CHARACTERIZATION OF SINGLE-EVENT EFFECTS IN COMBINATIONAL LOGIC USING THE C-CREST TECHNIQUE

By

Jonathan Ragnar Ahlbin

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

May, 2009

Nashville, Tennessee

Approved:

Professor Lloyd W. Massengill

Professor Robert A. Reed

ACKNOWLEDGMENTS

I would first like to thank my wife, Holly, who has stood by my side and supported me while working on my research. Along with her, my family and friends have been very supportive of my work and patient with me, especially my parents.

Now I would not have been able to do this research without the support and mentoring of my advisor Prof. Lloyd Massengill. His help and advice have been enlightening and informative as I worked on preparing this work for presentation. The weekly circuit meetings I had with him, Prof. Bhuva, Prof. Holman, Prof. Schrimpf, and Prof. Witulski were very useful in making this work a success.

Two people who deserve many of my thanks are Dr. Jeff Black and his wife Dolores Black. Without their help in designing the C-CREST technique and then writing the VHDL code to make the chip work, this work would never have been possible. They both were excellent sources of information and a joy to talk to about the numerous research issues I came up against.

Sitting in my cubicle in the RER office, I would be amiss not to thank my fellow RER students. They were in the “trenches” with me and understood many of the issues I ran into. They provided a great sounding board and source of ideas to enhance my research. I would like to especially thank Matt, Daniel, Wole, Brian, Sandeepan, Vishwa, Megan, and Sarah for their help with this work.

Lastly, I would like to thank the Defense Threat Reduction Agency (DTRA), Defense Advanced Research Project Agency (DARPA), NASA, and Boeing Phantom Works for sponsoring this work.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS ii

LIST OF TABLES iv

LIST OF FIGURES v

Chapter

I. INTRODUCTION 1

II. SINGLE-EVENT TEST CIRCUITS 8
Common Testing Technique 8
Circuit for Radiation Effects Self-Test 9
Proposed Technique 10

III. COMBINATIONAL CIRCUIT FOR RADIATION EFFECTS SELF-TEST 13

IV. EXPERIMENTAL SETUP 16

V. EXPERIMENTAL RESULTS 21

VI. DISCUSSION 26

VII. CONCLUSION 38

Appendix

A. SAMPLE OF VHDL PROGRAM 40

REFERENCES 45

LIST OF TABLES

Table Page

1. Inverter Designs. 15
2. 10 MeV/nucleon ion sources available at LBNL [22]. 16
3. Test conditions 20

LIST OF FIGURES

Figure Page

1. An illustration showing the generation of electron hole pairs caused by a single event [6] ... 3

2. An illustration showing how a pulse may or may not be latched by a storage element [8] ... 5

3. An illustration showing how a pulse may attenuate the farther it travels down a logic path [12] ... 6

4. A plot that describes the error rate for combinational and sequential logic over frequency [14] ... 7

5. Schematic presenting traditional technique of characterizing SEEs in combinational logic ... 8

6. Schematic presenting basic concept of the original CREST design [17] ... 10

7. Schematic presenting new approach to evaluating the SET impact of combinational logic allowing an increase in combinational logic area without a significant loss in maximum clock speed ... 11

8. Schematic presenting new SET evaluation approach implemented as C-CREST 14

9. Schematic presenting C-CREST test structure using an inverter chain ... 14

10. Layout of the IBM CMOS9SF die with highlighted area showing the combinational logic portion of the C-CREST circuit [21] ... 15

11. Block diagram of the low cost tester [23] ... 17

12. Diagram of the C-CREST test setup [21] ... 18

13. Block diagram of test process [24] ... 19

14. Plot of a basic cross-section curve [12] ... 22

15. Schematic presenting logic states of C-CREST cell before an upset ... 24

16. Schematic illustrating effects of a clock error when using an alternating input pattern [26] ... 25

17. SET cross-section plot depending on LET for the clock frequency of 100.5 MHz 26

18. SET cross-section plot depending on LET for the clock frequency of 201 MHz 27

19. SET cross-section plot depending on LET for the clock frequency of 50.25 MHz 28

20. SET cross-section plot depending on LET for the clock frequency of 12.56 MHz 29

21. Frequency dependence of SET cross-section for the ion neon with an energy of 3.49 MeV-cm^2/mg 30

22. Frequency dependence of SET cross-section for the ion argon with an energy of 9.74 MeV-cm^2/mg 30

23. Frequency dependence of SET cross-section for the ion copper with an energy of 21.0 MeV-cm^2/mg 31

24. Frequency dependence of SET cross-section for the ion krypton with an energy of 30.85 MeV-cm^2/mg 31

25. Frequency dependence of SET cross-section for the ion xenon with an energy of 58.68 MeV-cm^2/mg 32

26. Baseline shift register chain plot showing SET cross-section dependence on LET 33

27. Weak-p inverter shift register chain plot showing SET cross-section dependence on LET 34

28. Matched current inverter shift register chain plot showing SET cross-section dependence on LET 34

29. SET cross-section of the baseline shift register chain with varying clock frequency 35

30. SET cross-section of the weak-p shift register chain with varying clock frequency 36

31. SET cross-section of the matched current shift register chain with varying clock frequency 36

CHAPTER I

INTRODUCTION

Single-event effects (SEE) have been a growing concern for the space, military, and commercial electronic sectors since the 1970's [1]. These concerns have increased as data are collected on modern-day integrated circuits (ICs) showing an increased susceptibility to SEE as feature sizes decrease and frequencies increase [2]**.** A single-event (SE) occurs when a charged particle, such as a heavy-ion, passes through a semiconductor material creating electron-hole pairs (EHPs) along its strike path until it has lost all its energy or left the semiconductor [3]**.** Sensitive junctions within the material, usually reverse-biased p/n junctions, can collect these extra carriers causing harmful effects, such as, erroneous pulses and even complete failure of the device, based on the circuit topology and the amount of charge collected, [3]**.**

As an ion passes through the semiconductor material, it loses energy through Rutherford scattering with the lattice nuclei. The energy is transferred to bound electrons that are ionized in the conduction band. Ions can also lose energy by non-ionizing energy loss (NIEL) where the ion has elastic or inelastic collisions with the elemental atoms of the material. However, ion energy loss due to NIEL is much less than ion energy loss due to direct ionization so this thesis will focus solely on direct ionization.

Direct ionization energy transfer can be quantified by calculating the linear energy transfer (LET) value. LET is defined as the energy loss per unit path length of the particle, it is typically normalized by the density of the material and given in units of MeV-cm^2/mg.

Using the known LET of the ion, the average energy needed to create an EHP for the material, and the density of the material a calculation of charge deposited per unit length can be done. As a reference, in silicon an ion with an LET of 97 MeV-cm^2/mg corresponds to a charge deposition of 1 pC/μm [1].

Many different types of energetic particles can cause a single-event. Four particular energetic particles have been shown to cause SEs: alpha particles, neutrons, protons, and heavy-ions. In the 1970's, it was found that alpha particles present in the packages used on integrated circuits were radioactively decaying and causing soft errors [4], which are changes to the nominal data propagating through the circuit leading to observable errors at the output. Neutron strikes and heavy-ion strikes have also been found to cause SEs that can lead to soft errors [5].

Soft errors can be a troubling problem in a circuit and one type of soft error normally associated with a SE is a voltage transient called a single-event transient (SET) [3]. When a SE occurs, the charge deposited is collected by the reverse-biased junction shown in Fig. 1 [6] and generates a current within the device. This current spike can then lead to a momentary change in the nodal voltage – creating an SET. The pulse width of the SET along with the pulse shape is determined by numerous factors, such as location of the strike, nodal capacitance, nodal resistance, and design of the circuit. Those circuits that have a high amount of drive current, which is defined as the current at the output node that switches the resistive/capacitive load on the output, will recover quicker than circuits with a low amount of drive current. This occurs because the current induced by the SE competes with the original drive current of the circuit. Therefore, those circuits with high drive currents require larger amounts of charge to be deposited to cause SETs. The same also holds true for circuits

with high capacitances and resistances because the current induced by the SE must overcome the nodal impedances of the circuit.

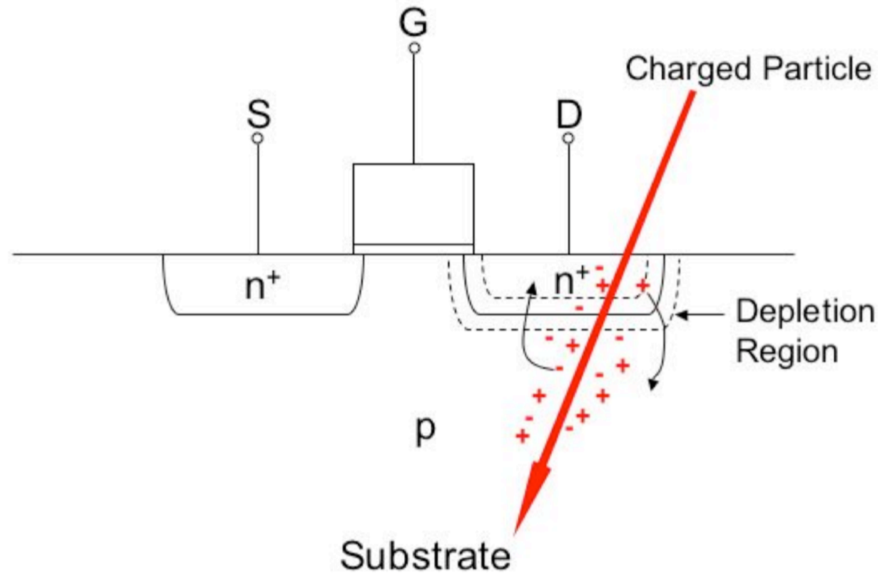

Figure 1. An illustration showing the generation of electron hole pairs caused by a single event [6].

In digital circuits, a SET that propagates unhindered can cause a single-event upset (SEU) [3] by altering digital states within the circuit and information stored in memory cells. When a memory cell latches a SET, the data latched is considered a SEU. As first reported by Binder, *et al.* in 1975 [7]**,** these SEUs can cause further errors within a circuit or can be read incorrectly at the output as corrupted data.

A key aspect of SEUs occurring in digital logic is that the SETs, generated from the logic, successfully propagate to the storage cells [6]. Combinational logic circuits can have many levels of logic, and these levels of logic can hinder the propagation of SETs. One type of barrier to SET propagation is called logic masking. Each level of logic may be made up of different logic gates with multiple input paths. A SET may try to propagate through one of those input paths but may not cause the logic gate to switch its output state. For example, if a

transient arrives that meets the minimum amplitude and pulse width to switch an inverter, it will propagate through the inverter, but if the same transient arrives at a NAND gate where the other input is a zero, then the transient will not propagate through.

Another type of masking is called temporal masking or “window of vulnerability”. The window of vulnerability is the period of time that determines whether the SET is latched or is not latched as shown in Fig. 2 [8]. A latch’s window of vulnerability is determined by many factors, such as the sampling window and SET shape [9] & [10]. The probability of latching an error is then given by the ratio of the window-of-vulnerability period to the clock period, which makes the pulse width of the SET a key factor in determining whether the SET is latched or not. Assuming that the transient is not logically masked and propagates through the combinational logic, then the last factor that determines whether the SET is latched or not is time. The SET must propagate within the latch’s window of vulnerability [8]. The probability of the SET falling within the latch’s window of vulnerability is tied to the SET’s pulse width. A latch’s window of vulnerability increases with increasing SET pulse width because the SET is more likely to fall within the latch’s sampling window as the SET increases in duration. Therefore, the probability of the SET falling within the latch’s window of vulnerability is directly proportional to the SET’s pulse width. Two final factors that influence the window of vulnerability are the technology node and the latch design.

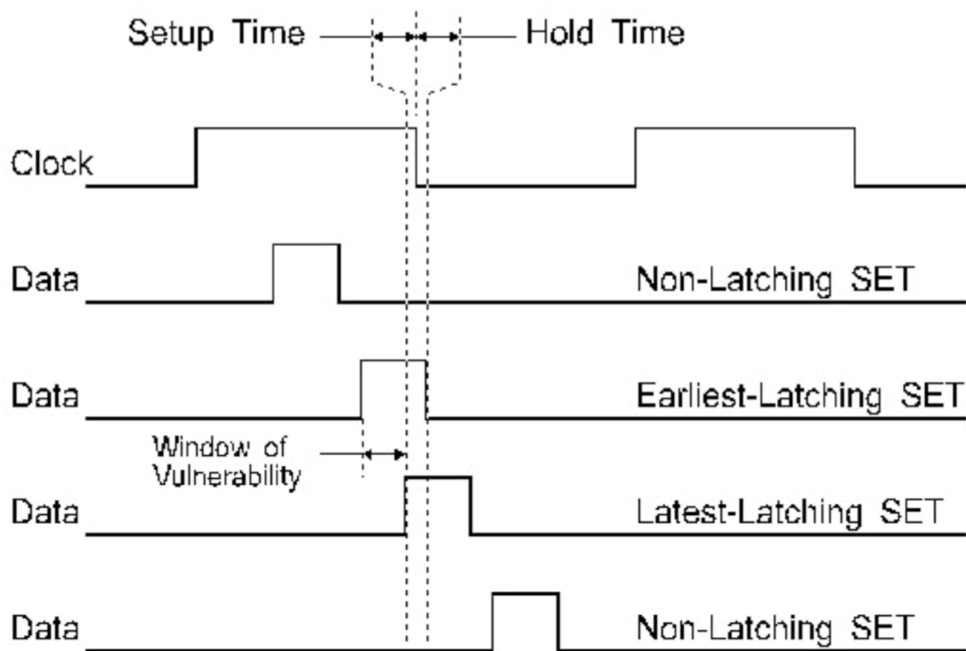

Figure 2. An illustration showing how a pulse may or may not be latched by a storage element [8].

