

Actel ProASIC Field Programmable Gate Array Single Event Effects (SEE) High-Speed Test - Phase I

Melanie Berg – Principle Investigator MEI
HAK Kim, Mark Friendlich, Chris Perez, Christina Seidlick: MEI
Ken Label: NASA/GSFC

Test Dates: 5/2010 and 8/2010 Report Date: 12/2010

Table of Contents

1.	Introduction	5
2.	Background	5
3.	Probability of Error and Mitigation strategies	6
4.	Devices and Designs Tested	8
4.1	Global Routes	9
4.2	Shift Register Architectures (WSRs)	9
4.2.1	Functional Description	9
4.2.2	Combinatorial Logic and Sequential Logic Elements in the WSR	10
4.2.3	WSR to Tester Interface	11
4.2.4	Data Input Patterns for WSRs	12
4.2.5	WSR Output	12
4.2.6	WSR Expected Upsets	13
4.3	Counter Array	14
4.3.1	Counter Array Implementation	15
4.3.2	Counter I/O Interface and Expected Outputs	17
4.3.3	Counter Expected Upsets	18
4.3.4	Summary of Counter Array Test Evaluations	18
4.4	Hamming Code 3 Finite State Machine (H3FSM)	19
4.4.1	H3FSM Interface to Tester	23
4.4.2	H3FSM Expected Upsets	23
4.4.3	Summary of H3FSM Array Test Evaluations	24
5.	High Speed Digital Tester (LCDT) Test Vehicle	24
5.1	Architectural Overview	24
5.1.1	I/O List and Definitions	25
5.2	RS232 communication from the LCDT to the Host PC	28
5.3	RS232 communication From the Host PC to the LCDT	28

5.3.1	User GUI	28
5.3.2	User Interface and Command Control	31
6.	DUT Test Procedures	32
6.1	DUT1 WSR Testing	32
6.1.1	Dynamic: Evaluate susceptibility of WSR	32
6.2	Counter Array Tests	33
6.2.1	Dynamic: Evaluate susceptibility of Counter DFF cells in biased-dynamic states	33
6.3	H3FSM Array Tests	33
6.3.1	Dynamic: Evaluate susceptibility of H3FSM DFF cells in biased-dynamic states	33
7.	Processing the DUT Outputs	33
7.1	WSR, Counter, H3FSM SHIFT_CLK Processing	34
7.2	WSR Data Processing	35
7.2.1	WSR SEU Cross Section Calculations	36
7.3	Counter Array Data Processing	36
7.3.1	Counter Array Data Capture and compare	36
7.3.2	Counter Array Error Record	37
7.3.3	Counter Array SEU Cross Section Calculations	37
7.4	H3FSM Array Data Processing	38
7.4.1	H3FSM Array Data Capture and compare	38
7.4.2	H3FSM Array Error Record	38
7.4.3	Counter Array SEU Cross Section Calculations	39
8.	Heavy Ion Test Facility and Test Conditions	39
9.	Preliminary WSR Heavy Ion Test Results	40
9.1	Testing Difficulties	40
9.1.1	First Test Trip 05/2010	40
9.1.2	Second Test Trip 08/2010	40
9.1.3	Summary of Reprogramming issue	40
9.2	No-TMR	41
9.2.1	No-TMR WSR String	42
9.2.2	No-TMR Counters	44
9.3	LTMR SEU Cross Sections	46
9.3.1	Effectiveness of LTMR	47
9.3.2	LTMR and LET Threshold	48
9.4	SET Propagation with LTMR Designs	49
9.4.1	LTMR: SET Generation, Propagation, and Capture in RTAX-S Devices	49
9.4.2	LTMR Data Pattern Effects	52
9.4.3	LTMR Architectural Effects: Counters	53
9.5	Global Routes: DTMR and P_{SEFI}	56
9.5.1	DTMR Single Cycle Upsets	57
9.5.2	Bursts	68
9.6	Hamming Code 3 Finite State Machine (H3FSM)	70

10.	ConclusionS	76
10.1	Reprogramming Issues	76
10.2	WSR and Counters	76
10.3	H3FSM	77
10.4	Bursts	77
11.	Appendix 1:	77

Figure 1: Applied LTMR ... Only the DFFs are triplicated. Consequently data inputs to each DFF are shared and are single points of failure 7

Figure 2: Application of DTMR. All functional logic is triplicated except global routes (Clocks and Resets are not triplicated) 7

Figure 3: Windowed Shift Register (WSR) Design contains 6 chains with various levels of combinatorial logic between each DFF 9

Figure 4: WSR Internal Data Input Circuit 9

Figure 5: WSR Top Level Architecture 10

Figure 6: WSR Shift Register Strings with Optional Combinatorial Logic. All DFFs are connected to the same Clock Input and the same reset... i.e. clocks and resets are shared among all WSR DFFs. $N > 0$ DFFs are connected to a global Enable route controlled by the tester. 11

Figure 7: WSR shift register operation for a checker board input. Every 4 clock cycles the last 4 shift register bits are equivalent. Every 4 clock cycles the window gets a snap shot of the last 4 bits of the shift register. Consequently, the window is static under normal operating conditions 13

Figure 8: Example of WSR SEE DUT output to tester 14

Fig 9: Schematic of the 24-bit Counters and their Output Selection Logic. In this case, the output selection logic is a shift register (Shifts up counter values to the output registers every 4 cycles) 15

Fig 10: Counter Shift Register Cycles; Numbers in shift registers represent counter labels at a given moment in time. If there is an x with the shift register, then it is considered a "don't-care" state 16

Figure 11: Typical SEE Counter Outputs. Each output represents a value from a different counter in the array. Counter selection is sequential, hence, the counter number and the counter values all increment by 1 each PROASIC_Counter_Clk cycle. 18

Figure 12: Finite State Machine (FSM) Schematic. FSMs consist of a bank of registers that feed combinatorial next-state logic. 19

Figure 13: Finite State Machine (FSM) Schematic with Hamming Code 3 Error Detection and Error Correction. 20

Figure 14: 32 bit Finite State Machine. Encoding used was Hamming Code 3. 21

Figure 15: Hamming Code 3 Finite State Machine (H3FSM) Array Interface to Tester. Each H3FSM has a total of 9 bits. Subsequently, the interface to the tester is also 9 bits wide. 23

Figure 16: Picture of Low Cost Digital Tester (LCDT) connected to Device Under Test (DUT) at Texas A&M Heavy Ion Facility 24

Figure 17: System Level Tester Architecture. Two PCs are running a Labview GUI. One PC is running a logic analyzer to have real time processing of the DUT WSR or Counter Array outputs. The PC connected to RS232(1) sends commands to the LCDT that set test parameters and starts test operations 25

