPICEA

- Avionics engineering issues, reduction of energy, EM issues, CEA.
- Study the CR effects on aircraft electrical system based on reconfigurable fabric, e.g., FPGAs.
- Extraction of the faulty response (Signature) — Sequential Circuits.

Algorithms and methodology for cosmic radiation effects study on aircrafts

- Model and analyze the faulty behavior of the sequential circuits.
- Develop High-Level fault model.

Problem Statement

Higher Altitude

- Vulnerability of the circuits is due to neutrons at an altitude of 50,000 ft (Xilinx, accessed 2017).
- Soft error occur due to radiation — radiation event causes enough charge disturbance.
- Fault caused by Single event upsets (SEUs) — implications on the behavior of the system.
- Corrupt the underlying functionality of the hardware.
- Fault management strategies.
- High-level fault model for FPGA based circuit.

Research Objectives

Analysis and methodology for modelling the faulty sequential circuits

- The main objective of this thesis — is to develop a methodology for modeling the faulty sequential circuits to model the output of the soft error problem at a behavioral level.
  - **Construct** a high-level model from the faulty response observed at a low-level circuit fault emulation.
  - **Develop** new models that can accurately used to estimate the severity of the faulty behavior from the signature.
  - **Analyze** the fault origination and propagation for faulty sequential circuits.

Research Objectives

Faulty behavior model at a high-level of abstraction

- The objective of this thesis is to develop a faulty behavior model at a high-level of abstraction.
  - The sub-objective of developing a behavioral fault model of a circuit is to generate a **library of faulty components** reusable at high-level of abstraction.
  - Develop a **library of the faulty behavior model** of the sequential circuit components comprising a Simulink model and VHDL entity. So, each time designers need to analyze potential faulty behavior of a circuit at a high-level of abstraction, designers can utilize the faulty components from the library.
  - **VHDL entity** helps to find the device utilization of the target FPGA and it is independent of the technology.

Challenges

Challenges

In order to achieve the above mentioned objectives. The main challenges we foresee are:

- Make a model at higher-level of abstraction from the data extracted at a lower level that represents the behavioral model of the respective signatures. **High-level model that can recognize low-level fault models.**
- Develop an efficient emulation technique, **Xilinx new tools**.
- Develop a relationship between the **bit-flips** and the **fault-model**.

Contribution

- To model the output of the soft error problem at a behavioral level, e.g., high level modelling technique Hidden Markov Model.
- This research thesis proposes a fault behavior model with the modeling techniques, e.g., Markovian-analysis in a novel way (utilize hidden Markov model (HMM) to represent faulty behavior of the sequential circuits).
- Our model will have the capability of efficiently reproduce the signature.The model is used for the signature analysis of the sequential circuit
- Developing relationship between the FPGA bits emulation information to the faulty models.

Background and Related Work

Fundamental concepts and current research

- Radiation Environment.
- Radiation Effects on SRAM based FPGAs.
- Fault Emulation.
- Fault behavioral modelling.

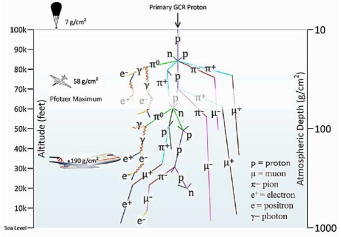
Radiation Environment

Radiation Sources

- Galactic Cosmic rays.
- Radiation from the sun, i.e., solar wind and solar flares.
- Earth's magnetic Field, e.g., magnetosphere and radiation belts.
- Atmospheric shower.

Influence of Radiation

- First observation (1992), bit flipped (Taber and Normand, 1995).
- Aircraft operation of the flight (Tobiska et al., 2015).



Space Radiation

Cosmic Radiation on Aircraft:

- **Cosmic rays - high-energy particles that bombard the earth from outer space - are responsilble for the on-flight computer malfunction. "Something happened in that box that sent the wrong data at various times to the (main) flight computer," ATSB chief commissioner *Martin Dolan* said.**

'Cosmic rays' may have hit Qantas plane off Australia's northwest coast

By Ben Packham, HeraldSun
November 18, 2009 8:01am

'Cosmic rays' may have hit Qantas plane

COSMIC rays may have been responsible for a near disaster involving a Qantas jet off Australia's northwest coast.