Figure 2 gives an example of the temporal relationship between a SET being latched and a SET not being latched [8]. The example latch captures input data on the falling edge of the clock with the data having to maintain the same value from the setup time to the hold time. In the first case, the SET is not latched because the transient occurs before the falling edge of the clock. In the second and third case, the transients are latched because they appear during a clock falling edge while maintaining the same value during the setup and hold time of the latch. The final case shows another non-latching SET, but this time the SET appeared after the latching event. As illustrated by the figure, the importance of SET pulse width in determining the probability that the SET is latched as an error cannot be overstated.

The last issue that can hinder a SET from propagating to a memory cell or output is pulse attenuation. As SETs propagate, they can be attenuated as they pass through logic gates

[11]. Each gate has intrinsic resistance and capacitance that equates to a RC time constant. This RC time constant causes the gate to act like a low pass filter and "filter" out narrow or high frequency pulse widths. Figure 3 shows how a SET can be attenuated as it propagates through several gates [12]. Notice that not only is the pulse width decreased as it propagates but also its amplitude is decreased. So for a SET not to be attenuated it must meet a minimum pulse width and amplitude where it will not be filtered out by the natural low pass filter of the gates [13].

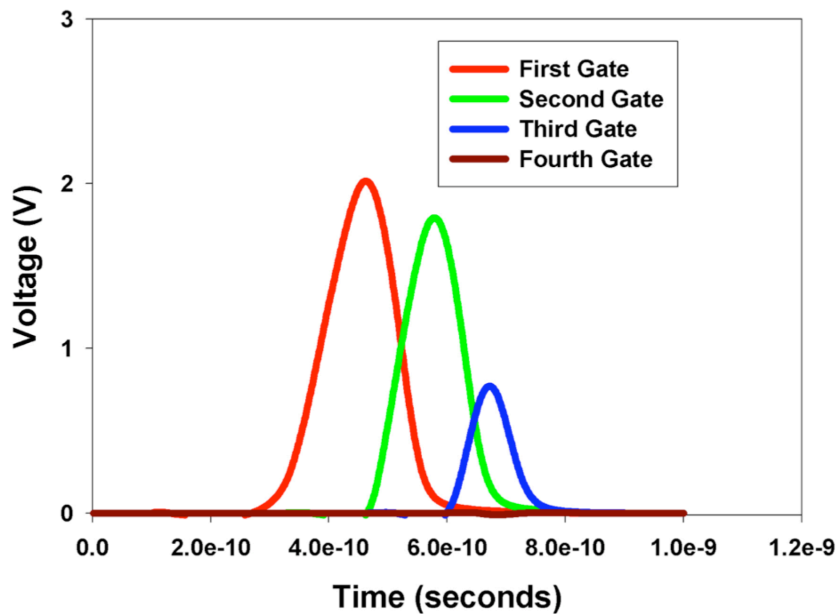

Figure 3. An illustration showing how a pulse may attenuate the farther it travels down a logic path [12].

Digital circuits are becoming faster, and with that there are more chances of latching errors because as circuits become faster so does the circuit's clock. An increase in clock speed translates to an increase in the number of clock pulses that cause a latch to change states. As circuit speeds increase, it is theorized that radiation induced combinational logic errors will dominate the radiation induced storage cell errors. Figure 4 predicts that as frequency increases the error rate of sequential logic will stay frequency independent while

the error rate of combinational logic will continue to increase with frequency [14]. Therefore, there exists a need to test combinational logic and storage cells at process specific clock speeds.

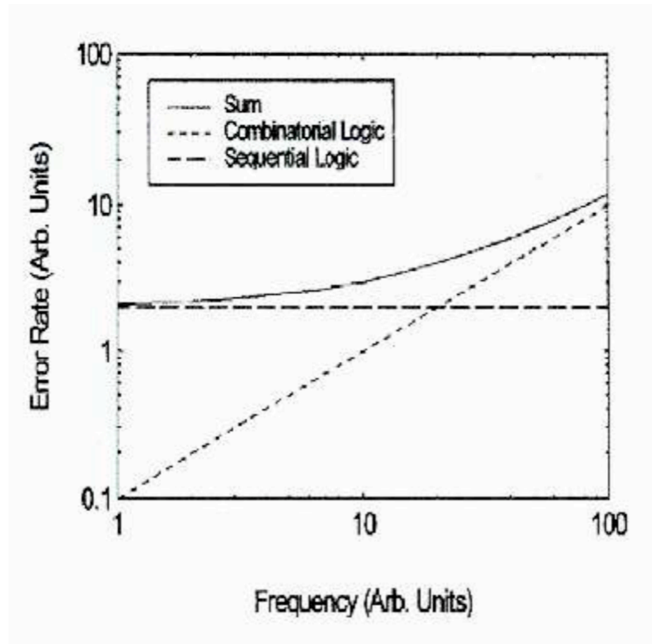

Figure 4. A plot that describes the error rate for combinational and sequential logic over frequency [14].

CHAPTER II

SINGLE-EVENT TEST CIRCUITS

Common Testing Technique

There are many ways to characterize SEEs in combinational logic [15][16]. A common technique of characterizing combinational logic involves strings of latches with interdigitated logic gates as shown in Fig 5. By adding logic gates between latches the logic increases the amount of chip area combinational logic uses. The increase in combinational logic area increases the probability a SE will occur and thus increase the single-event error rate. However, each logic gate has an inherent switching delay, and the more gates that are put between two latches, the longer it will take for a SET to propagate. The delay in signal propagation then hinders the ability of the latches to be run at full speed.

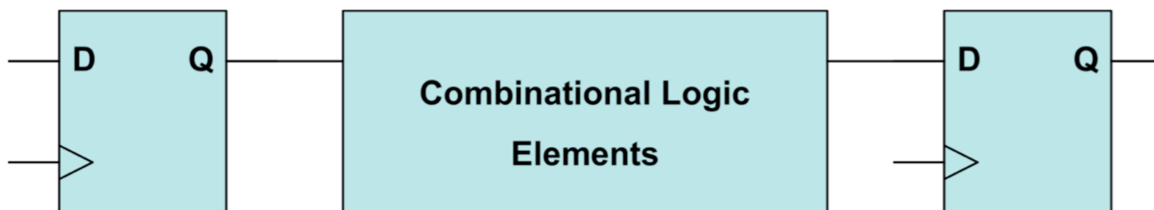

Figure 5. Schematic presenting traditional technique of characterizing SEEs in combinational logic.

Unfortunately, for the technique to work without suffering from severe propagation delay issues, only few logic gates can be used. The lack of logic gates creates only a minor difference between the chip areas taken up by the combinational logic and the storage cells. There is not a sufficient enough difference between the two types of areas to allow the distinguishing of errors generated from combinational logic and from storage cells.

A final issue with the common technique is that it requires the use of external Bit Error Rate Testers (BERTs) and clocks. The addition of external equipment forces the addition of

high frequency input/ouput buffers that take up additional space on the test chip. Also the external equipment costs additional money compared to a test circuit that has those capabilities on chip.

Circuit for Radiation Effects Self-Test

A recent technique to characterize SEEs is the Circuit for Radiation Effects Self-Test (CREST) [17]. The CREST design is innovative in moving many of the elements necessary for high speed testing on-chip. Moving many of these functions on-chip significantly reduces the switching frequency needed for chip I/O and the costs associated with purchasing high frequency test equipment for testing.

Using IBM's 5AM SiGe HBT logic family, the prototype CREST design implemented a built-in self test (BIST) allowing all error/upset detection to be handled on the test chip. The on-chip BIST allowed the design to operate at speeds of up to 5 Gbit/s. The high on-chip operating speeds allowed circuits to be tested close to their native operating speeds.

An overview of the CREST design is shown in Fig. 6 [17]. From the figure it can be seen that the data source feeds into both the test structure and the error detection circuitry. Within the test structure are 127 master/slave D flip-flop stages that output to the error detection circuitry. In normal operating conditions, where a specific data pattern is used, the output from the test structure and the data source are identical. However, when there is a mismatch, an error is registered by the circuit.

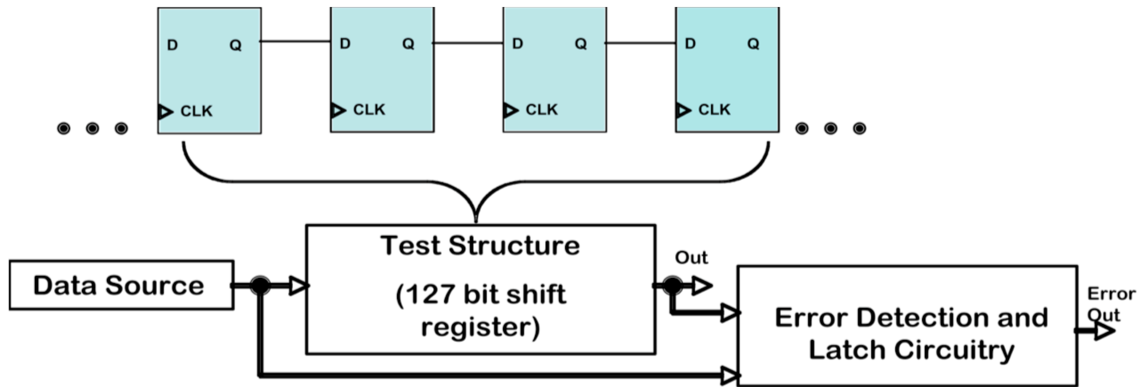

Figure 6. Schematic presenting basic concept of the original CREST design [17].

Proposed Technique

The new approach discussed in this thesis is an extension of the CREST approach and builds on the idea of growing the area of the combinational logic. However, it takes the bulk of the combinational logic elements out of the propagation path between two adjacent storage cells and keeps only a single XOR gate as illustrated by Fig. 7. In this approach, the maximum operating speed of latches is only affected by the single XOR gate delay. Also the number of combinational logic elements can be made arbitrarily large without reducing the clock speed. The combinational logic can then be designed to take up a greater amount of chip area than the storage cells, which would increase the probability that the error cross-section of the combinational logic can be distinguished from the error cross-section of the latches. Therefore, the combinational logic cross-section is limited only to chip area, and the combinational logic can be characterized at its native operating speeds.

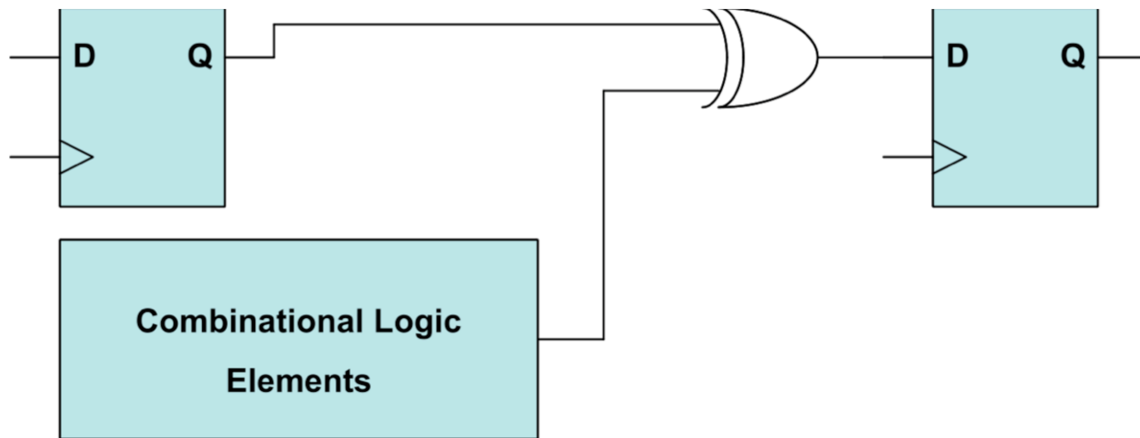

Figure 7. Schematic presenting new approach to evaluating the SET impact of combinational logic allowing an increase in combinational logic area without a significant loss in maximum clock speed.

The separation of the combinational logic elements from the main input signal path allows the logic elements to use static inputs. Multiple static inputs can be designed to drive logic loads that vary in depth of design while still being able to output a static state of ‘0’. Having an output of ‘0’ from the combinational logic elements as the nominal output maximizes the ability to observe a SET occurrence because a SE in the combinational logic will cause the output to switch from ‘0’ to ‘1’. The switching of the output caused by the propagation of the SET can then be registered by the XOR as it switches to allow the propagation of the SET. Once the SET propagates through the XOR as a full rail pulse, it can be latched as an error.

Without having to worry about propagation delay caused by using numerous logic elements along with being able to use static inputs, many different types of logic, in addition to combinational logic, can be used. Traditionally inverters are used between the latches; but because of the separate input of the combinational elements into the XOR gate, NAND gates and NOR gates can also be used. A pitfall of using combinational logic between latches is that there is only one input. Using a separate block attached to an XOR allows different fan-in and fan-out designs where multiple static inputs can be used. Other logic elements, outside

of the previously mentioned logic families, that could be used are CVSL, guard gates, static logic designs, and dynamic logic designs.