Figure 18: WSR GUI. Communicates with the Tester via the RS232(1) and TX232 (1) Interfaces. Commands are sent using this GUI. Command Echoes and WSR error reports are sent to this GUI from the Tester 29

Figure 19: Counter Labview Interface. Communicates with the Tester via the TX232 (1) Interfaces. Counter error reports are sent to this GUI from the Tester. 30

Figure 20:H3FSM GUI. Communicates with the Tester via the RS232(1) and TX232 (1) Interfaces. Commands are sent using this GUI. Command Echoes and H3FSM error reports are sent to this GUI from the Tester 31

Figure 21: Shift_CIK Capture consists of a Metastability Filter and a Edge Detect 34

Figure 22: Another look at a WSR string. The 4-bit window is the registers in the illustration that are underneath the shift register string. The last 4-bits of the string are shifted into window, once every 4 clock cycles. 35

Figure 23: WSR Error Record Data Fields. Each error record is prefaced with an error header (00 FA F3 21) when being sent from the LCDT to the Host PC. 35

Figure 24: DUT2 Counter Array Error Record. Cycle n represents capture cycles. Capture cycles are once every 4 LCDT tester clock cycles. 37

Figure 25: Counter Array Error Record. Cycle n represents capture cycles. Capture cycles are once every 4 LCDT tester clock cycles. 38

- Figure 26: No-TMR Log-linear curves at 100MHz for N=0 and N=8 Strings 41
- Figure 27: No-TMR at 100MHz with LET values $\geq 12 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. N=0 chains 42
- Figure 28: No TMR for WSR chains N=0 to N=8. 100MHz Operational Frequency Checkerboard Pattern. N=16 WSRs not in graph because they cannot operate at 100MHz. 43
- Figure 29: No TMR for all WSR chains: 50MHz Operational Frequency with Checkerboard Pattern. Due to time limitation, No TMR designs at LET $> 4.0 \text{ MeV}\cdot\text{cm}^2/\text{mg}$ were not tested. 43
- Figure 30: No TMR Counter bits. Bits are placed into bins of 4. Average Cross Section per bit-bin is illustrated 44
- Figure 31: No TMR Counter bits. Bits are placed into bins of 4. Average Cross Section per bit-bin is illustrated. At higher LET values, $P_{\text{SET}\rightarrow\text{SEU}}$ is more prevalent as shown by the variation in SEU cross sections across bits. 45
- Figure 32: 100MHz Checkerboard pattern LTMR SEU Cross Sections per LET. WSR N=0 and N=8 Chains are shown. 46
- Figure 33: ProASIC LTMR Details. Mitigation is susceptible to SETs because it contains non-redundant combinatorial logic prior to the data input of the LTMR DFFs. 47
- Figure 34: Comparison of 100MHz Checkerboard N=0 and N=8 WSRs with No TMR and LTMR 48
- Figure 35: LTMR Counter Array SEU Cross Sections at 80MHz operational frequency. Low LET values are observed. On-set or LET threshold has been increased over No-TMR. 49
- Figure 36: SET Propagation through Combinatorial logic gates 50
- Figure 37: $P_{\text{SET}\rightarrow\text{SEU}}$ with LET=75 for Various WSR Strings Containing Different Levels of Combinatorial Logic Cells. (N=0: No Combinatorial logic; N=8: 8 Levels of Combinatorial logic blocks between DFFs; N=20: 20 Levels of Combinatorial Logic Blocks between DFFs) 51
- Figure 38: $P_{\text{SET}\rightarrow\text{SEU}}$ with LET=28.7 for Various WSR Strings Containing Different Levels of Combinatorial Logic Cells. (N=0: No Combinatorial logic; N=8: 8 Levels of Combinatorial logic blocks between DFFs; N=20: 20 Levels of Combinatorial Logic Blocks between DFFs) 51
- Figure 39: $P_{\text{SET}\rightarrow\text{SEU}}$ with LET=8.6 for Various WSR Strings Containing Different Levels of Combinatorial Logic Cells. (N=0: No Combinatorial logic; N=8: 8 Levels of Combinatorial logic blocks between DFFs; N=20: 20 Levels of Combinatorial Logic Blocks between DFFs) 51
- Figure 40: 100MHz LTMR WSR Data Pattern Comparisons. Checkerboard versus Static 0 Pattern for N=0 Chains. Checkerboard has a significantly higher cross section over LET values. 52
- Figure 41: 100MHz LTMR WSR Data Pattern Comparisons. Checkerboard versus Static 0 Pattern for N=8 Chains. Cross sections are not significantly different, however, 0-pattern with buffers seem to consistently have higher cross sections than other strings across LET. 53
- Figure 42: Comparing No-TMR versus LTMR. Left portion of figure shows 80MHz Counters. Right most portion of figure shows 100MHz WSRs. 54
- Figure 43: No-TMR Versus LTMR: 80MHz Counters and 100MHz WSR Chains at LTMR $\text{LET}_{\text{TH}}=20.3 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. Counter SEU Cross sections are within the N=0 to N=8 SEU Cross section range 55
- Figure 44: LTMR: 80MHz Counters and 50MHz WSR Chains at LTMR $\text{LET}_{\text{TH}}=20.3 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. SEU Cross Sections of counters and WSR strings begin to approach each other. 55
- Figure 45: LTMR: 80MHz Counters and 100MHz WSR Chains at LTMR $\text{LET}_{\text{TH}}=53.1 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. Counter SEU Cross sections are within the N=0 to N=8 SEU Cross section range 56
- Figure 46: LTMR: 80MHz Counters and 100MHz WSR Chains at LTMR $\text{LET}_{\text{TH}}=53.1 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. SEU Cross Sections of counters and WSR strings begin to approach each other. 56
- Figure 47: DFF with Data Input (D), Clock Input (C), Enable Input (E), Reset $\text{\textcircled{R}}$, and Data Output (Q). 57
- Figure 48: 50MHz DTMR WSR SEU Cross Sections per LET. 58
- Figure 49: 50MHz Checkerboard WSR DTMR and LTMR Cross Sections at LET = $20.3 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. DTMR LET_{TH} near $20 \text{ MeV}\cdot\text{cm}^2/\text{mg}$ for checkerboard data pattern and for a design containing global Clocks, Resets, and Enables. 59
- Figure 50: 50MHz Checkerboard WSR DTMR and LTMR Cross Sections at LET = $53.1 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. DTMR WSR SEU Cross Sections occupy the left portion of the graph. LTMR occupy the right. 59
- Figure 51: 50MHz Checkerboard WSR DTMR and LTMR Cross Sections at LET = $106.2 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. 60
- Figure 52: 50MHz WSR DTMR Static 1 Data Pattern at $\text{LET}_{\text{TH}} = 20.3 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. The graph represents mostly upsets on Reset Pin. LET_{TH} was observed at $20.3 \text{ MeV}\cdot\text{cm}^2/\text{mg}$ for DTMR Static 1 pattern. 61
- Figure 53: 50MHz WSR DTMR Static 1 Data Pattern. Represents mostly upsets in I/O 62
- Figure 54: DTMR SEU Cross Sections at 50 MHz. LET_{TH} is close to $20 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. This is a great improvement over LTMR and No-TMR 63
- Figure 55: 1MHz DTMR WSR Checkerboard Pattern SEU Cross sections. SEU cross Sections seem to be slightly higher than 50MHz. LET_{TH} for the WSRs remains at approximately $20 \text{ MeV}\cdot\text{cm}^2/\text{mg}$. 64