Safety investigators have isolated the cause of two terrifying dives by the Airbus A330-303 to an onboard computer.

But the computer itself, fitted to about 900 aircraft worldwide, was found to be in perfect working order, the *Herald Sun* reports.

A flight attendant and 11 passengers were seriously injured and many others experienced minor injuries in a near-miss on October 8 last year.

Radiation Effects on SRAM based FPGAs

Faults caused by cosmic rays in Digital Circuits

- Particle strikes a sensitive node in a semiconductor device.
- Data corruption, transient disturbance.
- Single Event Effects, e,g., SEU.
- Soft error and Hard error.

Fault Injection

Design Verification by Fault Injection

- **Simulation**
  - Simulation based testing is low cost and flexible, but its is difficult to get accurate results.
  - In (Violante et al., 2004) simulation based testing, during the early design phase when hardware is not ready.
  - (Robache et al., 2013) demonstrates how signatures allow building high-level models in MATLAB simulink.
- **Emulation**
  - In (Hobeika et al., 2014) emulation platform for the signature generation, identification of the emulation zone.
  - In (Souari et al., 2016) Fault injection based on the relative sensitivity of the bits.
  - In (DiCarlo et al., 2014) random fault injection.
- **Radiation Testing**
  - Expensive approach but accurate result.
  - In (Hobeika et al., 2014) effects of radiation on a circuit
  - In (Dsilva et al., 2015) Flash based FPGA under neutron beam. FIT for FFs, SRAM cell.

Fault Behavioral Model

Fault Models and Fault Analysis

- Behavioral Domain.
- Transformation Domain.
- Circuit Domain.

Fault Behavioral Model

Behavioral Domain

- Emulation using micro-controller. Model based fault injection (Svenningsson et al., 2010).
  - In **MY THESIS**: Fault emulation is based on the FPGA, independent of the technology, same VHDL could be ported to different FPGAs.
- In (Hayne and Johnson, 1999) Behavioral Fault Mapper. Fault free design and N-faults. Introduced different fault models, e.g., Dead process.
  - VHDL simulator, no hardware real-time fault emulation, code modification.
  - In **MY THESIS**: Investigate further their fault models.
- In (Chen and Jiao, 2017) fault propagation between subsystems.
  - Inconsistency "claim" and "results".
  - System's behavior "Pass" and "Fail".
  - In **MY THESIS**: Signature into their respective FSM.

Fault Behavioral Model

Behavioral Domain

- In (Mirzadeh, 2014) fault behavior model developed with a neural network.
  - Simulation work, no hardware experimental results.
  - Claim to make a library of faulty components.
  - In **MY THESIS**: provide the library of faulty components.
- In (Janschek, 2017) Developed tool to simulate different fault models, e.g., offset, stuck-at-fault, etc.
  - Simulation based high-level fault model.
  - Error Propagation Analysis.
  - In **MY THESIS**: The fault emulation system we describe in this thesis based on the **real-time bit flip information**.

Fault Behavioral Domain

Behavioral Domain

- In (Hobeika et al., 2013) New fault models, adaptive control system. But these fault models were also presented in (Georg, accessed 2017).
- Fault emulation bit information.
  - Xilinx software tool flow **out-dated**.
  - In **MY THESIS**: New vivado tools to perform efficient fault emulation.
- In (Thibeault et al., 2013) based on the C/C++ description of the application.
  - Technique to convert the circuit into control and data flow graph file.
  - Resource estimation tool to find the resources required to implement an application on an FPGA.
  - The work is based on the *.xdl and .ncd* file **out-dated**.
- In **MY THESIS**: we propose to find the faulty FSM and it's respective VHDL entity to compute resource utilization.

Fault Behavioral Model

Transformation Domain

- Transformation of the circuit into respective domain, e,g., BDD (Ubar et al., 2014).
- Boolean Satisfiability Problem (SAT) solvers (Shazli, 2011).
- Mathematical and analytical expressions, fault assumption, assume fault will create an erroneous output.

- **MY THESIS**: has separated itself from the transformation and work solely on low-level for fault emulation and high-level for behavioral modelling.