CHAPTER III

COMBINATIONAL CIRCUIT FOR RADIATION EFFECTS SELF-TEST

Utilizing the CREST approach and extending upon it with inserting XOR gates between latches, the designs from Fig. 6 [17] & 7 were integrated together as shown in Fig. 8. The design in Fig. 8 is called the Combinational Circuit for Radiation Effects Self-Test (C-CREST). It consists of the original CREST approach where errors can be detected on chip, and it consists of the single XOR gate between two latches. All of the 576 latches are dual interlocked cells (DICE) [18]. DICE latches are inherently designed to be less susceptible to radiation effects than basic latches making them harder to be upset by a SE, which minimizes the static error cross-section of the latches. The minimizing of the error cross-section of the latches increases the probability that the error cross-section of the combinational logic will dominate the total error cross-section of the latches. To also increase the probability that error cross-section of the combinational logic dominates the total error cross-section of the latches, the two DICE latches with the XOR and combinational logic are repeated creating a combinational logic cross-section that is larger than the cross-section of the latches. The XOR gate minimizes the propagation delay and allows the combinational logic to be characterized at native operating speeds.

A test chip with the C-CREST design was fabricated through MOSIS in the IBM 90 nm CMOS9SF process [19], a process that has been shown to be SE vulnerable [20]. On the test chip, three different test structures were developed. The first variation, the baseline, consisted only of DICE latches and did not include any logic or XOR gates. Figure 9 shows

the second and third variations that focused on inverter designs, specifically matched current and weak-p designs shown in Table 1. The weak-p design consists of the NMOS being twice the size of the PMOS along with the NMOS having a higher current drive than the PMOS. In the last two variations, each combinational logic block contained 74 inverters. Last, the drain area of the two inverters is similar so the number of events recorded should also be similar.

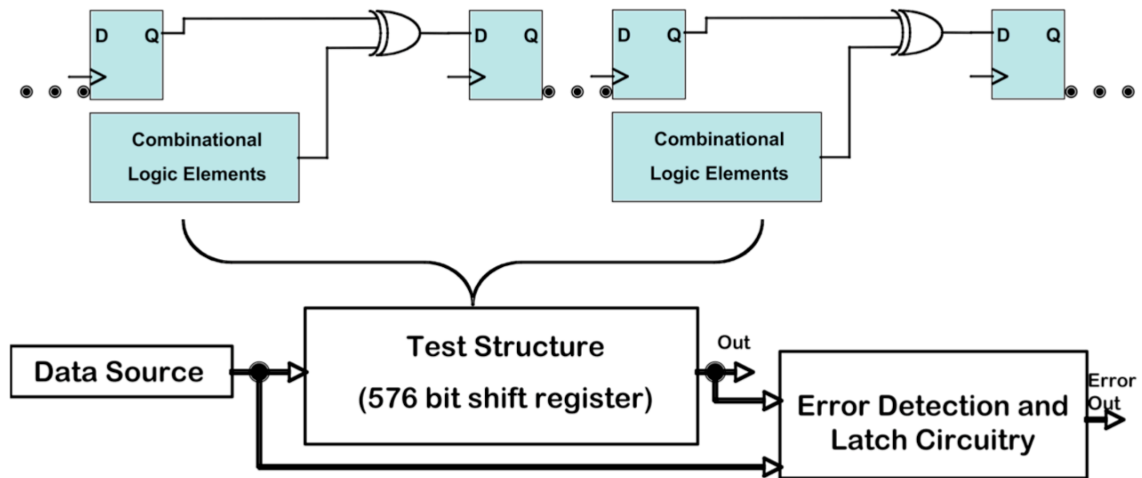

Figure 8. Schematic presenting new SET evaluation approach implemented as C-CREST.

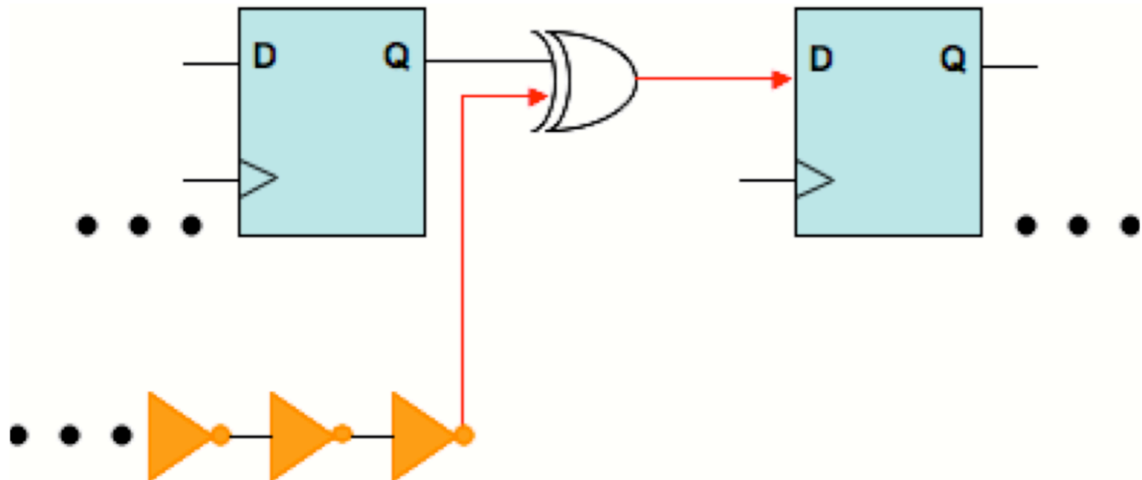

Figure 9. Schematic presenting C-CREST test structure using an inverter chain.

Table 1 – Inverter Designs

Circuit Type
Baseline – No Inverters
Matched Current Inverters (480 x 100 nm PMOS) (200 x 100 nm NMOS)
Weak-P Inverters (200 x 100 nm PMOS) (400 x 100 nm NMOS)

Finally, as stated previously, the combinational logic uses a significant amount of chip area. Figure 10 is an image of the C-CREST layout highlighting the combinational logic portion of the chip and showing how the logic takes up almost half the chip area [21]. A comparison of the silicon area taken up by the shift register without inverters to the area taken up by shift registers with inverters showed a 3X increase in silicon area used. Having such a large silicon area difference on the test chip increased the probability of SETs being generated in the combinational logic and of them being latched as errors by the DICE latches.

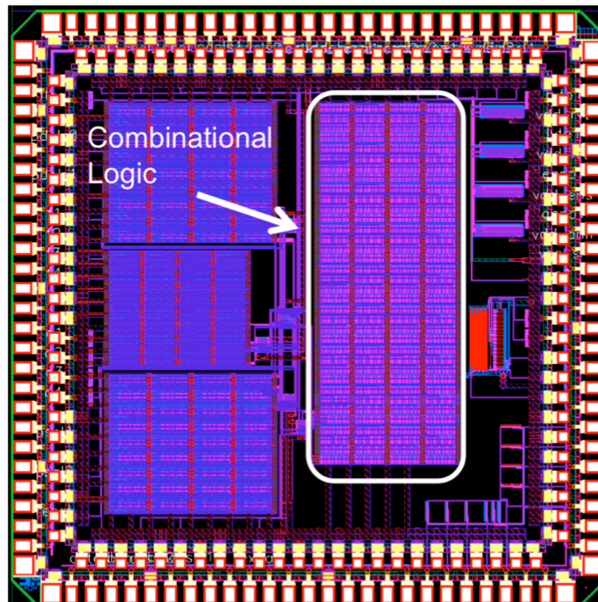

Figure 10. Picture of the IBM CMOS9SF die with highlighted area showing the combinational logic portion of the C-CREST circuit [21].

CHAPTER IV

EXPERIMENTAL SETUP

For the C-CREST testing experiment, all heavy-ion testing was performed with the 88" Cyclotron at the Lawrence Berkley National Laboratory. All the ions used for the experiment were from the 10 MeV/nucleon ion cocktail. Table 2 shows the selection of ions that are part of the cocktail, and for the experiment the ions of Ne, Ar, Cu, Kr, and Xe were selected. Those ions were selected because they provided a wide range of LETs to characterize the logic.

Table 2 – 10 MeV/nucleon ion sources available at LBNL [22].

Ion	**Energy (MeV)**	**Z**	**A**	**Chg. State**	**% Nat. Abund.**	**LET 0° (MeV/(mg/cm^2))**	**LET 60° (MeV/(mg/cm^2))**	**Range (μm)**
B	108.01	5	11	+3	80.1	0.89	1.78	305.7
O	183.47	8	18	+5	0.2	2.19	4.38	226.4
Ne	216.28	10	22	+6	9.25	3.49	6.98	174.6
Si	291.77	14	29	+8	4.67	6.09	12.18	141.7
Ar	400.00	18	40	+11	99.6	9.74	19.48	130.1
V	508.27	23	51	+14	99.75	14.59	29.18	113.4
Cu	659.19	29	65	+18	30.83	21.17	42.34	108.0
Kr	885.59	36	86	+24	17.3	30.86	61.72	109.9
Y	928.49	39	89	+25	100	34.73	69.46	102.2
Ag	1039.42	47	107	+29	51.839	48.15	96.30	90.0
Xe	1232.55	54	124	+34	0.1	58.78	117.56	90.0

To run the chip and obtain data, it was necessary to use the NASA-GSFC low cost tester along with a daughter card and counter. The low cost tester consisted of a Spartan-3 field programmable gate array (FPGA), voltage regulators, configuration components, memory, I/O header, clock circuitry, and other components shown in Fig. 11 [23]. The components used for this test were the FPGA, JTAG header, and the User I/O header with

the counter attached to the outputs of the daughter card.

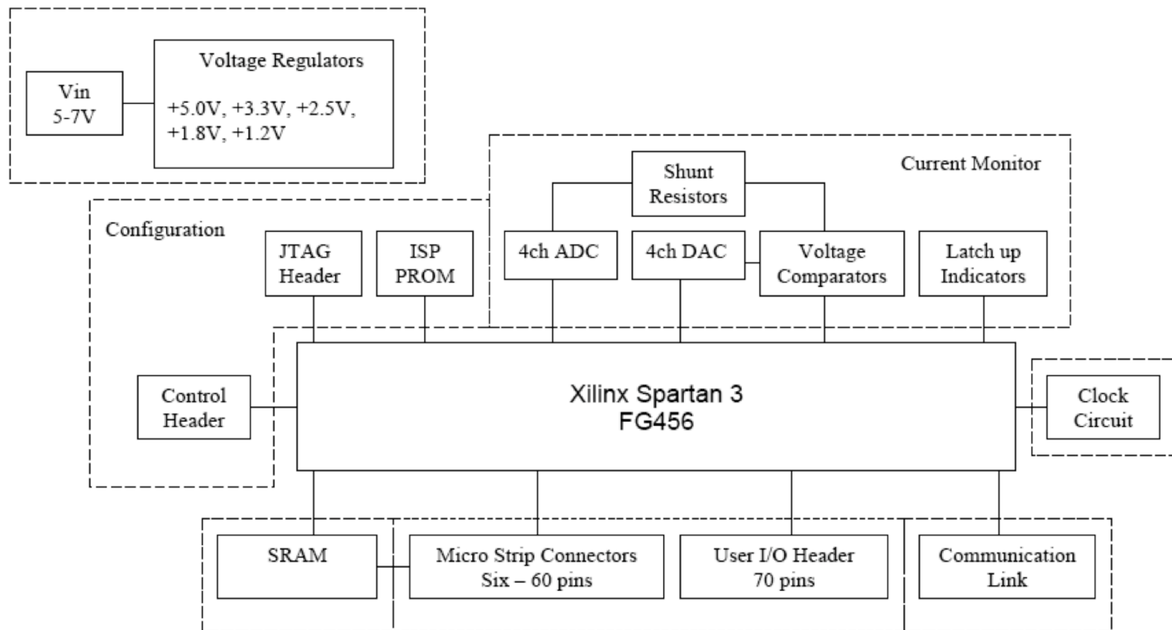

Figure 11. Block diagram of the low cost tester [23].

To program an FPGA, the typical language used is a hardware descriptive language (HDL), such as Verilog or VHDL. In the case of the C-CREST chip, VHDL was used to program the FPGA and DUT. The process to program the chip and store the output data involved two other external devices. The low cost tester contained a JTAG header that was used to configure the board via an EEPROM and was connected to a laptop. Every time a test was conducted it was necessary to reprogram the low cost tester. To read the errors from the DUT, BNC cables were run out of the vacuum chamber via mounting hardware and connected to a counter. This counter then was setup to count every time there was a difference in the output data stream from the chip. An illustration of this complete setup is shown in Fig. 12 [21].

A simple program was written in VHDL for each test of the DUT. This configuration sets-up the default state of the DUT and low cost tester. These included determining the input

pattern into a high state or low state, the frequency of the on-chip clock, and the shift registers that would be under test. Figure 13 shows the basic flow chart of the VHDL program [24] and a sample VHDL program is included in the Appendix.

To program from the laptop to the FPGA via the JTAG header, a program called the Xilinx ISE Impact tool was used. The basic function of the program was to send the compiled VHDL for that specific test to the FPGA, which it auto-configured upon startup. This process was then repeated for each test that required different parameters for the DUT.