- Figure 56: DTMR Data Pattern Comparison at $LET_{TH} = 20.3 \text{ MeV*cm}^2/\text{mg}$. 50MHz Static 0 Pattern is not shown in graph because no upsets were observed until an $LET = 75\text{MeV*cm}^2/\text{mg}$ 65
- Figure 57: Comparison between Checkerboard WSR and LSB bits0-3 of the Counters. SEU Cross Sections are not significantly different with DTMR insertion. DTMR upsets rely on global nets: Clocks, Resets, and global enables. 66
- Figure 58: 8MHz Counter-bit DTMR Cross Sections. LET_{th} for the Counters was at $53\text{MeV*cm}^2/\text{mg}$. There is no enable pin connected to the counters. Hence upsets are local resets or clocks 67
- Figure 59: 80MHz Counter-bit DTMR Cross Sections. LET_{th} for the Counters was at $53\text{MeV*cm}^2/\text{mg}$. There is no enable pin connected to the counters. Hence upsets are local resets or clocks 67
- Figure 60: 100MHz WSR LTMR Bursts per Chain. Differences in Burst Cross Sections are insignificant 68
- Figure 61: 50MHz WSR LTMR Bursts per Chain. Differences in Burst Cross Sections are insignificant 69
- Figure 62: 1MHz WSR LTMR Bursts per Chain. Differences in Burst Cross Sections are insignificant 69
- Figure 63: 100MHz WSR No-TMR Bursts per Chain. Differences in Burst Cross Sections are insignificant per LET 70
- Figure 64: H3FSM potential upsets and EDAC masking of upsets. EDAC Combinatorial logic becomes the most significant contributor to SEU cross sections at lower LET values 71
- Figure 65: H3FSM Cross Sections for Single bit upsets and multiple bit upsets 72
- Figure 66: 80MHz Comparison between H3FSM, Counter No-TMR, and Counter LTMR. It is shown that LTMR provides a lower overall SEU cross section and reduces LET_{TH} . H3FSM slightly reduces the SEU cross sections versus No-TMR Counters. However, LET_{TH} does not change. 73
- Figure 67: Comparison of 80MHz H3FSM Single bit upset SEU Cross sections vs. Multiple Bit SEU Cross sections. Multiple bit upsets are mostly non-global route related. All cross sections are single cycle. They are generated due to incorrect next-state calculation. 73
- Figure 68: Comparison of 80MHz and 8MHz H3FSM Single bit upset SEU Cross sections vs. Multiple Bit SEU Cross sections. Frequency effects exist. No test runs were performed for 8MHz with $LET < 8.6 \text{ MeV*cm}^2/\text{mg}$ 74
- Figure 69: 80MHz H3FSM SEU Cross Sections at Lower LET values across the 5 state bits. 75
- Figure 70: 80MHz H3FSM SEU Cross Sections at Higher LET values across the 5 state bits. 75

1. INTRODUCTION

This study was undertaken to determine the single event destructive and transient susceptibility of various designs implemented in the ProASIC FPGA family of devices. The ProASIC is considered the Device Under Test (DUT). The DUTs were monitored for Single Event Transient (SET) and Single Event Upset (SEU) induced faults by exposing them to a heavy ion beam. .

2. BACKGROUND

Following convention, bit-error rates (dE_{bit}/dt) pertaining to utilized DFFs are generally applied to characterize the SEE response of a FPGA. Also following convention, dE_{bit}/dt has been calculated monitoring the response of shift register strings during radiation testing. As a result, designers extrapolate shift register error rates ($dE_{bit}(fs)/dt$) to calculate error rates (dE/dt) of complex circuits as noted in equation 1.

$$\frac{dE}{dt} < \frac{dE_{bit}(fs)}{dt} * (\#UsedDFFs) \quad (1)$$

Shift registers are considered to have a linear data path because each node only has one input and one output (i.e. fanout = fanin = 1). Due to the decrease in transistor geometries and capacitive node loading it has come to question if the calculated shift register dE_{bit}/dt can be applied to complex circuits. In other words, will the predicted system error rate be accurate using shift register data as parameters? As an example, counter architectures are not linear. They contain nets with fan-out and fan-in >1. The fan-out will change both the capacitive loading of cells and the utilization of routing resources within FPGA fabrics. Depending on the rise/fall time and width of a Single Event Transient (SET) the capacitive loading of a cell can filter away the SET. On the other hand, a SET that is not filtered can fan-out to multiple nodes and have the effect of a multiple bit upset. Regarding nodes that have fan-out=1 or have minimal capacitive loading, it has been proven that SETs can increase in width as they traverse a circuit. As noted, variations of SET signatures exist and will depend on design topology and operational parameters.

Understanding the various upset event probabilities and their effects are essential when designing critical applications and predicting error rates. As a response, a more in-depth approach to SEE characterization of complex circuits has been performed by NASA Goddard Radiation Effects and Analysis Group (REAG). The study incorporates testing shift registers, counters, and finite state machines.