Fault Behavioral Model

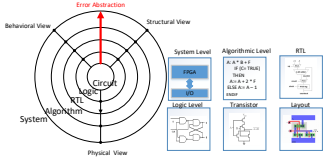
Circuit Domain

- The work done in this domain (Miskov-Zivanov and Marculescu, 2007), (Miskov-Zivanov and Marculescu, 2006) SEUs convergence meaning after how many clock cycles the circuit becomes fault free.
- Insert the fault into the simulator and observe the output.
- Estimate the likelihood of the SET in a sequential circuit, and find the how many clock cycles need to get the SER below the threshold level.
- Keener to find the part of circuit that has the highest error generating probability.
  - Bottleneck to work in this domain fault assumption, glitch size, assign the probabilities for fault propagation.
  - **MY THESIS**: will have the real output error probabilities.

Proposed Approach

Relation to State-of-the-Art

- **In literature:** studied fault behavior at high-level/low-level abstraction under the assumption of fault occurrence.
- **Robache et al. (2013) Hobeika et al. (2014)**.
- Fault models for sequential circuit.
- Fault models in the literature didn't provide any experimental data that these models exist under the radiation or fault emulation.
- Relationship between these fault models with the underlying hardware architecture regarding bit-flips.
- Solve with HMM.



Contribution of this work on Gajski-Kuhn chart.

Proposed Approach

Fault Behavior

- Lost Signal or event fault.
- Stuck Signal.
- Variable change fault.
- Changing the specified range of output.
- Delayed fault, invert fault.
- Short and open circuit fault.
- Stuck-then, Else, Dead Process.
- Micro-operation fault.
- Swap Value Fault, Constant fault, amplification.
- Increase and Decrease.

Proposed Approach

Hidden Markov Model

- The low-level faulty response to the high-level behavioral model.
- HMM, which uses the concept of hidden states (stuck-at-fault, delay fault) and observed states (signatures) to find not only the hidden states of the faulty system but also accurately model the observed states.

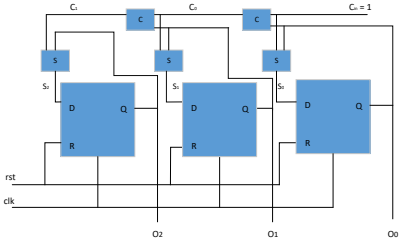
Why HMM ?

- Boolean decision diagram and Algebraic decision diagram.
- Boolean Satisfiability Problem (SAT).
- Monte-Carlo Sampling, approximate approach, symbolic method, simulation.
- **Need to transform the circuit into their respective tool.**
- HMM can directly model the system just looking into the signature values.
- HMM suitable for the systems to model which consists of different observable state (signatures) on different hidden conditions, i.e., fault behavior, e.g., stuck-at-fault, or delay fault.

Modelling Hidden Markov Model

HMM-Example

- To apply the HMM we need a system that generates the probabilistic output pattern, e.g., faulty response of a 3-bit counter.
- The observed sequence is the "signature" and the hidden is the "stuck-at-fault" or any other fault.

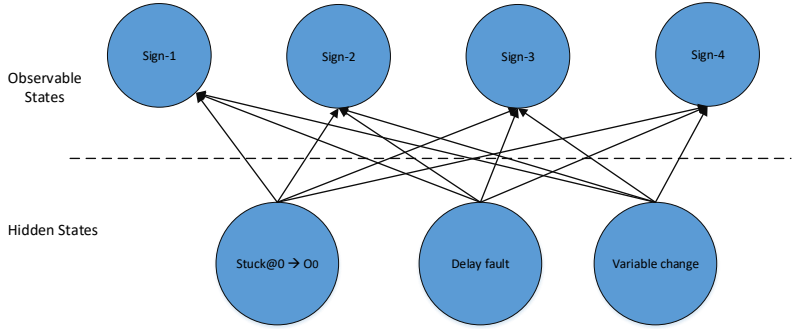


3-bit Counter Signature Generation.

Modelling Hidden Markov Model

HMM-Example

- Four different signature *sign-1, sign-2, sign-3, and sign-3*
- The observed signatures are probabilistically related to the hidden process.
- Signatures: Observable; hidden : faults occur due to bit flip.