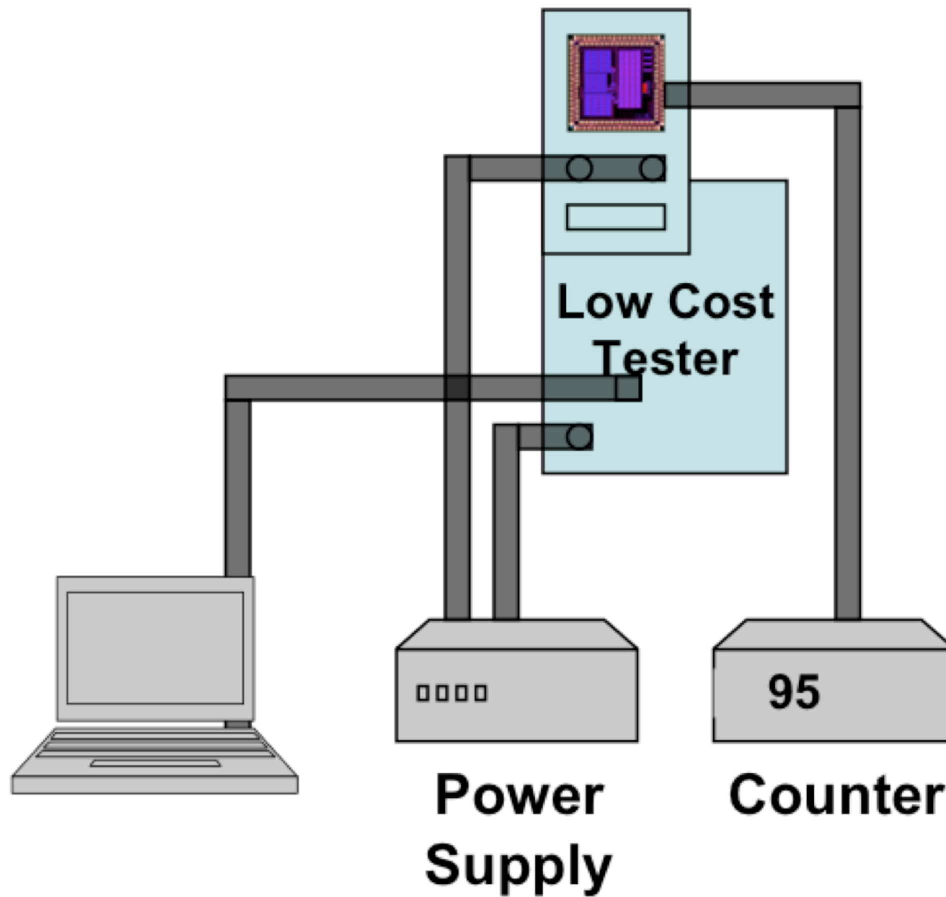

Figure 12. Diagram of the C-CREST test setup [21].

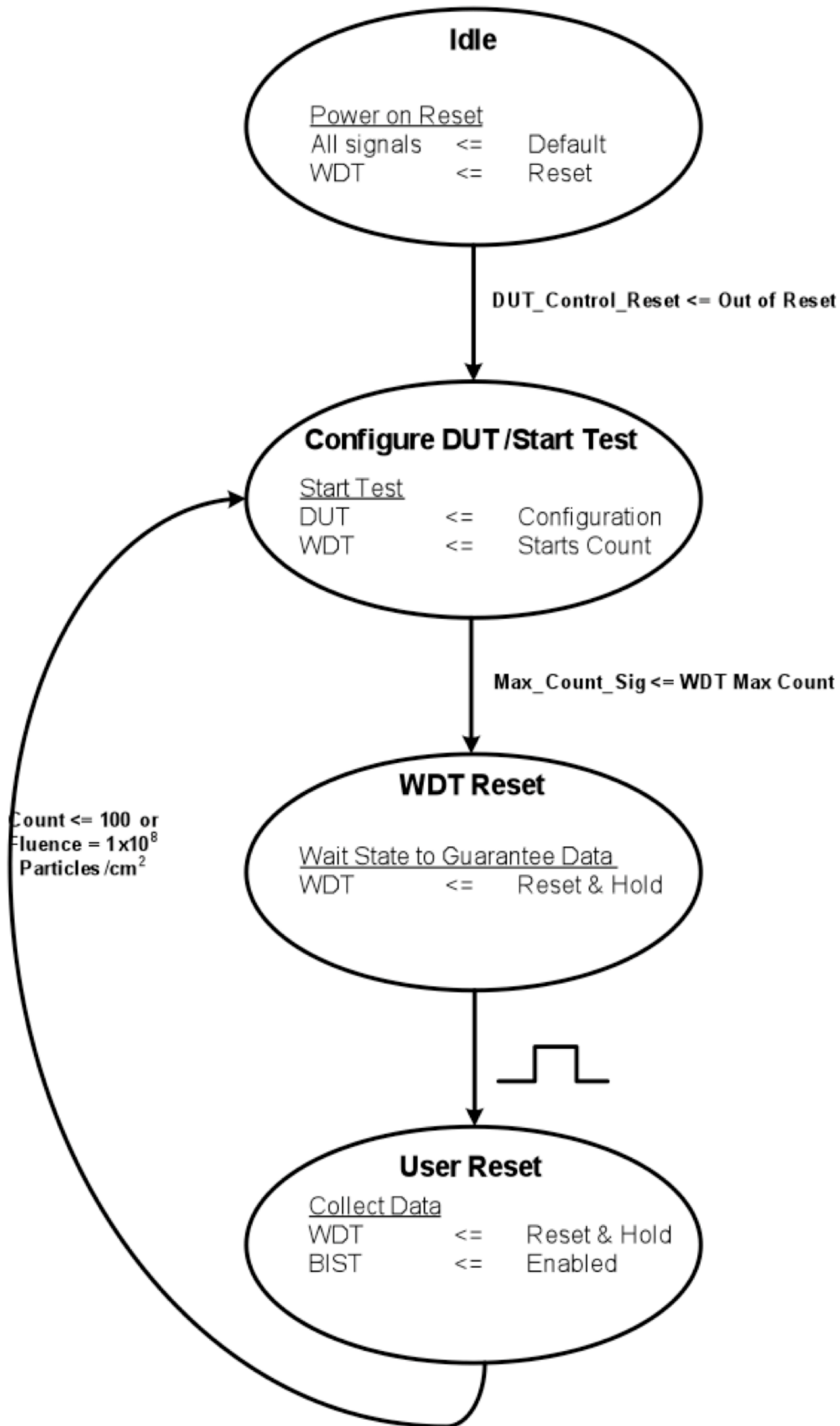

Figure 13. Block diagram of test process [24].

Each shift register chain was tested over the same varying frequencies and heavy-ion energies with the DUT aligned so the ions struck the chip at normal incidence, and these conditions are shown in Table 3. Before the experiment and to insure the clock frequency was accurate, an output from the C-CREST chip was attached to an oscilloscope. A 010 pattern was generated from the output and the clock frequency was varied. For each clock variation, the period of the output pattern was measured with the oscilloscope and the clock frequency was found. However during the test, only one pattern could be used because of testing time constraints, which was a constant string of 1's that was generated by the FPGA. To attain the higher frequencies of 100.5 MHz and 201 MHz on the chip, it was necessary to run the chip at 1.39 V. Normally the voltage of the chip would be run at 1.2 V. Last, each experiment was tested for either a 100 errors or a fluence of 1 x 10^8 particles/cm^2.

Table 3 – Test Conditions

Circuit Type	**Clock Freq. (MHz)**	**LET (MeV-cm^2/mg)**
Baseline – No Inverters	12.56, 50.25, 100.5, 201	3.49, 9.74, 21.0, 30.9, 59.1
Matched Current Inverters	12.56, 50.25, 100.5, 201	3.49, 9.74, 21.0, 30.9, 59.1
Weak-P Inverters	12.56, 50.25, 100.5, 201	3.49, 9.74, 21.0, 30.9, 59.1

CHAPTER V

EXPERIMENTAL RESULTS

A common method of collecting and displaying heavy-ion data is by using error cross-section curves. Figure 14 gives an example of a typical heavy-ion cross-section curve [12]. The data collected during the C-CREST heavy-ion testing is displayed no differently. Every plot of the heavy-ion data contains an error cross-section curve. To find the error cross-section for a set of data, the calculation is simply the number of errors over the fluence. When analyzing an error cross-section curve, there are three key parts of the curve. The first part is called the threshold or onset LET that is defined as a value 10% - 15% below saturation [25]. The next part of the curve is the knee, which is the point where the cross-section begins to saturate or plateau. Error cross-section saturation is where the number of errors stops increasing with increasing LET. The error cross-section will then saturate near the sensitive area, which is the maximum amount of area on the chip that can cause errors in the circuit.

The C-CREST error cross-section data is displayed as the SET cross-section in units of cm^2/bit. SET cross-section is used because the C-CREST technique measures the number of SETs that are generated and detected. Thus, the error cross-section is representative of the number of SETs being generated in the circuit. The SET data is also normalized by the number of bits in the shift register chains. In the shift register chains, a bit is defined as one DICE latch. Regardless of whether a shift register chain contains combinational logic or not, each chain has 576 DICE latches or 576 bits.

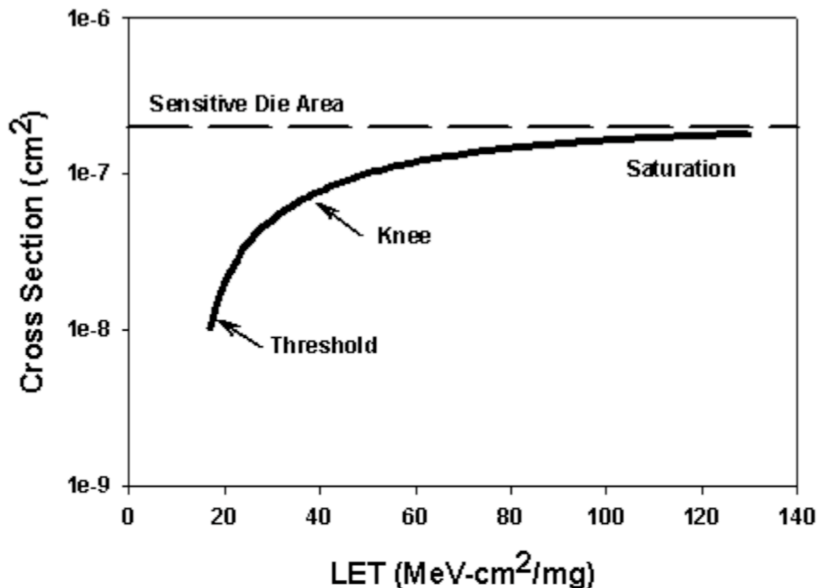

Figure 14. Plot of a basic cross-section curve [12].

As previously discussed, the C-CREST technique measures the number of SETs that are generated and then latched. SETs are only recorded if they are latched, and for a SET to be latched it must be wide enough to maintain the same value during a latch's setup and hold time. So any SET that is smaller than the latch's setup and hold time will not be detected. Knowing there is a minimum SET pulse width the C-CREST circuit can detect, the DICE latch was simulated in Cadence Spectre at the maximum clock frequency attainable during testing – 201 MHz. Simulations showed that the minimum SET pulse width that could be latched was 150 ps.

In any experiment, there are sources of experimental error that can affect the data collected. Two sources that make up experimental error are statistical error and systematic error. Systematic error is the inherent error that exists in the equipment being used to run the experiment. This error is very difficult to be quantified, but as long as the same equipment is used for all experiments then each experiment should be equally affected and not affect the

possible trends of the data. Statistical error is dependent on the number of data points collected during an experiment. Traditionally, error bars are used to show the accuracy of the data by giving a range where the true data or error free data would fall. The more data points collected the better the statistics are and thus the smaller the error bars are on plots. Fewer data points leads to a larger variability in the data because there are fewer data points to support the trends of the data. To calculate error bars for heavy-ion data, the standard deviation of the number of errors is taken and it is normalized by the fluence. Standard deviation is calculated because it is a statistical method to show the variability in the data. Lastly, error bars are included in every plot of the C-CREST data, but the data points obscure the error bars because the amount of statistical error is low.

A common source of error found in digital circuit experiments is clock skew. Luckily, errors due to clock skew are minimized in the C-CREST data because a constant pattern was used. The advantage of using a constant input pattern whether a string of 1's or a string of 0's is that clock skew will not cause an incorrect value to be latched. Referring to Fig. 15, the inverter string nominally outputs a '0' value and only goes to '1' when an upset occurs. If a constant pattern of 1's or 0's is passing through the latch and through one input of the XOR, the output of the XOR will be constant until an SE occurs.

One place clock skew or hits on the clock line could have introduced errors is when a SET is propagating through the latches. Clock skew could have caused SETs to not have been latched and thus masked – never being read at the output as an error. Purposefully, errors bars are used in the plots to take into account these possibly masked SETs.

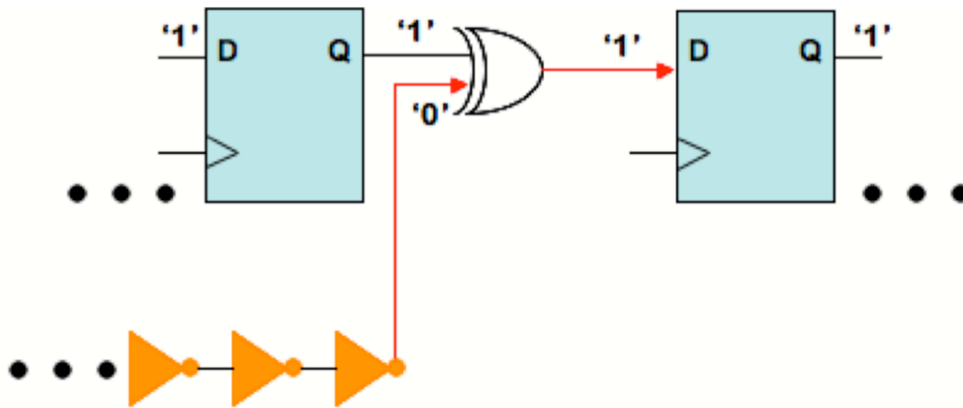

Figure 15. Schematic presenting logic states of C-CREST cell before an upset.