3. PROBABILITY OF ERROR AND MITIGATION STRATEGIES

The Actel PROASIC is a family of high performance, commercial grade, flash based, Field Programmable Gate Arrays (FPGAs). It has been shown that SEE upset rates pertaining to FPGA devices have 4 major probability components:

$P_{\text{configuration}}$: Probability that the FPGA configuration can upset

P_{DFFSEU} : Probability that a DFF can incur an upset

$P_{\text{SET} \rightarrow \text{SEU}}$: Probability that a DFF can capture a Single Event Transient (SET)

P_{SEFI} : Probability that a Single Event Functional Interrupt (SEFI) can occur. This document includes global routes such as Clocks, Resets, and global Enables as SEFIs (e.g. signals with significantly large fanout directly connected to DFFs input pins: C (clock), R (reset), or E (enable)). The probability that a design can incur an error is $P(fs)_{\text{error}}$ and is shown in Equation(2).

$$P(fs)_{\text{error}} \propto P_{\text{Configuration}} + P_{\text{DFFSEU}} + P_{\text{SET} \rightarrow \text{SEU}} + P_{\text{SEFI}} \quad (2)$$

Because the ProASIC configuration is flash-based, the configuration is considered hard regarding radiation environments ($P_{\text{configuration}} \approx 0$). In other words, the configuration's SEU rate is low to null. However, the devices are commercial and do not contain Radiation Hardened by Design (RHBD) Circuits. Hence, a design without mitigation will have substantial P_{DFFSEU} and $P_{\text{SET} \rightarrow \text{SEU}}$ upset components. In order to investigate the device and the effectiveness of applied mitigation strategies, common SEU test structures were implemented such as: WSRs and Counter Arrays [2]. All designs tested have different versions of applied mitigation. Application of mitigation was performed using the [Mentor Graphics Precision Rad-Tolerant](#) Synthesis tool set[4]. Mitigation strategies were:

None: no additional circuitry is added to the design pertaining to SEU mitigation

LTMR: Localized Triple Modular Redundancy. Only DFFs are triplicated. Combinatorial logic paths, Clocks, and resets are shared and consequently single sources of failure. With this mitigation strategy, only P_{DFFSEU} is reduced. Subsequently, transients can be captured by their destination DFFs ($P_{\text{SET} \rightarrow \text{SEU}}$) and Global routes can cause Single Event Functional Interrupts (P_{SEFI}). Figure 1 is an illustration of applied LTMR.

DTMR: Distributed Triple Modular Redundancy. The entire design is triplicated except for global routes (clocks, resets, and high fanout enables). This mitigation strategy reduces P_{DFFSEU} and $P_{\text{SET} \rightarrow \text{SEU}}$. However, since the global routes are not mitigated, then P_{SEFI} still exists. Figure 2 is an illustration of the application of DTMR. REAG chose the option to not triplicate the I/O of the DTMR designs. Hence the all I/O stem from DFFs and are not driven by combinatorial logic (besides the I/O buffer).

Global Triple Modular Redundancy (GTMR – everything is triplicated included clock domains) was not examined because this scheme requires at least three separate global clock trees to have less than 500ps of skew from each other. The ProASIC clock trees have minimal skew within a clock tree but contain too much skew (for GTMR purposes) between separate clock trees.

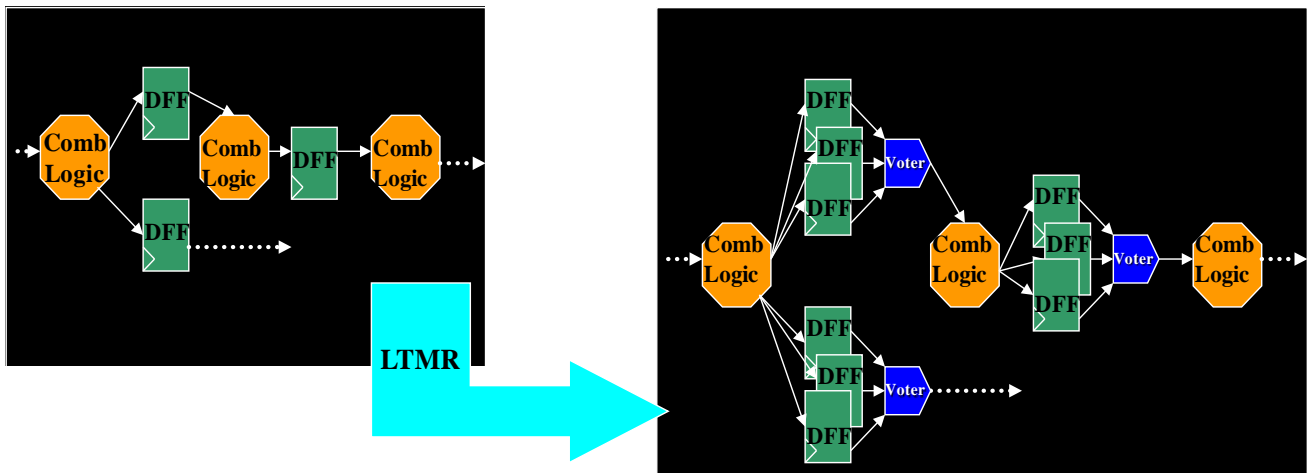


Figure 1: Applied LTMR ... Only the DFFs are triplicated. Consequently data inputs to each DFF are shared and are single points of failure

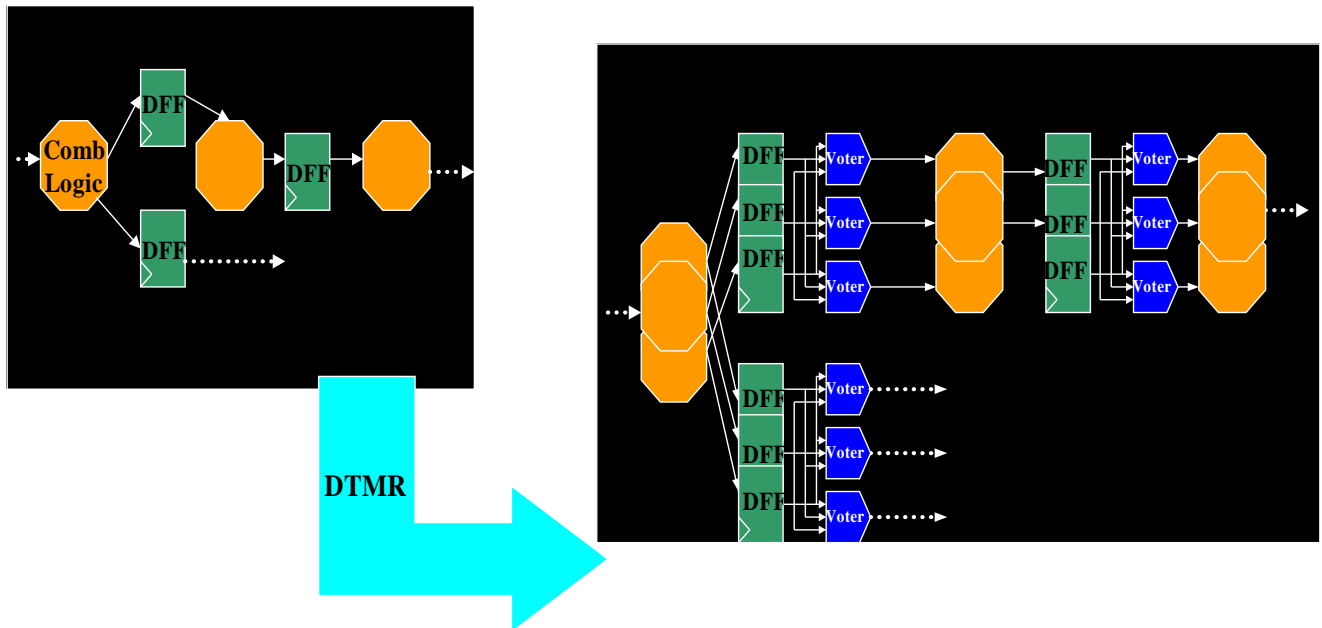


Figure 2: Application of DTMR. All functional logic is triplicated except global routes (Clocks and Resets are not triplicated)