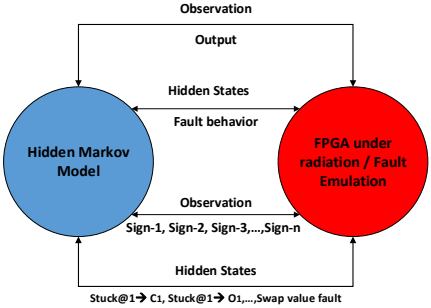


HMM model 3-bit counter.

Modelling Hidden Markov Model

HMM applied to the FPGA based emulation system

- Outputs are emitted by the system.
- Observable — signatures.
- Bits associated with the fault behavior of the system.
- HMM seeks to recover the the states from the observed data.



Hidden Markov Model to the FPGA based fault emulation system.

HMM Application for Signature

- Compute the probability of a given sequence of signatures.
- Compute the most probable sequence of states.
- Given a sequence of observation and learn the best HMM model.

Probability Evaluation

Find the of an observed signature given an HMM.

Decoding

Find the hidden states (fault models) that most probably generated an observed sequence.

Learning

Generate an optimized HMM given a sequence of signatures (observations).

HMM Application for Signature

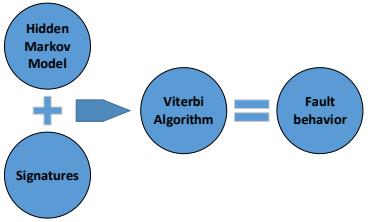
Probability Evaluation

- For probability evaluation, we need to compute the likelihood of an observed signature sequence $O = O_1, O_2, ..., O_t$ given a particular HMM $\Pi = \pi, A, B$. The computation of this probability involves all the possible hidden state sequence and evaluate the corresponding probability.
- This problem can be solved by using the Forward Algorithm.
- This algorithm helps to find the probability of an observe sequence.

HMM Application for Signature

Decoding Application

- The decoding capability of HMM helps to find the sequence of the fault behavior.
- Viterbi algorithm: Most likely sequence.
- Posterior decoding: Most likely state each position.
- Sequence of the hidden states (use in simulator to reproduce the signatures) that give the respective signatures.



Signatures to Fault model.

HMM Application for Signature

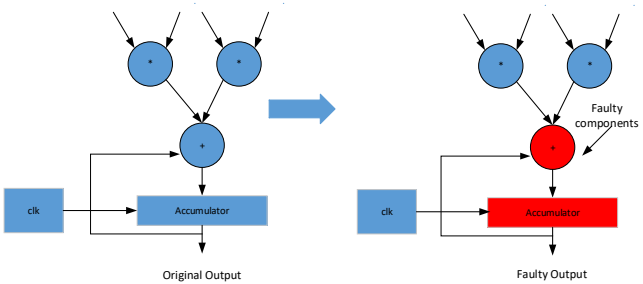
Learning Application

- Optimizing the parameters of the model.
- Supervised learning: mapping function, Unsupervised learning:
- Model parameters and the observations to find the model that fits the data.
- There are three different techniques to do: a) Maximum Likelihood Estimation, b) Viterbi Training, and c) Baum Welch = Forward-Backward Algorithm.
- The Viterbi algorithm only finds the single most likely path, and its corresponding probability. Viterbi algorithm only computes an approximation.
- The Baum-Welch algorithm computes more than this: it does not consider just one path but all possible paths; compute the most likely hidden transition probabilities as well as most likely set of emission probabilities.
- Baum-Welch is more accurate and it would therefore lead to better estimates of the model's parameters

Library Utilization

Library of Faulty Components

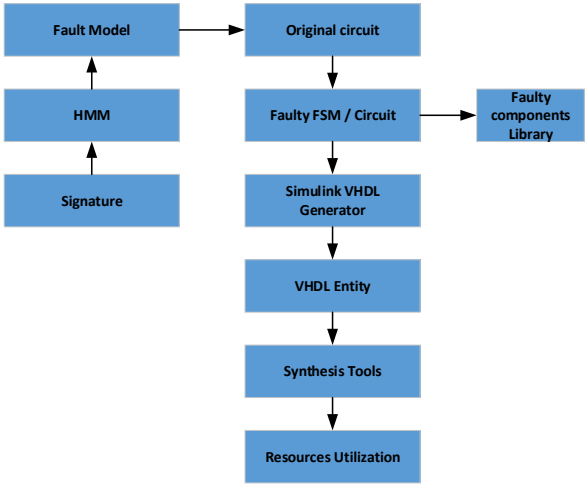
- HMM parameters, probabilities, fault models, hidden states.
- Library of faulty components.
- Designers can use to observe the faulty behavior of each sub circuit of the system.