One source of error that was not introduced in the experiment was the use of an alternating input pattern. Before the experiment, it was found that clock skew existed in the design and that the alternating input pattern did not match the output pattern, which led to bit errors being detected. In the circuit there was most likely clock delay, causing the latches to occasionally miss a bit, registering one of the input patterns incorrectly, as shown in Fig. 16 [26]. In future iterations, clock skew should be addressed to allow alternating input patterns to be used.

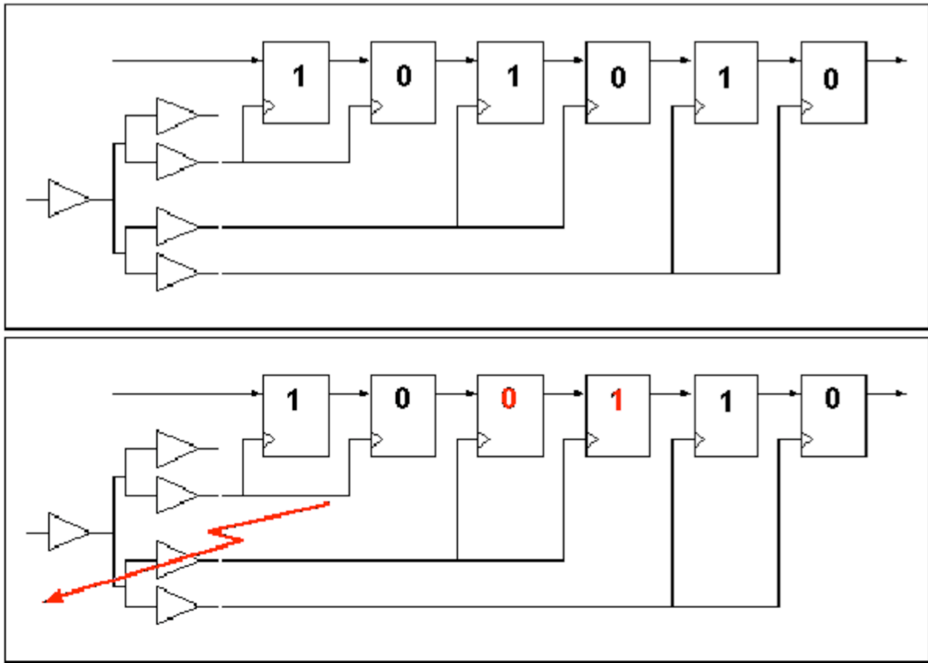

Figure 16. Schematic illustrating effects of a clock error when using an alternating input pattern [26].

CHAPTER VI

DISCUSSION

As discussed in Chapter IV, the C-CREST circuit was tested under various conditions. One condition of interest was characterizing the three shift register chains at 100.5 MHz. Figure 17 shows the measured SET cross-section versus LET at 100.5 MHz. From the figure, it can be seen that there is roughly an order of magnitude difference between the baseline and the combinational logic chains. The main reason for the difference is the amount of SETs uniquely generated within the combinational logic due to the fact that a significant portion of the chip area is dedicated to the combinational logic.

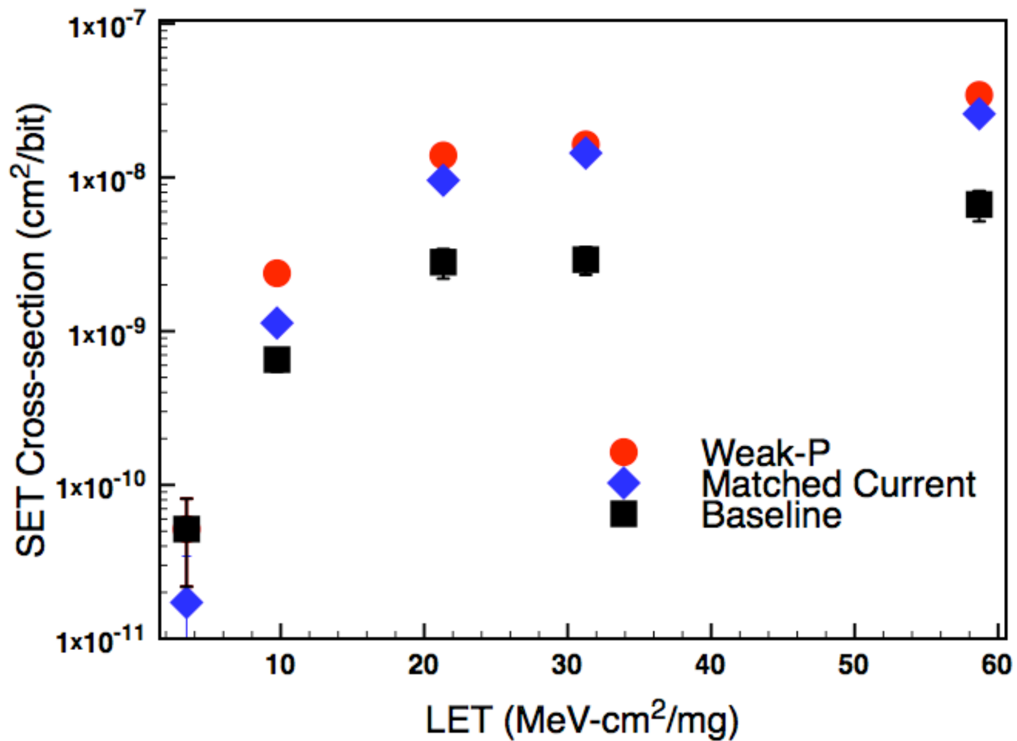

Figure 17. SET cross-section plot depending on LET for the clock frequency of 100.5 MHz.

In this version of the C-CREST design, 201 MHz was the highest frequency attainable. Figure 18 shows the SEE characterization of that frequency. The plot is very similar to Fig. 17 in that there is roughly an order of magnitude difference between the baseline data and the combinational logic data. This difference can also be attributed to the SETs generated in the combinational logic.

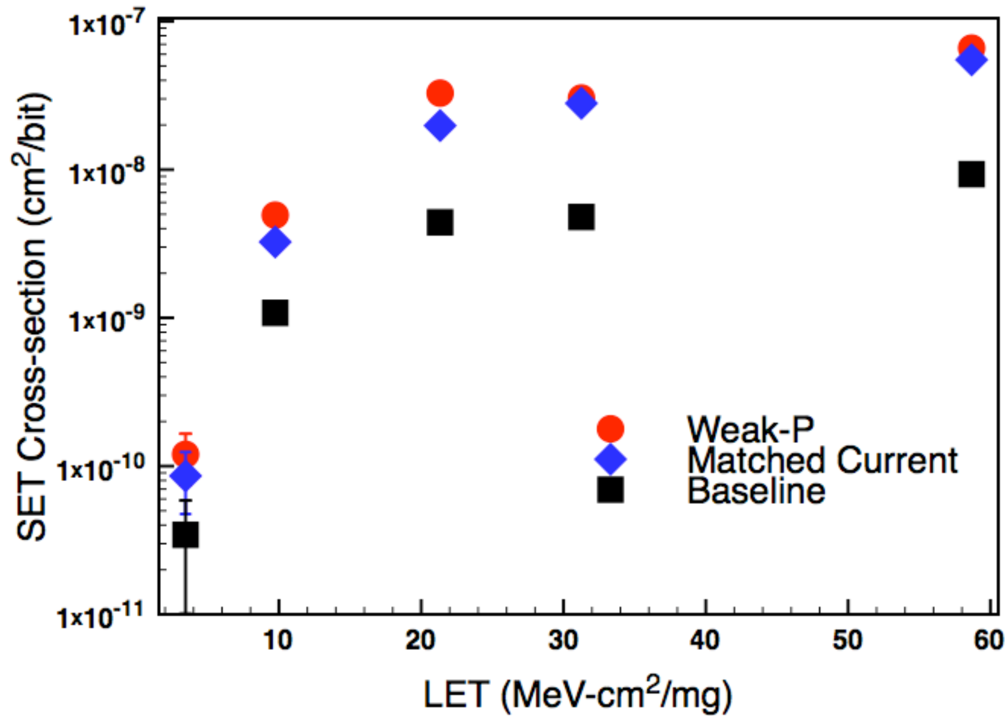

Figure 18. SET cross-section plot depending on LET for the clock frequency of 201 MHz.

However, as data were plotted at lower clock frequencies the significant difference in SET cross-sections between the combinational logic and the latches became less apparent. Figures 19 and 20 show the SET cross-sections for the clock frequencies of 50.25 MHz and 12.56 MHz respectively. The likely reason for the lack of distinction is because the majority of SETs generated from the combinational logic are not being latched. At the lower clock frequencies, there are not enough latch events during a given time to detect the SETs.

Also of note in the previous figures is that the baseline has a threshold around 8 – 9 LET. At first this may seem surprising given that the DICE latches are hardened, but previous work by Black, *et al.* showed a 130 nm DICE latch, similar in design to the C-CREST DICE, upsetting around 13 LET [27]. Knowing the C-CREST chip was fabricated in 90 nm, observing DICE upsets around 8-9 LET is not unexpected.

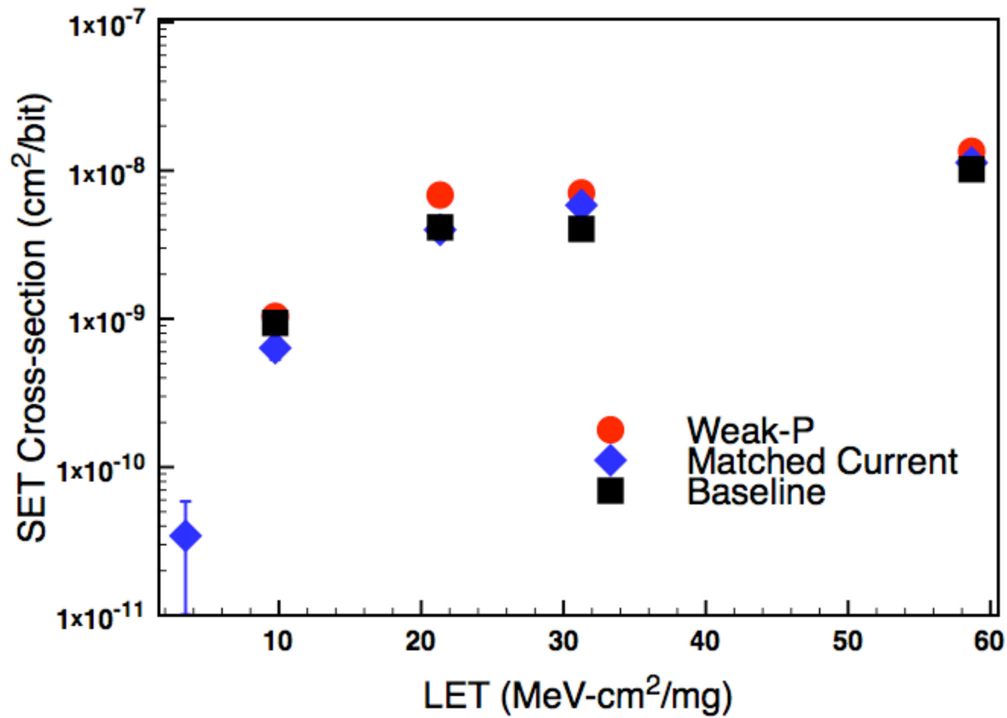

Figure 19. SET cross-section plot depending on LET for the clock frequency of 50.25 MHz.

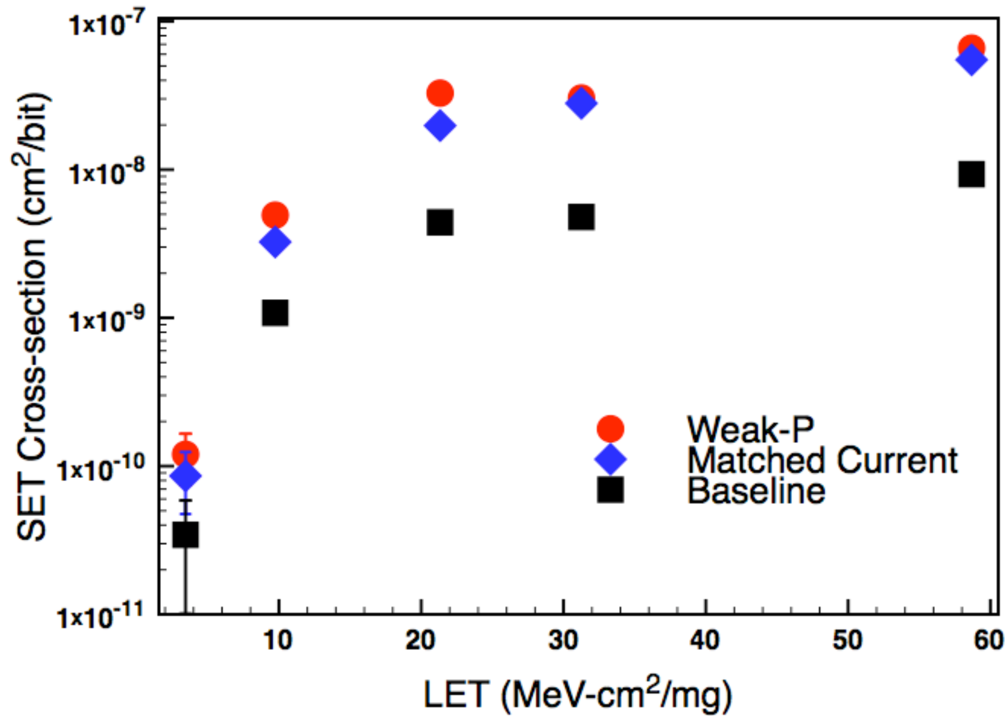

Figure 20. SET cross-section plot depending on LET for the clock frequency of 12.56 MHz.