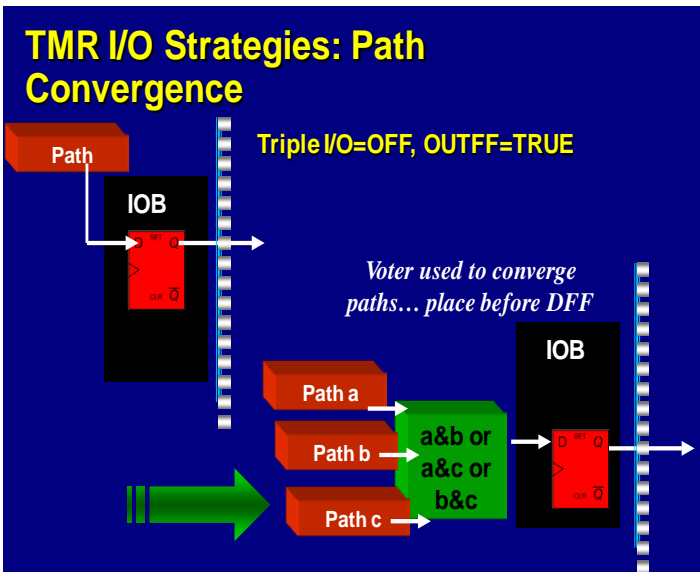


Figure 3: DTMR I/O Selected Solution. Triplicated paths converge into one DFF I/O.

4. DEVICES AND DESIGNS TESTED

There were various designs tested within 11 ProASIC devices. The sample size of devices (in this case) is not the focus since they are production- high speed parts with very little variation across the CMOS process. The emphasis was to test variations over the design state space. The devices were manufactured on an advanced 0.13 μ m, 7-level metal CMOS Process enhanced with Flash Technology. The manufacturer is Actel. The DUT Lot Date Codes and device markings are as follows.

ProASIC3E
 APE3000
 PQG208 0832
 QHJ3T

There were 3 major designs that were tested: Windowed Shift Registers (WSR) ,Counters, and Hamming Code 3 Finite State Machine (H3FSM). The following is a summary of the three designs:

Shift Registers: 6 Windowed Shift Register (WSR) Chains with varying numbers of inverters have been implemented.

No mitigation: $P_{\text{DFFSEU}} + P_{\text{SET} \rightarrow \text{SEU}} + P_{\text{SEFI}}$

LTMR: $P_{\text{SET} \rightarrow \text{SEU}} + P_{\text{SEFI}}$

DTMR: P_{SEFI}

Counter Arrays (CA): It should be emphasized that the design implemented to test the counters is a novel approach. In this study, there exist 100 counters (labeled 0 to 99) in the counter array. The LCDT can determine if an upset occurs in any of the counters and what bits were affected. A more detailed description will be provided in the following section.

No mitigation: $P_{\text{DFFSEU}} + P_{\text{SET} \rightarrow \text{SEU}} + P_{\text{SEFI}}$

LTMR: $P_{\text{SET} \rightarrow \text{SEU}} + P_{\text{SEFI}}$

DTMR: P_{SEFI}

Hamming Code 3 Finite State Machines (H3FSM): 5 bit state machine containing redundant 4-bit hamming code. There are a total of 32 states in the tested H3FSM (2^5). All states have a hamming distance of 3. No TMR mitigation was applied to this design because the design is considered to be self mitigated. A more detailed description will be provided in the following section.

4.1 Global Routes

All of the designs for radiation testing utilized the special global routing networks provided by the ProASIC device. There are two types of global networks: High-Drive, Low Skew Clocks, and high-drive minimal skew nets. Global Clocks can only be connected to the clock-pin of a DFF. This design rule keeps the Clock tree balanced and subsequently keeps the skew minimized. The alternate high-drive global routes are used for resets and high fan-out enables. The susceptibility of the global routes is under investigation.

The following sections provide more detailed descriptions for each implemented design.

4.2 Shift Register Architectures (WSRs)

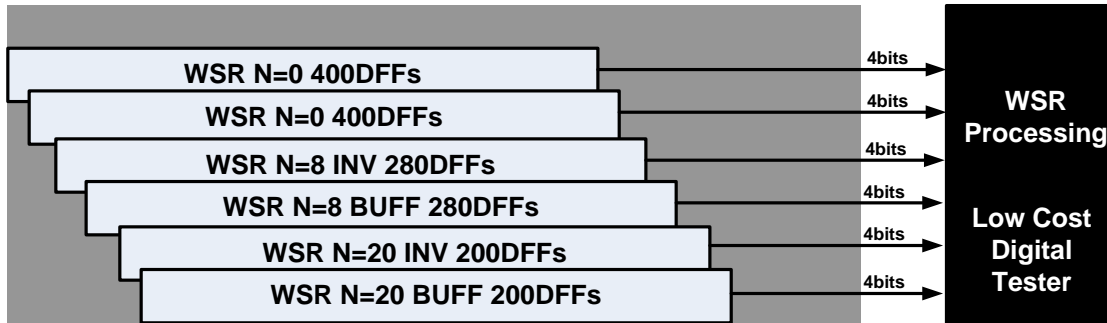


Figure 4: Windowed Shift Register (WSR) Design contains 6 chains with various levels of combinatorial logic between each DFF

4.2.1 Functional Description

In order to examine basic gate (sequential and combinatorial logic) sensitivity, REAG has chosen a simple (yet enhanced) shift-register architecture as one of the FPGA DUT architecture for radiation testing. Data input to the shift registers is generated inside of the DUT. However, 2-bit selection is controlled by the tester. The tester selects between a constant 0-pattern, a constant 1-pattern, and an alternating checkerboard pattern. The selection architecture is illustrated in Figure 5.