Library Utilization

System's behavior

- Create a faulty finite state machine.
- N-faulty behavioral models.
- Faulty VHDL entities.
- Fault occurrences and propagation.
- Study hardware faults at high-level of abstraction.



Project Plan

Phase: 01

The emulation platform will be the starting point of the research. We will use the **sensitivity aware bit-flip** technique for the configuration memory upsets. Selection of a suitable benchmark, which is probably ITC'99 used for the testing purpose and signature generation.

Phase: 02

Evaluate the emulation and radiation-based experimental results for signature.

Phase: 03

Implement the above-mentioned methodology and high-level model for soft-error of the sequential circuits, i.e., HMM, Faulty FSM, VHDL entities.

Preliminary Results

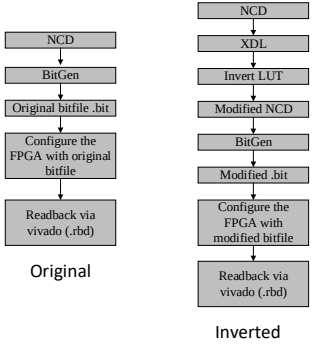
A new Emulation technique based on Bits relative sensitivity

- The purpose of this work is to studying the relative sensitivity ratio difference between the configuration bit set to "1" and those set to "0" to produce the highly **accurate behavior** as expected in the real-time radiation testing environment.
- Prior work: bits at "1" are more likely to generate faults than the bits set at "0" (Souari et al., 2016).
- In this work, we propose a fault injection method by using relative sensitivity values between the **bits features**: e.g., bits belong to LUT at "0", bits belong to LUT at "1", bits belong to Non-LUT at "0", bits belong to Non-LUT at "1".
- Signature computed by this method: 2.13 % percent difference to the adder, and 0.5 % percent difference to the multiplier circuit observed from the radiation based experiment.

Preliminary Results

Methodology

- Experimental Setup (Hobeika et al., 2014).
- Identification of the bits.
- Tools have been developed that extract the bit address and the bit location from the *.rbd* file.



Preliminary Results

Bits Classification Algorithm

Algorithm 1 Bits classification algorithm.

Require: *.rbd original and .rbd inverted*

if $Original = 0$ $Inverted == 0$ **then**
  NL0 = Not LUT bits at 0
else if $Original = 0$ $Inverted == 1$ **then**
  L0 = LUT bits at 0
else if $Original = 1$ $Inverted == 0$ **then**
  L1 = LUT bits at 1
else if $Original = 1$ $Inverted == 1$ **then**
  NL1 = Not LUT bits at 1
end if

Experimental Results

Experimental Results

- Emulation performed by flipping the bits at "1" and at "0."
- Fifty different runs.
- Percentage of zero Signature.

Adder Emulation Tests Comparison.

Test	Zero Signature (%)	Difference with Radiation (%)
Flip@1	49.0	22.5
Flip@0	62.8	2.2
Random	58.1	5.5
SA-01	56.7	7.95
SA-02	60.1	2.13
Radiation	61.4	0

Experimental Results

Experimental Results

Multiplier Emulation Tests Comparison.

Test	Zero Signature (%)	Difference with Radiation (%)
Flip@1	62.0	18.3
Flip@0	76.0	1.9
Random	72.1	3.3
SA-01	73.4	1.5
SA-02	74.9	0.5
Radiation	74.6	0

Experimental Results

Bit Flip Validation

Relative Sensitivity from experimentation performed at TRIUMF.

Bits feature	Relative Sensitivity	Adder (Observed)	Multiplier (Observed)
Bits@0 non LUT	1.00	1.00	1.00
Bits@1 non LUT	1.41	1.43	1.44
Bits@0 LUT	2.06	2.08	2.09
Bits@1 LUT	1.91	1.92	1.91

Signature for High-level Modelling

Signature for High-level Modelling

- Signature Observed at the Triumf Experiment (Hobeika et al., 2014).
- Signature Observed FPGA Emulation.
- High-level Model[1].
- Adder signature fault model.
- Multiplier signature fault model.