The dependence on clock frequency to detect SETs was a key part of the C-CREST technique. It was hypothesized that the number of SETs detected would increase as clock frequency increased. The previous plots agree with this hypothesis but Figs 21, 22, 23, 24, and 25 plot the data differently with the x-axis varying over frequency showing better agreement. The reason that there were more SETs detected as clock frequency increased was because there were more latch events (clock edges that caused the latch to latch) leading to an increase in the probability to capture those transients.

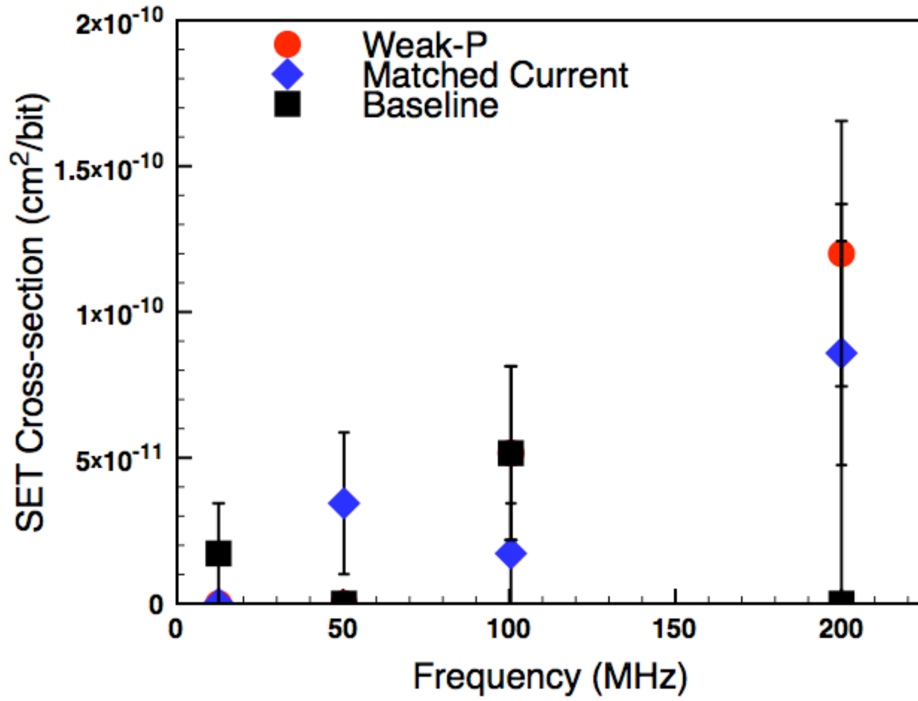

Figure 21. Frequency dependence of SET cross-section for the ion neon with an energy of 3.49 MeV-cm^2/mg.

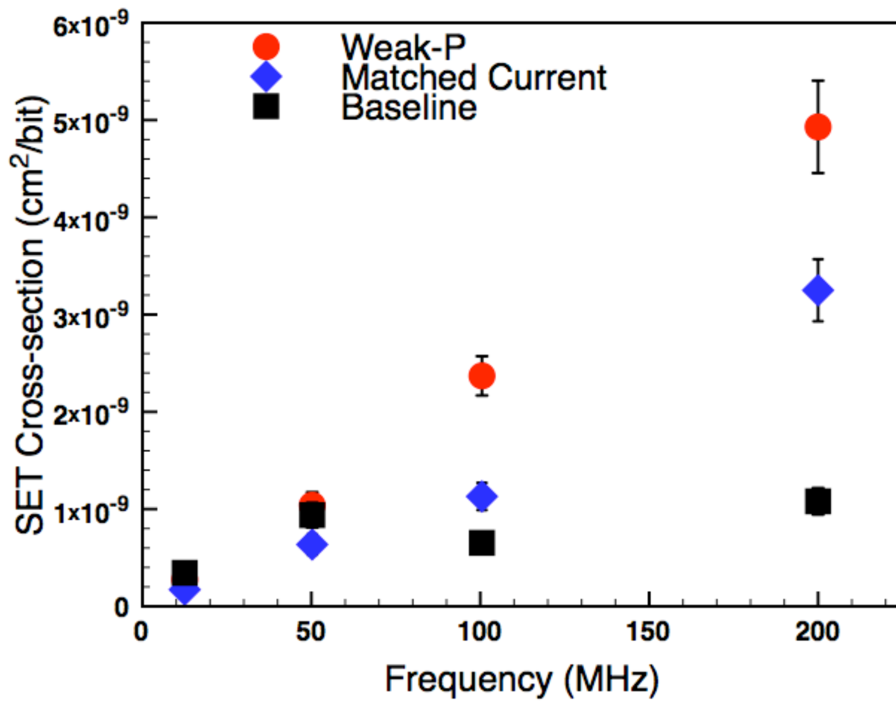

Figure 22. Frequency dependence of SET cross-section for the ion argon with an energy of 9.74 MeV-cm^2/mg.

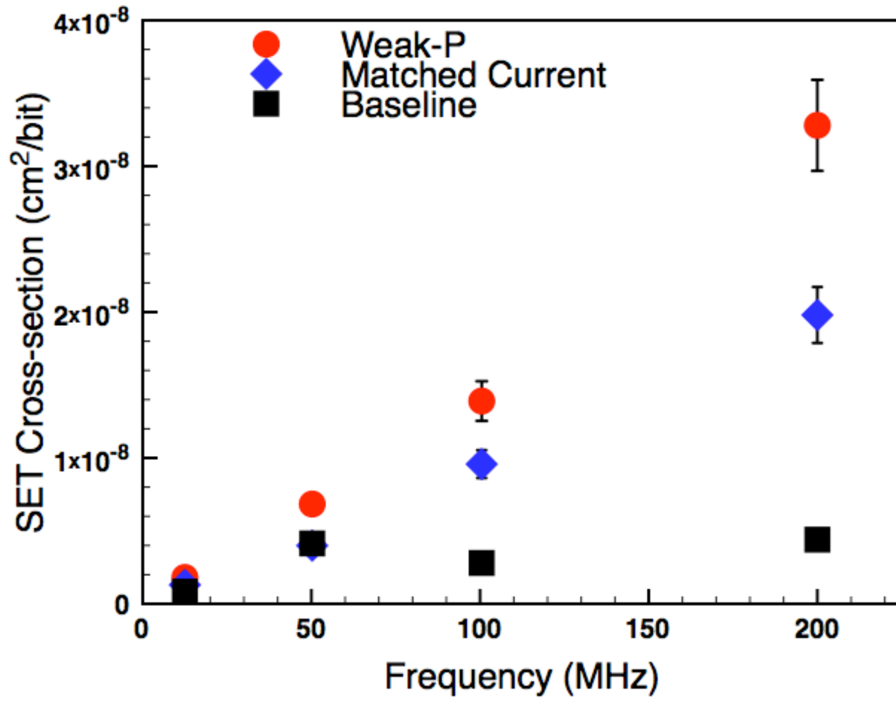

Figure 23. Frequency dependence of SET cross-section for the ion copper with an energy of 21.0 MeV-cm^2/mg.

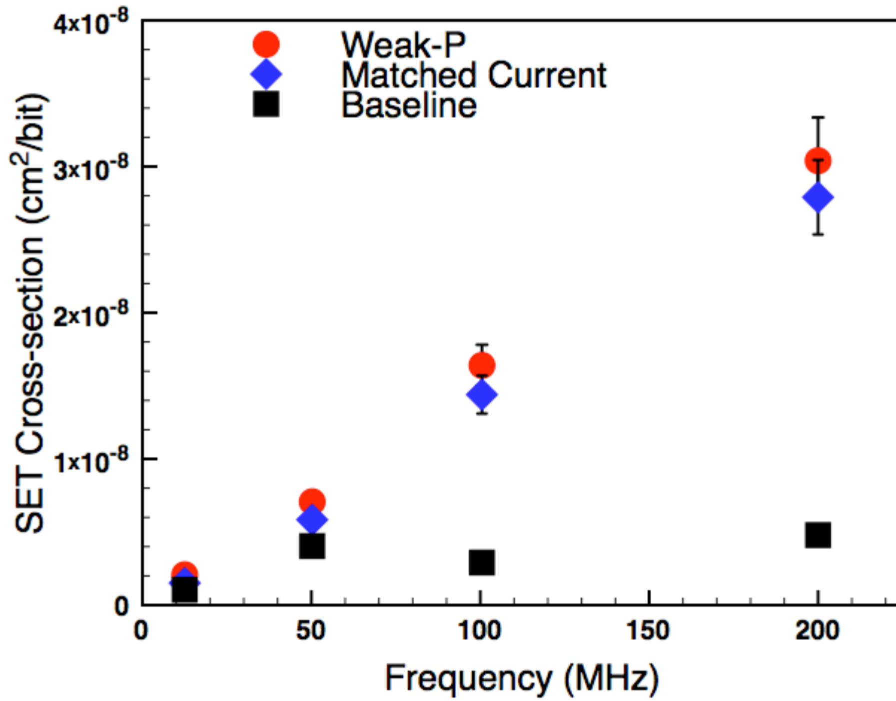

Figure 24. Frequency dependence of SET cross-section for the ion krypton with an energy of 30.85 MeV-cm^2/mg.

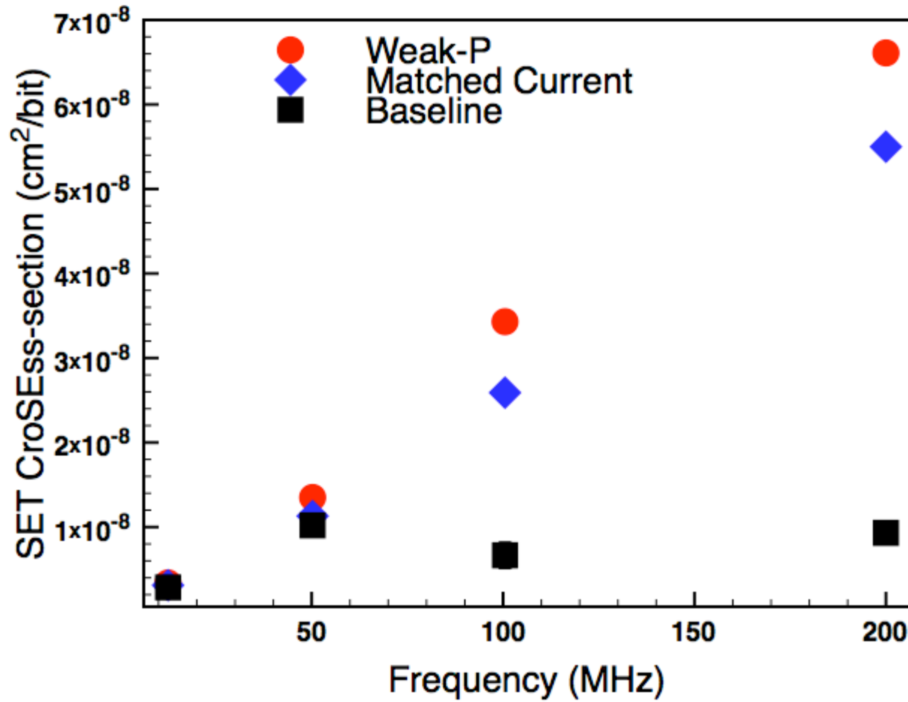

Figure 25. Frequency dependence of SET cross-section for the ion xenon with an energy of 58.68 MeV-cm^2/mg.

Figures 21 through 25 show data from various ions, and it can be seen that initially in Fig. 21 for Ne that the statistical error is very high because of the error bars. For each data point, there is a large range where the correct data may lie allowing no conclusive trend to be found in the plot. While in Figs. 23, 24, and 25, a significant difference can be seen between the baseline and the combinational logic chains as their SET cross-sections increase linearly with time. A 7.5X increase in slope between the weak-p and baseline circuits in Fig. 24 highlights this difference between the shift register chains. The baseline SET cross-section increases slightly in Figs. 22, 23, 24, and 25 compared to the significant linear increase of the combinational logic SET cross-sections agreeing with Buchner, *et al.* [14] that errors in latches are frequency independent. In addition, the data agrees with Buchner, *et al*. [14] that errors generated from the combinational logic increased linearly as clock frequency was increased.