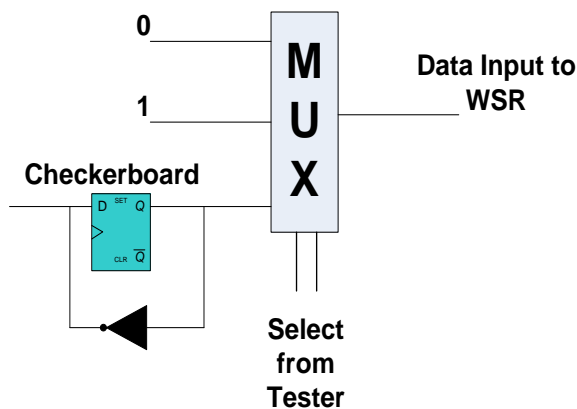


Figure 5: WSR Internal Data Input Circuit

The basic shift-register was enhanced by two features: (1) variations of inverter logic between flip-flop stages and (2) Windowed shift-register. The architecture is illustrated in Figure 6. The implementation of the windowed shift-register allows for reliable high-frequency testing by increasing board level signal integrity and simplifying DUT shift-register output data capture.

The following is the DUT configuration schematic:

DUT Top Level Architecture

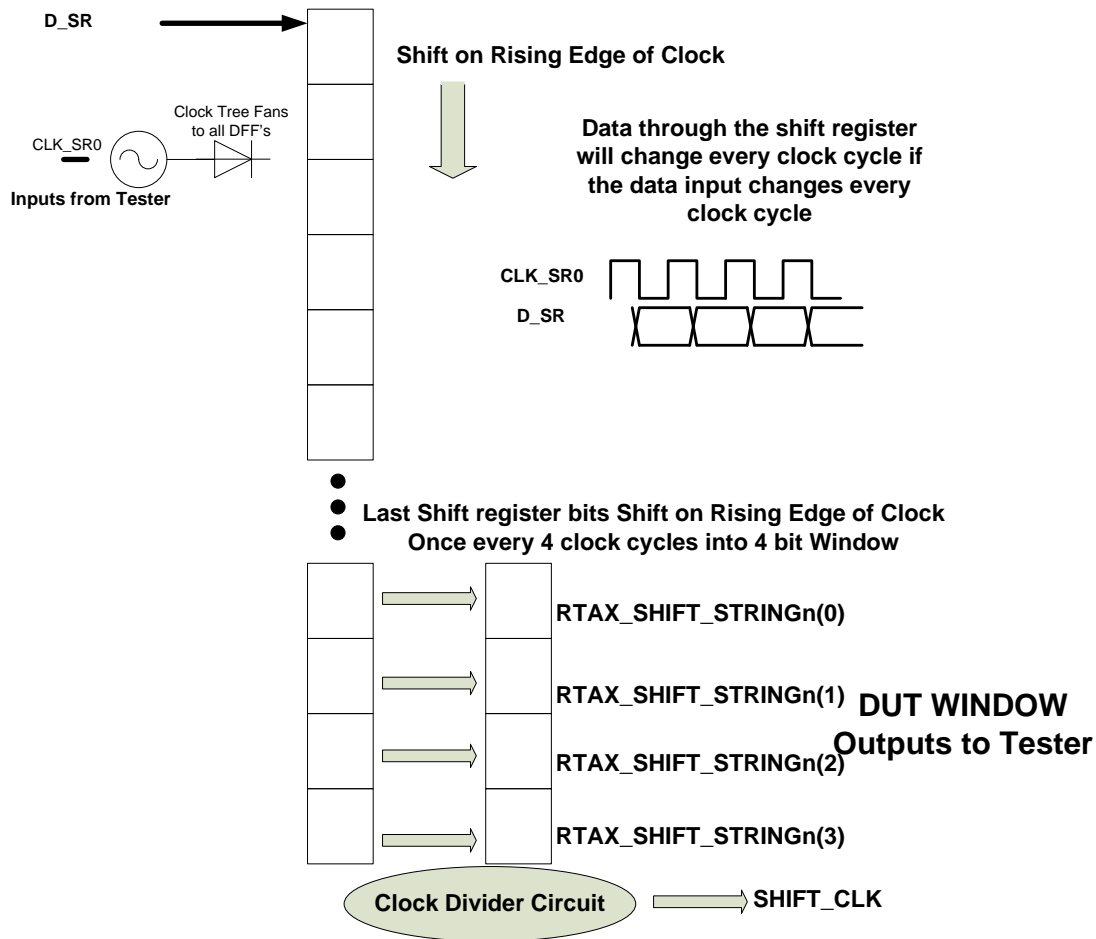


Figure 6: WSR Top Level Architecture

4.2.2 Combinatorial Logic and Sequential Logic Elements in the WSR

The Principle Configuration of the WSRs contains 200 to 2000 DFF's with varying levels of combinatorial logic (0, 2, 4, 8, 16, or 20 inverters) between DFF's. A By-4 clock divider circuit is implemented to shift the last 4 bits of the Shift register string into a DFF window (PROASIC_SHIFT_STRINGn). The window is output to the tester. A data clock (PROASIC_SHIFT_CLK) is also output to the tester for high speed synchronous data capture.