[1] Prof. Claude performed experiments on tessent.

Signature for High-level Modelling

Adder signature fault model

- Positive signatures can be modeled as single stuck-at-one fault model.
- The negative signature can be modeled as single as stuck-at-zero fault model.

Adder Signature Format.

Decimal Format	Hexadecimal
16	00000010
64	00000040
128	00000080
-1024	$FFFFFC00$
-512	$FFFFFE00$
-8	$FFFFFFF8$

Signature for High-level Modelling

Adder signature fault model

Rule#01 Example:

Original = 24; *binary equivalent* 11000
Faulty = 8; *binary equivalent* 01000
fourth bit flipped from "1" *to* "0"
signature = 8 - 24 = 8 + (−24) = 16
4^{th} *bit flipped*; $i = 4$, $2^4 = 16$

$$\pm 2^i \quad 0 \leq i \leq 15 \tag{1}$$

Rule#02
Example valid signatures (Adder) for rule # 02:

1010000000000000
1000100000000000
1000100000000000
1000000000000100
1000000000000001

Signature for High-level Modelling

High Level Model of Adder in Simulink

Algorithm 2 Generate a High-level Model of Adder in Simulink

Require: $0 \leq i \leq 15$

Compute

$Original = A + B$

$Faulty = A + B$

Stuck-at-1:

if $Original(i) == 1$ *and* $Faulty(i) == 0$ **then**

$Signature \leftarrow \{+2^i, zeros\{i-1\}\}$

end if Stuck-at-0:

if $Original(i) == 0$ *and* $Faulty(i) == 1$ **then**

$Signature \leftarrow \{\text{-}2^i, zeros\{i-1\}\}$

end if

if $Original - Faulty == MSB = 1$ *and* 1 **then**

$Signature \leftarrow \{Original - Faulty\}$

end if

Signature for High-level Modelling

Multiplier signature fault model

- 8-bit multiplier.
- Two rules have been formulated.
- **Rule # 01:** This rule defines the signature can be expressed by the Equation 1 that represents the stuck-at-1 and stuck-at-0 fault model, and the multiplier behaves as the bit flipped occurred at the output of the multiplier.
- **Rule # 02:** The rule number two defines the format of the signature obtained when a bit flipped cause the multiplier to behave as if when an input bit is either stuck-at-0 or stuck-at-1 can be represented $\{-A \times 2^i \text{ or } -B \times 2^i\}$, $\{+A \times 2^i \text{ or } +B \times 2^i\}$ respectively.

Signature for High-level Modelling

High-level model of multiplier in Simulink

Algorithm 3 Generate a High-level Model of Multiplier in Simulink

Require: $i = 16$ *bit unsigned*

Compute

$Original = A \times B$

$Faulty = A \times B$

Stuck-at-1:

if $Original - Faulty == +2^i$ **then**

$Signature \leftarrow \{+2^i\}$

else if $Original - Faulty == +A \times 2^i$ *or* $+B \times 2^i$ **then**

$Signature \leftarrow \{+\text{A} \times 2^i \text{ or } +\text{B} \times 2^i\}$

end if

Stuck-at-0:

if $Original - Faulty == -2^i$ **then**

$Signature \leftarrow \{-2^i\}$

else if $Original - Faulty == -A \times 2^i$ *or* $-B \times 2^i$ **then**

$Signature \leftarrow \{-\text{A} \times 2^i \text{ or } -\text{B} \times 2^i\}$

end if

Signature for Sequential Circuit

Derived the Signature for:

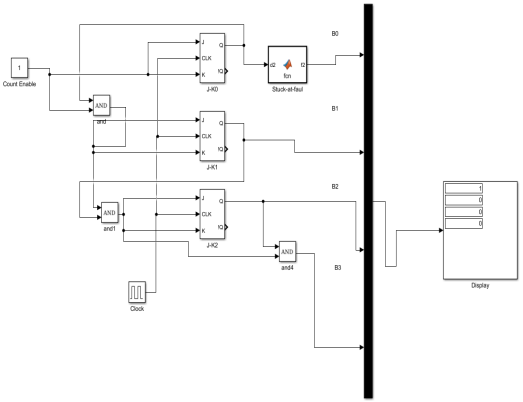
- 3-bit counter
- FIR Filter

Signature for Sequential Circuit

3-bit counter

- 3-bit counter in MATLAB Simulink
- Derived the signatures by stuck-at-f
- Stuck-at-1→ B_0.