Previous plots contained data from all three shift register chains, Figs. 26, 27, and 28 illustrate data from the baseline, weak-p, and matched current shift register chains respectively. As in Figs. 22 to 25 the baseline SET cross-section in Fig. 26 does not show a significant change over the four different clock frequencies. While in Figs. 27 and 28, there is an increase of 16-18X in cross-section from 12.56 MHz to 201 MHz. Once again, something that is not unexpected with the increased number of latch events at the increased clock frequencies.

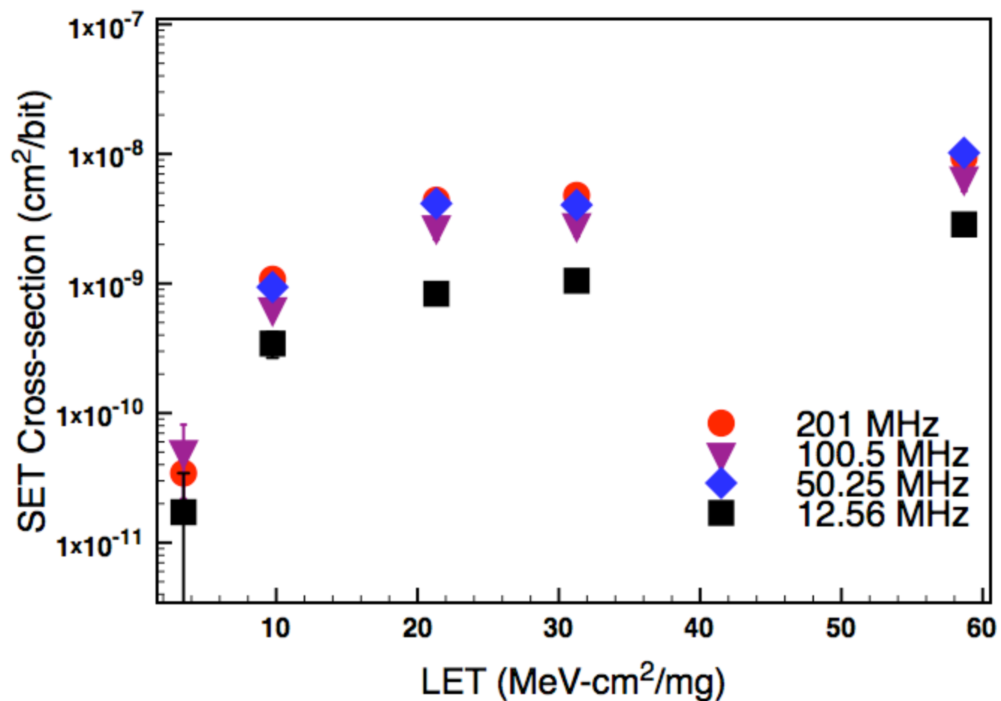

Figure 26. Baseline shift register chain plot showing SET cross-section dependence on LET.

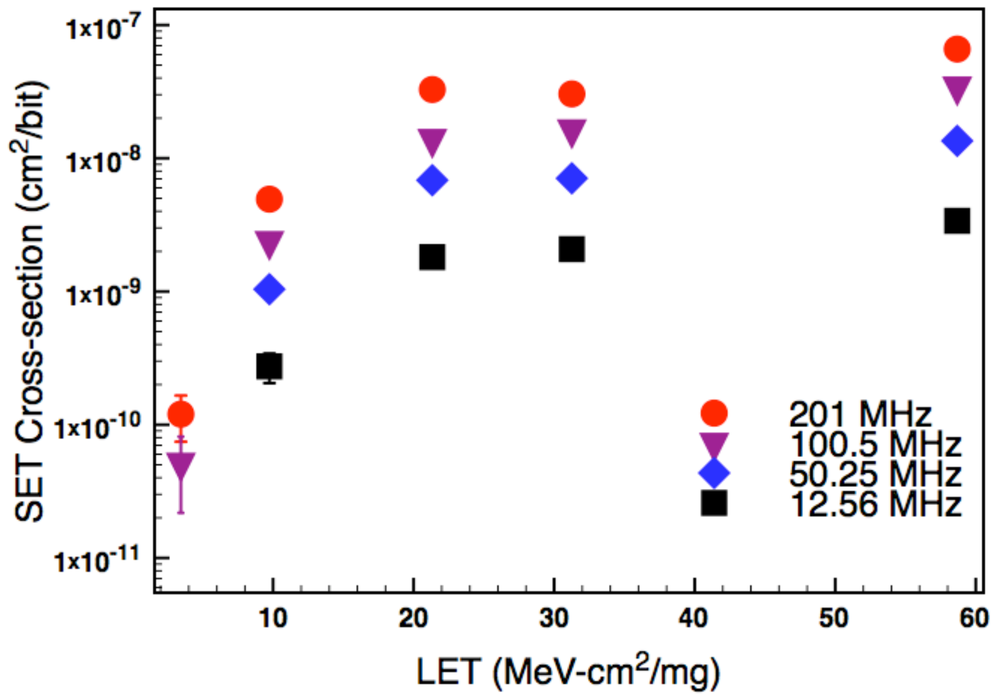

Figure 27. Weak-p inverter shift register chain plot showing SET cross-section dependence on LET.

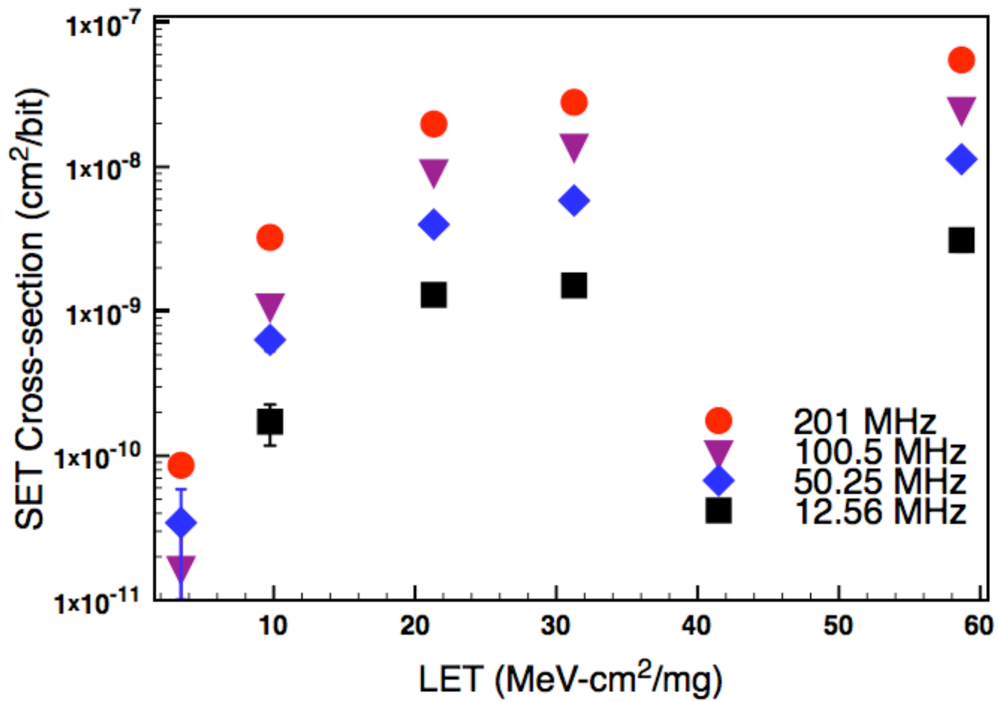

Figure 28. Matched current inverter shift register chain plot showing SET cross-section dependence on LET.

Figures 29, 30, and 31 plot the shift register data in a different way compared to the pervious plots in a plot of SET cross-section versus frequency. In Fig. 29, the baseline SET cross-section is shown and it can be seen there is little change in SET cross-section over frequency for any of the different ions. While in Figs. 30 and 31, as the energy of the ion is increased, the slope of the line increases too. The reason that the slope of the lines continues to increase with increasing ion LET is that higher LET ions create a greater number of SETs than lower energy ions and that many more SETs have pulse widths that last multiple clock cycles. Therefore, SETs are more readily detected as the ion energy increases.

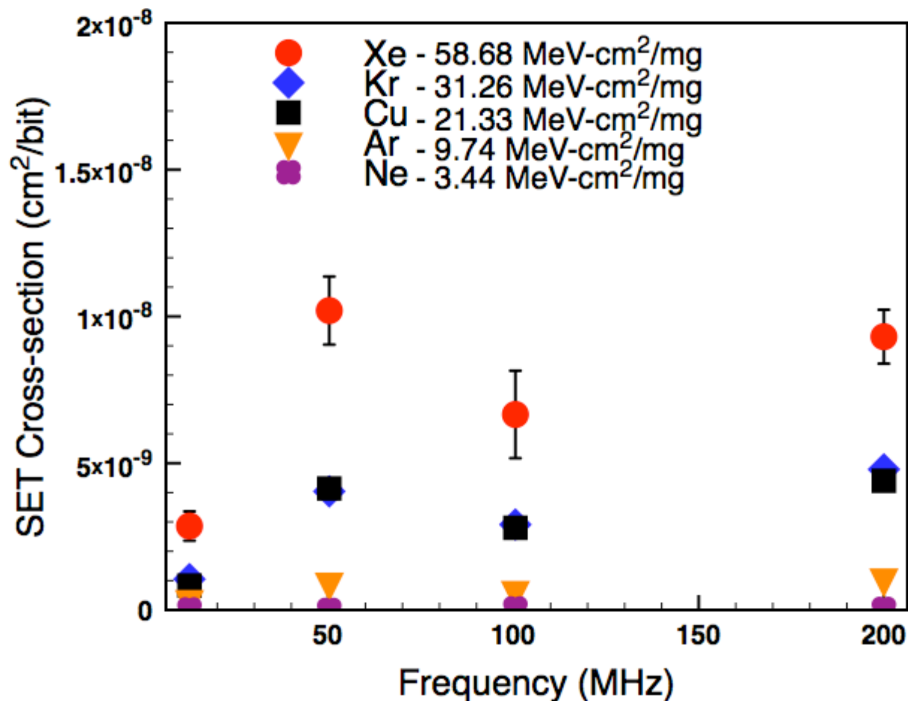

Figure 29. SET cross-section of the baseline shift register chain with varying clock frequency.

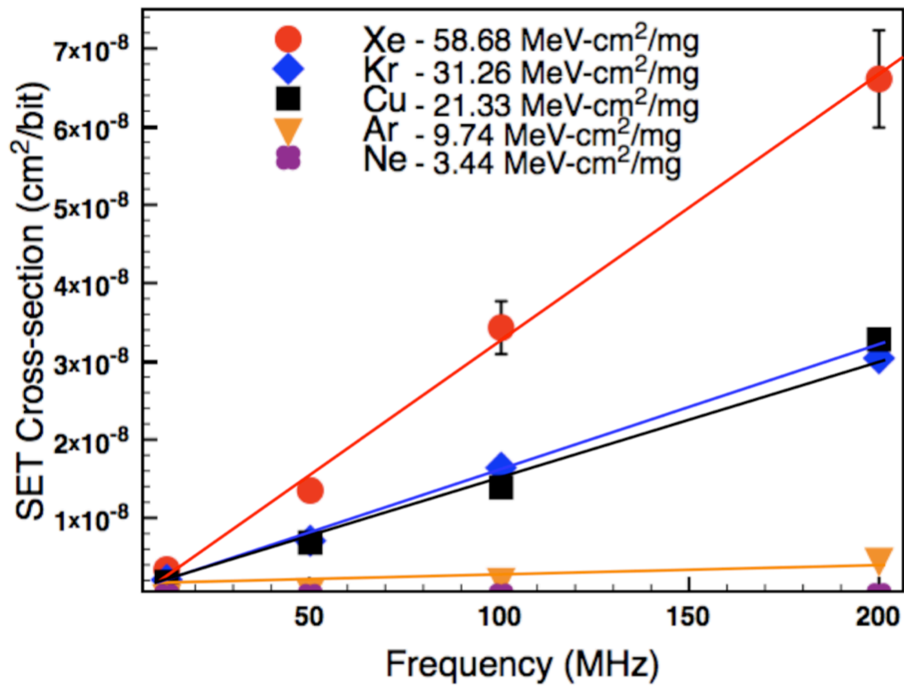

Figure 30. SET cross-section of the weak-p shift register chain with varying clock frequency.

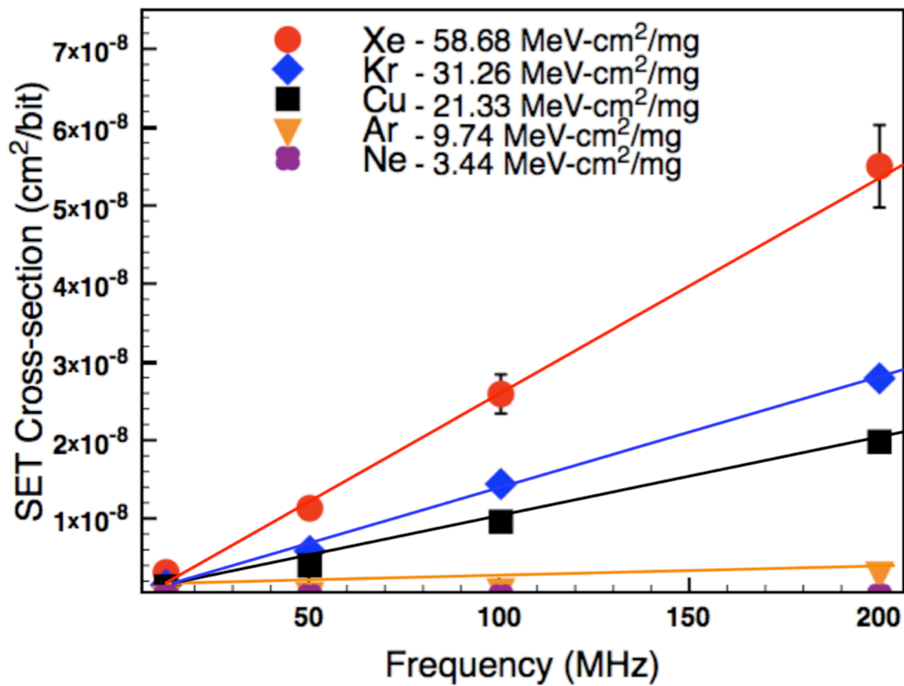

Figure 31. SET cross-section of the matched current shift register chain with varying clock frequency.