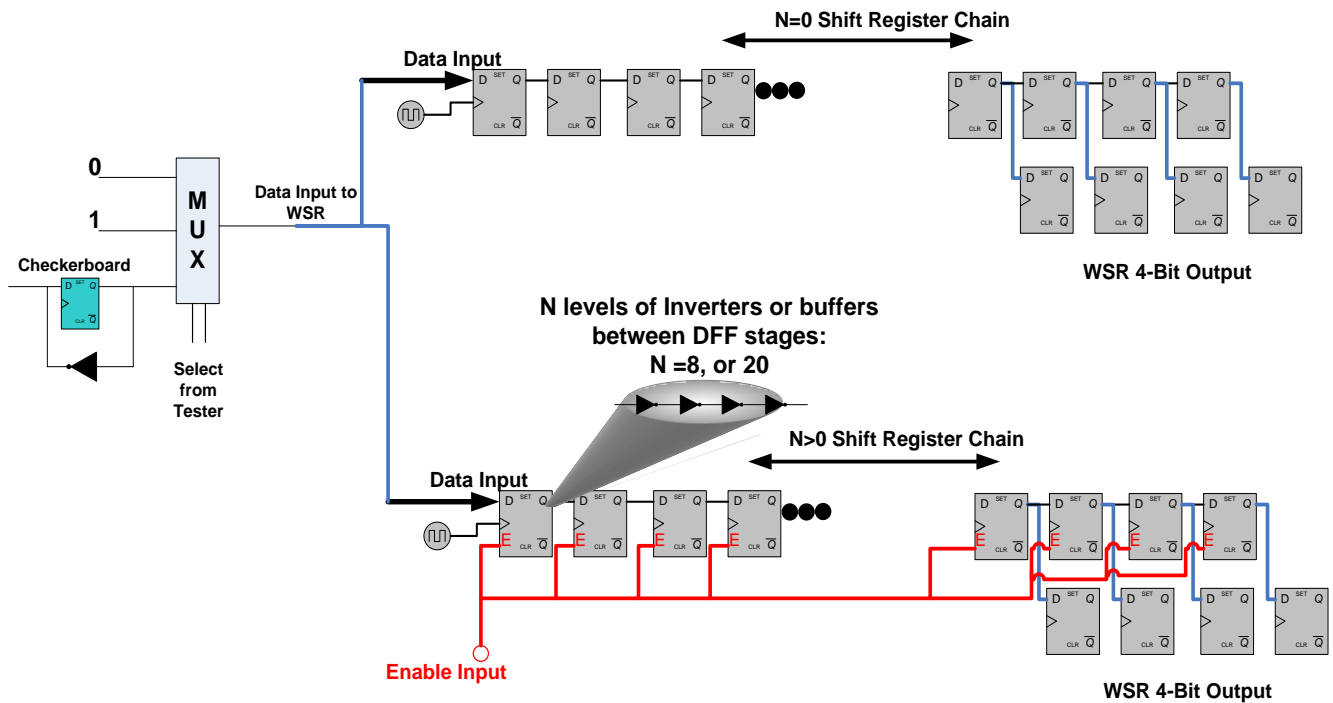


Figure 7: WSR Shift Register Strings with Optional Combinatorial Logic. All DFFs are connected to the same Clock Input and the same reset... i.e. clocks and resets are shared among all WSR DFFs. N>0 DFFs are connected to a global Enable route controlled by the tester.

Various levels of combinatorial logic are used in order to measure possible transient susceptibility. If the ProASIC device is susceptible to transients, then faults will be frequency dependent. Table 1 lists the logic resources contained within each shift register chain.

Table 1: Detailed Breakdown of WSR Chain Elements

Channel	0	1	2	3	4	5
Combinatorial logic	None	None	Inverters	Buffers	Inverters	Buffers
# Inv/Buf	0	0	8	8	20	20
#DFF	400	400	280	280	200	200
Global Enable	No	No	Yes	Yes	Yes	Yes

4.2.3 WSR to Tester Interface

Pertaining to Table 1 and Table 2, each WSR chain is supplied a clock (CLK_SR0) and reset (CLR_SR0) from the tester. Based on a pattern select input (PROASIC_PATTERN) to the DUT (output from tester), the DUT will calculate the desired WSR data input. All WSR chains except Chain0 and Chain1 (N=0 chains), use global enables. Enables were implemented for high frequency tests. WSRs with N>0 cannot

operate as fast as N=0. Hence, they are disabled during high speed N=0 Tests (160MHz). For the same reasoning, N=8 chains have a separate global enable than the N=20 chains.

Table 2: WSR Interface. I/O direction is with respect to Tester.

I/O Name	Bits	Dir wrt to DUT	Description
PROASIC_SHFT_CLK0	1	OUT	Chain 0 shift Clock
PROASIC_SHFT_CLK1	1	OUT	Chain 1 shift Clock
PROASIC_SHFT_CLK2	1	OUT	Chain 2 shift Clock
PROASIC_SHFT_CLK3	1	OUT	Chain 3 shift Clock
PROASIC_SHFT_CLK4	1	OUT	Chain 4 shift Clock
PROASIC_SHFT_CLK5	1	OUT	Chain 5 shift Clock
PROASIC_SHIFT_STRING0	1	OUT	4-bit windowed output of chain 0
PROASIC_SHIFT_STRING1	1	OUT	4-bit windowed output of chain 1
PROASIC_SHIFT_STRING2	4	OUT	4-bit windowed output of chain 2
PROASIC_SHIFT_STRING3	4	OUT	4-bit windowed output of chain 3
PROASIC_SHIFT_STRING4	4	OUT	4-bit windowed output of chain 4
PROASIC_SHIFT_STRING5	4	OUT	4-bit windowed output of chain 5
PROASIC_PATTERN	4	OUT	Each chain shares the 2 bits from this vector: The bits select which data input pattern (all 0's, 1's, or checker board). See Table 3 for selection details
PROASIC_EN0	4	OUT	Sent to chain enable or disable chain 2 and 3 (N=8 chains)
PROASIC_EN1	4	OUT	Sent to chain enable or disable chain 4 and 5(N=20 chains)
CLK_SR_TMR0	4	IN	Clock to all of the WSR circuitry
CLR_SR_TMR0	2	IN	Reset to all of the WSR circuitry

4.2.4 Data Input Patterns for WSRs

The possible shift-register data patterns are static-0, static-1, or checkerboard. The selection is controlled by a user command prior to each test. The pattern selection is 2 bits per DUT chain.

Table 3: Shift Register Data Pattern Select Using the PROASIC_PATTERN Tester Output

2 bit Pattern Selection	D_SR
00	static-0
01	static-1
10 or 11	Checkerboard

The PROASIC_EN0 and PROASIC_EN1 control signals are used to disable the shift registers that contain a significant amount of inverters between DFFs while performing high speed tests (up to 160MHz).

4.2.5 WSR Output

Every ProASIC_SHFT_CLKn cycle, ProASIC_SHIFT_STRINGn is evaluated to determine if an SEE occurred. For a data pattern of all 0's, the output will be all 0's. For a data pattern of all 1's, the output will be all 1's (after the equivalent number of clock cycles as the length of the shift register). For a checkerboard pattern,

the last 4 bits change every clock cycle. Because the WSR window is a snapshot of the last 4 shift register bits every 4 clock cycles, the window stays static (either a hex 5 or a hex A). The operation is illustrated in Figure 8.