Faulty Value (Binary)	Faulty Value	Original Value	Arithmetic Signature
001	1	0	-1
011	3	1	-2
101	5	2	-3
111	7	3	-4
001	1	4	3
011	3	5	2
101	5	6	1
111	7	7	0

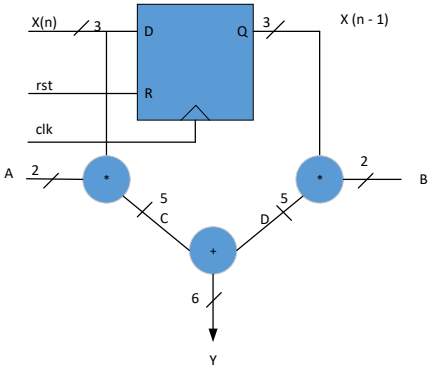


Signature for Sequential Circuit

FIR filter

- Stuck-at-fault model to different nodes in the FIR filter.
- Stuck-at-0$\rightarrow X_1$.

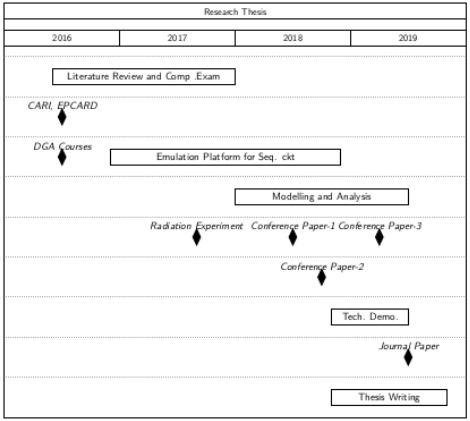
Decimal Golden	Binary	Stuck@0$\rightarrow X_1$	Decimal Faulty	Arithmetic Signature
0	000000	000	0	0
1	000001	001	1	0
6	000110	000	2	4
15	001111	001	3	12
14	001110	100	10	4
27	011011	101	27	0
11	001011	100	9	2
0	000000	101	0	0
0	000000	100	0	0
11	001011	101	9	2
18	010010	100	18	0
21	010101	001	15	6
10	001010	000	2	8
9	001001	001	3	6
1	000001	000	1	0
0	000000	001	0	0



Time Table

Task Development

- **(1) Conference paper** - 2018**:**
  The new fault emulation strategy, bits sensitivity, and it's feature.
- **(2) Conference Paper** - 2018**:**
  High-level fault models for adder and multiplier.
- **(3) Conference Paper** - 2019 **:**
  Fault modelling with HMMs.
  Signature for sequential circuits.
  New emulation technique.
  Fault models.



Timetable.

- **Journal Paper - 2019:**
  Emulation technique, low-level signature generation, Signature comparison, high-level modelling, faulty FSM, VHDL entity, faulty components library.

Thank You!
Questions and Suggestions

References I

Chen, X. and Jiao, J. (2017), A fault propagation modeling method based on a finite state machine, *in* 'Reliability and Maintainability Symposium (RAMS), 2017 Annual', IEEE, pp. 1–7.

DiCarlo, S., Prinetto, P., Rolfo, D. and Trotta, P. (2014), A fault injection methodology and infrastructure for fast single event upsets emulation on xilinx sram-based fpgas, *in* 'Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2014 IEEE International Symposium on', IEEE, pp. 159–164.

Dsilva, D., Wang, J.-J., Rezzak, N. and Jat, N. (2015), Neutron see testing of the 65nm smartfusion2 flash-based fpga, *in* 'Radiation Effects Data Workshop (REDW), 2015 IEEE', IEEE, pp. 1–5.

Georg (accessed 2017), 'Available:https://www.mogentes.eu/public/deliverables/'.