All of the plots previously displayed LET in some way, whether on the x-axis or as different data series. For all the data, there ideally exists an LET dependence as best shown in Figs. 26, 27, and 28. As the LET increases, the SET cross-section increases until it plateaus or saturates around 21 MeV-cm^2/mg showing that the amount of errors that can be generated within the circuit has reached its limit.

CHAPTER VII

CONCLUSION

In this thesis, a new approach of evaluating SET cross-section in combinational logic has been described. The C–CREST design was fabricated in the IBM 90 nm CMOS9SF process and successfully tested with exposure to heavy-ions. Data taken shows that the C-CREST approach works and provides an effective technique for characterizing SEE in various types of combinational logic while also agreeing with previously published data [2].

Traditional approaches to characterizing SEEs in combinational logic have been described along with their advantages and disadvantages. These approaches have suffered from being unable to balance increased combinational logic cross-section with increased propagation delay. This has made it difficult to test combinational circuits at maximum clock speeds for the particular process.

On the other hand, the C-CREST approach has been shown to test combinational logic at speed with the inclusion of a single XOR gate to minimize propagation delay between latches. The ability to separate the combinational logic elements and use static inputs makes the approach very versatile. Different logic families can be used along with different logic depths. Also, multiple static inputs can be used to allow a combinational logic design to be put into a certain state. The combinational logic can be only limited by the chip area, which gives the experimenter the ability to distinguish errors from the combinational logic and from the latches. Therefore, the C-CREST approach can easily be adapted to

multiple types of logic at their native speeds and can be implemented in a variety of processes.

APPENDIX A

SAMPLE VHDL PROGRAM

```
--------------------------------------------------------------------------------
-- Author:      DAB
-- History:
--  DAB           -- First version
--
-- ^^^^^^
-- DAB           -- 3/20/2007
--               Added WDT_Reset Wait state
-- ~~~~~~
-- DAB           -- 3/21/2007
--             -- Test37
--               Internal clock/64, External 0 (Constant), Block 4, Circuit 2, Comb. Inv Chain
--               Comb. Inv Chain
--
-- ******
--JRA            --6/07
--               --Modified to work without error flag

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.conv_std_logic_vector;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_signed.all;
use ieee.std_logic_unsigned.all;
use IEEE.std_logic_misc.all;

library Unisim;
use Unisim.all;

ENTITY dut_control IS

   port (
       clk                  : in  std_logic;
       dut_control_reset    : IN   std_logic;
       flag                 : IN   std_logic;
       flag_mask            : OUT  std_logic_vector(3 downto 0);
       autostop             : OUT  std_logic;
```

```
    csel            : OUT  std_logic_vector(2 downto 0);
    psel            : OUT  std_logic_vector(1 downto 0);
    roen            : OUT  std_logic_vector(1 downto 0);
    blocksel        : OUT  std_logic_vector(1 downto 0);
    cdels           : OUT  std_logic_vector(1 downto 0);
    pdels           : OUT  std_logic_vector(1 downto 0);
    prnsel          : OUT  std_logic;
    eclk            : OUT  std_logic;
    dut_reset       : OUT  std_logic;
    expat           : OUT  std_logic
   );

end entity dut_control;


-------------------------------------------------------------------------------
-- Architecture
-------------------------------------------------------------------------------

ARCHITECTURE implementation  OF dut_control IS

-------------------------------------------------------------------------------
-- Signal and Type Declarations
-------------------------------------------------------------------------------
signal  count           : unsigned (31 downto 0);
signal  max_count_sig   : std_logic;
signal  wdt_clk         : std_logic;
signal  WDT_rst         : std_logic;



-------------------------------------------------------------------------------
-- WDT_PROCESS

-------------------------------------------------------------------------------
-- State Machine
-------------------------------------------------------------------------------


type DUT_SM is (Idle,
         Start_test,
         wdt_reset,
         Wait_for_flag
         );

signal current_State        : DUT_SM := Idle;
```

```
signal next_State           : DUT_SM;



begin -- architecture IMP
-------------------------------------------------------------------------------
-- This process counts clocks after WDT_rst is asserted.
-- The completion of the count ~1,000,000 terminates the testbench.
-------------------------------------------------------------------------------
   WDT_PROCESS: process (wdt_clk, WDT_rst)

   variable MAX_COUNT  : unsigned(31 downto 0) := X"00002000";

   begin
      if wdt_clk'event and wdt_clk = '1' then
         if WDT_rst = '1' then
            count <= X"00000000";
            max_count_sig <= '0';
         else
            count <= count + 1;
            max_count_sig <= '0';
         end if;

         if count = MAX_COUNT then
         max_count_sig   <=  '1';
         end if;
      end if;

   end process WDT_PROCESS;


-------------------------------------------------------------------------------
-- Processes to connect only needed
-------------------------------------------------------------------------------

wdt_clk  <=  clk;


SM_COMB_PROCESS: process (current_State, flag, dut_control_reset, max_count_sig
                    ) is
begin
  -- Default outputs
      next_State   <= current_State;
      flag_mask    <= "0100";
      autostop     <= '0';
```

```
        csel         <= "100";
        psel         <= "00";
        roen         <= "00";
        blocksel     <= "11";
        cdels        <= "00";
        pdels        <= "00";
        prnsel       <= '1';
        eclk         <= '0';
        dut_reset    <= '0';
        expat        <= '0';
        wdt_rst      <= '1';


-- Next States
  case current_State is
    when IDLE          =>
        if  dut_control_reset = '1' then
           next_State   <= start_test;
        end if;

    when start_test   =>
           wdt_rst      <= '0';
           dut_reset    <= '1';
           autostop     <= '0';
           roen         <= "11";
           if max_count_sig = '1' then
              next_State   <= wdt_reset;
           end if;

    When wdt_reset  =>
           dut_reset    <= '1';
           autostop     <= '0';
           roen         <= "11";
           wdt_rst      <= '1';
           next_State   <= wait_for_flag;

    when wait_for_flag  =>
           wdt_rst      <= '1';
           dut_reset    <= '0';
           autostop     <= '1';

           roen         <= "11";
           if flag = '1' then
              next_state  <= start_test;
           end if;
```

```
  end case;
end process SM_COMB_PROCESS;


SM_SEQ_PROCESS: process (Clk,dut_control_reset) is
begin
  -- use asynch reset
  if dut_control_reset = '0' then
      current_State         <= IDLE;
  elsif Clk'event and Clk='1' then
      current_State         <= next_State;

  end if;
end process SM_SEQ_PROCESS;


end architecture implementation;
```

REFERENCES

[1] P. E. Dodd and L. W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," *IEEE Trans. Nucl. Sci.*, vol. 50, pp. 583–602, June 2003.

[2] S. Buchner and M. Baze, "Single-event transients in fast electronic circuits," in *IEEE NSREC Short Course*, 2001, pp. V-1–V-105.

[3] L. W. Massengill, "SEU modeling and prediction techniques," in *IEEE NSREC Short Course*, 1993, pp. III-1–III-93.

[4] T. C. May and M. H. Woods, "Alpha-Particle-Induced Soft Errors in Dynamic Memories," *IEEE Trans. Elec. Dev.,* vol. 26, no. 1, pp. 2-9, Jan. 1979.

[5] B. Narasimham, M. J. Gadlage, B. L. Bhuva, R. D. Schrimpf, L. W. Massengill, W. T. Holman, A. F. Witulski, X. Zhu, A. Balasubramanian, and S. A. Wender, "Neutron and Alpha Particle-Induced Transients in 90 nm Technology," *IEEE 46th International Reliability Physics Symposium,* Phoenix, 2008.

[6] Andrew T. Kelly, "Differential Analog Layout for Improved ASET Tolerance", M. S. Thesis, May 2007.

[7] D. Binder, E.C. Smith, and A. B. Holman, "Satellite anomalies from galactic cosmic rays," *IEEE Trans. Nucl. Sci.*, vol. 22, pp. 2675-2680, Dec. 1975.

[8] David G. Mavis and Paul H. Eaton. Soft Error Rate Mitigation Tecnniques for Modern Microcircuits. *40th Annual Reliability Physics Symposium Proceedings,* pages 216-225, 2002.

[9] R. A. Reed, M. A. Carts, P. W. Marshall, C. J. Marshall, S. Buchner, M. La Macchia, B. Mathes, and D. McMorrow, "Single-event Upset Cross Sections at Various Data Rates," *IEEE Trans. Nucl. Sci.,* vol. 43, no. 6, pp. 2862-2867, Dec. 1996.

[10] S. Buchner, K. Kang, D. Krening, G. Lannan, and R. Schneiderwind, "Dependence of the SEU Window of Vulnerability of a Logic Circuit on Magnitude of Deposited Charge," *IEEE Trans. Nucl. Sci.,* vol. 40, no. 6, pp. 1853-1857 Dec. 1993.

[11] M.P. Baze, and S.P. Buchner, "Attenuation of Single-event Induced Pulses in CMOS Combinational Logic," *IEEE Trans. Nucl. Sci.,* vol. 44, no. 6, pp. 2217-2223, Dec. 1997.

[12] Mathew John Gadlage, "Analysis of Single-event Transients in Modern Digital Microcircuits," M. S. Thesis, August 2004.

[13] L. W. Massengill and P. W. Tuinenga, "Single-Event Transient Pulse Propagation in Digital CMOS," *IEEE Trans. Nucl. Sci.* vol. 55, no. 6, pp. 2861-2871, Dec. 2008.

[14] S. Buchner, M. Baze, D. Brown, D. McMorrow, and J. Melinger, "Comparison of Error Rates in Combinational and Sequential Logic," *IEEE Trans. Nucl. Sci.* vol. 44, no. 6, pp. 2209-2216, Dec. 1997.

[15] P. Eaton, J. Benedetto, D. Mavis, K. Avery, M. Sibley, M. Gadlage, and T. Turflinger, "Single-event transient pulsewidth measurements using a variable temporal latch technique," *IEEE Trans. Nucl. Sci.*, vol. 51, no.6, pp. 3365–3368, Dec. 2004.

[16] M.J. Gadlage, R.D. Schrimpf, P.H. Eaton, J.M. Benedetto, D.G. Mavis, M. Sibley, K. Avery, and T.L. Turflinger, "Single-event transient pulse widths in digital microcircuits," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 6, pp. 3285–3290, Dec. 2004.

[17] P. Marshall, M. Carts, S. Currie, R. Reed, B. Randall, K. Fritz, K. Kennedy, M. Berg, R. Krithivasan, C. Siedleck, R. Ladbury, C. Marshall, J. Cressler, Niu Guofu, K. LaBel, and B. Gilbert. "Autonomous bit error rate testing at multi-gbit/s rates implemented in a 5AM SiGe circuit for radiation effects self test (CREST*)," IEEE Trans. Nucl. Sci.,* vol. 52, no. 6, pp. 2446-2454, Dec. 2005.

[18] T.Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron CMOS technology," *IEEE Trans. Nucl. Sci.*, vol. 43, no. 6, pp. 2874-2878, Dec. 1996.

[19] MOSIS Website. http://www.mosis.org.

[20] E.H. Cannon, D.D. Reinhardt, M.S. Gordon, and P.S. Makowenskyj, "SRAM SER in 90, 130, and 180 nm bulk and SOI technologies," in Proc. Int. Reliability Physics Symp., 2004, pp. 300-304.

[21] J.D. Black, R. L. Reed, D. A. Black, L. W. Massengill, B. L. Bhuva, O. A. Amusan, M. C. Casey, A. Balasubramanian, and J. R. Ahlbin, 2007 SEE Symposium, Long Beach Island, CA, April 2007.

[22] 88-Inch Cyclotron Website. http://cyclotron.lbl.gov.

[23] H.S. Kim, "Low Speed Digital Tester Hardware Manual," Oct. 2006.

[24] D. Black, "C-CREST Design & Testing Requirements," April 2007.

[25] E. L. Petersen, "Soft Error Results Analysis and Error Rate Prediction," in *IEEE NSREC Short Course*, 2008, pp. III-1–III-303.

[26] Paul Eaton Technical Presentation. Presented October 15, 2003 in Albuquerque, NM.

[27] J. D Black, A. L. Sternberg, M. L. Alles, A. F. Witulski, B. L. Bhuva, L. W. Massengill, J. M. Benedetto, M. P. Baze, J. L. Wert, and M. G. Hubert, "HBD Layout Isolation Techniques for Multiple Node Charge Collection Mitigation," *IEEE Trans. Nucl. Sci.,* vol. 52, no. 6, pp. 2536-2541, Dec. 2005.