Hayne, R. J. and Johnson, B. W. (1999), Behavioral fault modeling in a vhdl synthesis environment, *in* 'VLSI Test Symposium, 1999. Proceedings. 17th IEEE', IEEE, pp. 333–340.

References II

Hobeika, C., Pichette, S., Ghodbane, A., Thibeault, C., Audet, Y., Boland, J.-F. and Saad, M. (2013), 'Flight control fault models based on seu emulation', *SAE International Journal of Aerospace* **6**(2013-01-2246), 643–649.

Hobeika, C., Pichette, S., Leonard, M., Thibeault, C., Boland, J.-F. and Audet, Y. (2014), Multi-abstraction level signature generation and comparison based on radiation single event upset, *in* 'On-Line Testing Symposium (IOLTS), 2014 IEEE 20th International', IEEE, pp. 212–215.

Janschek, K. (2017), Errorsim: A tool for error propagation analysis of simulink models, *in* 'Computer Safety, Reliability, and Security: 36th International Conference, SAFECOMP 2017, Trento, Italy, September 13-15, 2017, Proceedings', Vol. 10488, Springer, p. 245.

Mirzadeh, Z. (2014), Modeling the faulty behaviour of digital designs using a feed forward neural network based approach, PhD thesis, École de technologie supérieure.

References III

Miskov-Zivanov, N. and Marculescu, D. (2006), Mars-c: modeling and reduction of soft errors in combinational circuits, *in* 'Proceedings of the 43rd annual Design Automation Conference', ACM, pp. 767–772.

Miskov-Zivanov, N. and Marculescu, D. (2007), Soft error rate analysis for sequential circuits, *in* 'Proceedings of the conference on Design, automation and test in Europe', EDA Consortium, pp. 1436–1441.

Robache, R., Boland, J.-F., Thibeault, C. and Savaria, Y. (2013), A methodology for system-level fault injection based on gate-level faulty behavior, *in* 'New Circuits and Systems Conference (NEWCAS), 2013 IEEE 11th International', IEEE, pp. 1–4.

Shazli, S. Z. (2011), High level modeling and mitigation of transient errors in nano-scale systems, PhD thesis, Northeastern University.

Souari, A., Thibeault, C., Blaquière, Y. and Velazco, R. (2016), 'Towards an efficient seu effects emulation on sram-based fpgas', *Microelectronics Reliability* **66**, 173–182.

References IV

Svenningsson, R., Vinter, J., Eriksson, H. and Törngren, M. (2010), *MODIFI: A MODel-Implemented Fault Injection Tool*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 210–222.

Taber, A. H. and Normand, E. (1995), Investigation and characterization of seu effects and hardening strategies in avionics., Technical report, BOEING DEFENSE AND SPACE GROUP SEATTLE WA.

Thibeault, C., Hariri, Y., Hasan, S. R., Hobeika, C., Savaria, Y., Audet, Y. and Tazi, F. Z. (2013), 'A library-based early soft error sensitivity analysis technique for sram-based fpga design', *Journal of Electronic Testing* **29**(4), 457–471.

Tobiska, W. K., Atwell, W., Beck, P., Benton, E., Copeland, K., Dyer, C., Gersey, B., Getley, I., Hands, A., Holland, M., Hong, S., Hwang, J., Jones, B., Malone, K., Meier, M. M., Mertens, C., Phillips, T., Ryden, K., Schwadron, N., Wender, S. A., Wilkins, R. and Xapsos, M. A. (2015), 'Advances in atmospheric radiation measurements and modeling needed to improve air safety', *Space Weather* **13**(4), 202–210. 2015SW001169.
URL: *http://dx.doi.org/10.1002/2015SW001169*

References V

Ubar, R., Marenkov, M., Mironov, D. and Viies, V. (2014), Modeling sequential circuits with shared structurally synthesized bdds, *in* 'Design & Test Symposium (IDT), 2014 9th International', IEEE, pp. 130–135.

Violante, M., Sterpone, L., Ceschia, M., Bortolato, D., Bernardi, P., Reorda, M. S. and Paccagnella, A. (2004), 'Simulation-based analysis of seu effects in sram-based fpgas', *IEEE Transactions on Nuclear science* **51**(6), 3354–3359.

Xilinx (accessed 2017), 'Available: https://www.xilinx.com/xapp1073/'.