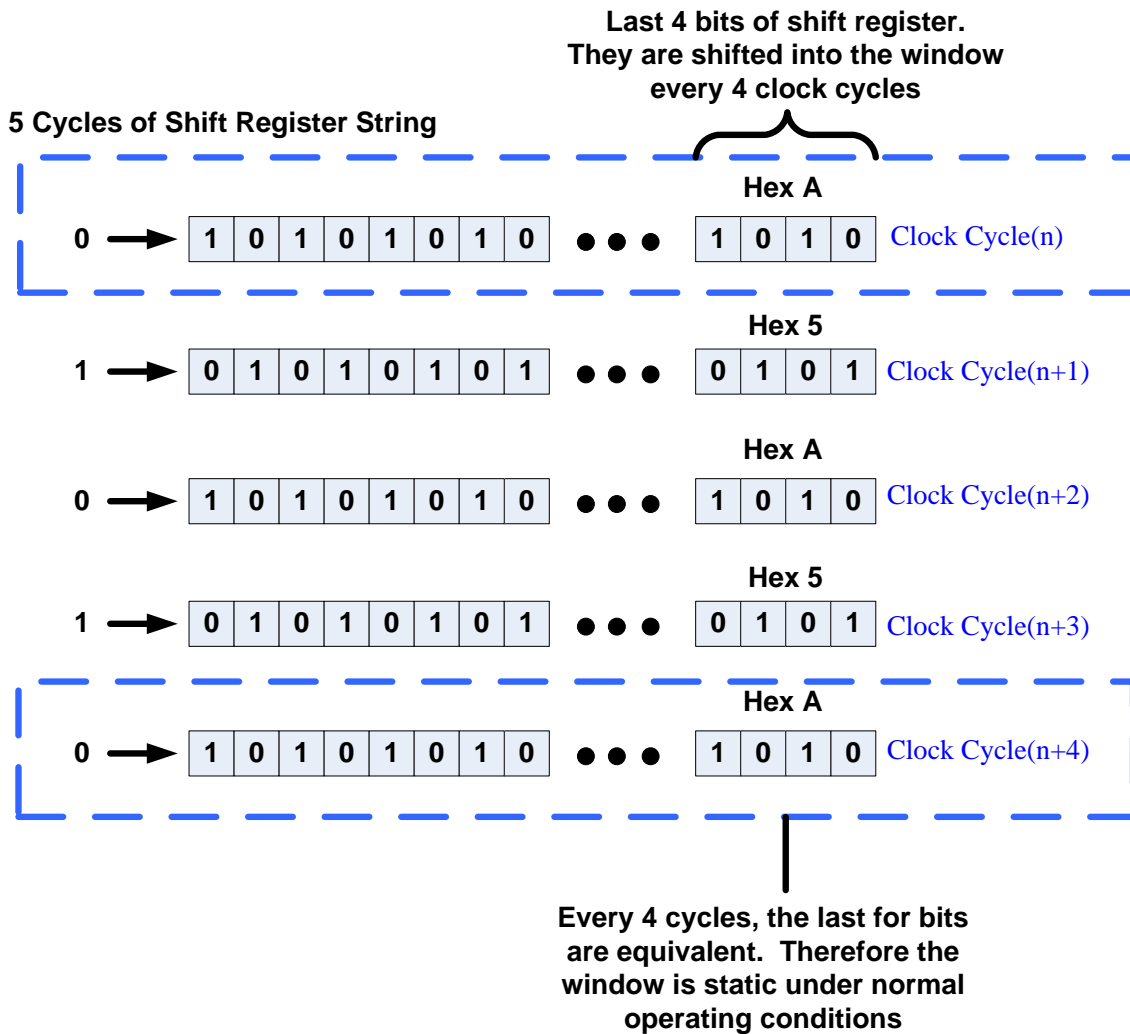


Figure 8: WSR shift register operation for a checker board input. Every 4 clock cycles the last 4 shift register bits are equivalent. Every 4 clock cycles the window gets a snapshot of the last 4 bits of the shift register. Consequently, the window is static under normal operating conditions

4.2.6 WSR Expected Upsets

Because of the WSR structured, the string outputs are expected to be constant after the length of the string cycles following reset de-assertion. Therefore, an error is easily detected by monitoring any change within the WSR outputs as illustrated in Figure 9: Example of WSR SEE DUT output to tester.

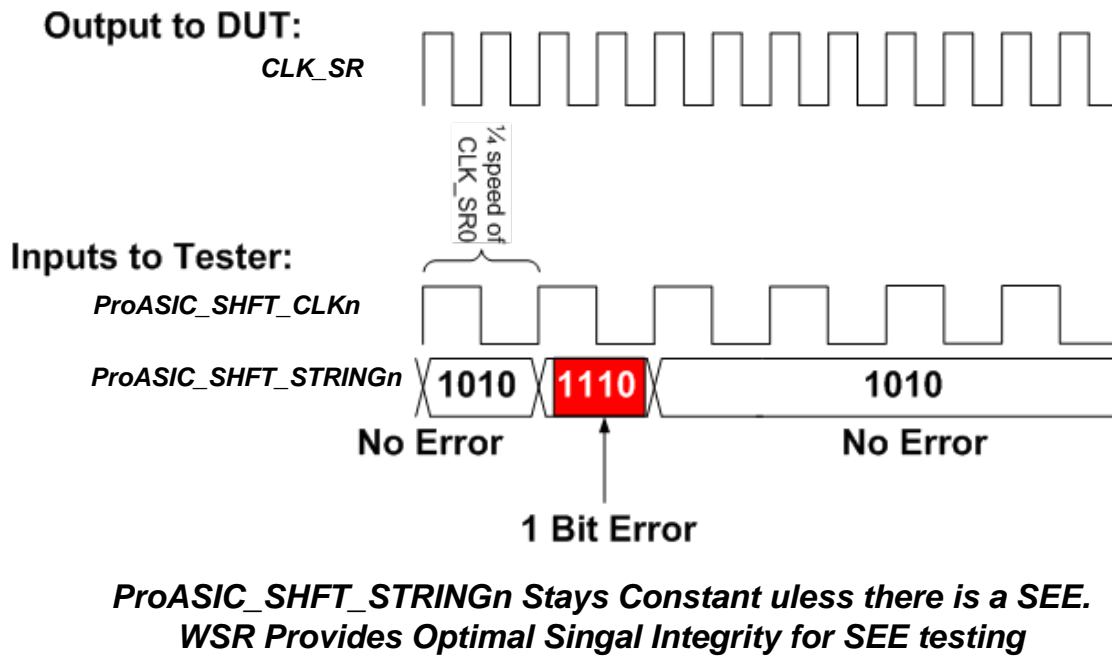


Figure 9: Example of WSR SEE DUT output to tester

Primary Expected WSR Upsets:

- Bit flip in shift register: Will be observed in the window for 4 cycles (because window can only change once every 4 cycles).
- Bit flip in window: Upset will be observed for less than 4 clock cycles
- Output transient: May not be able to distinguish from bit flip in window. However, the window be upset for less than one cycle.
- Global routes: An upset can occur in the clock or reset circuitry or enable circuitry (4 out of the 6 strings have enables).
- Shift_clks can get disrupted or completely stop.

4.3 Counter Array

During SEU testing, it would be ideal to be able to monitor every element of a complex design for every cycle. This is generally not feasible because it would require a DUT output to the test vehicle for every observable node. Therefore, more creative designs and interfaces must be developed such that operation during irradiation is unrestricted (fast, continuous, and unobstructed) yet node observation is maximized. Just as important, the tester must be fast enough and robust enough to capture and process the data supplied by the DUT. Processing integrity is very important. Dropped or incorrectly processed data can drastically change error cross sections.

For the SEU testing of the ProASIC Counters, a simple yet effective interface was developed.

4.3.1 Counter Array Implementation

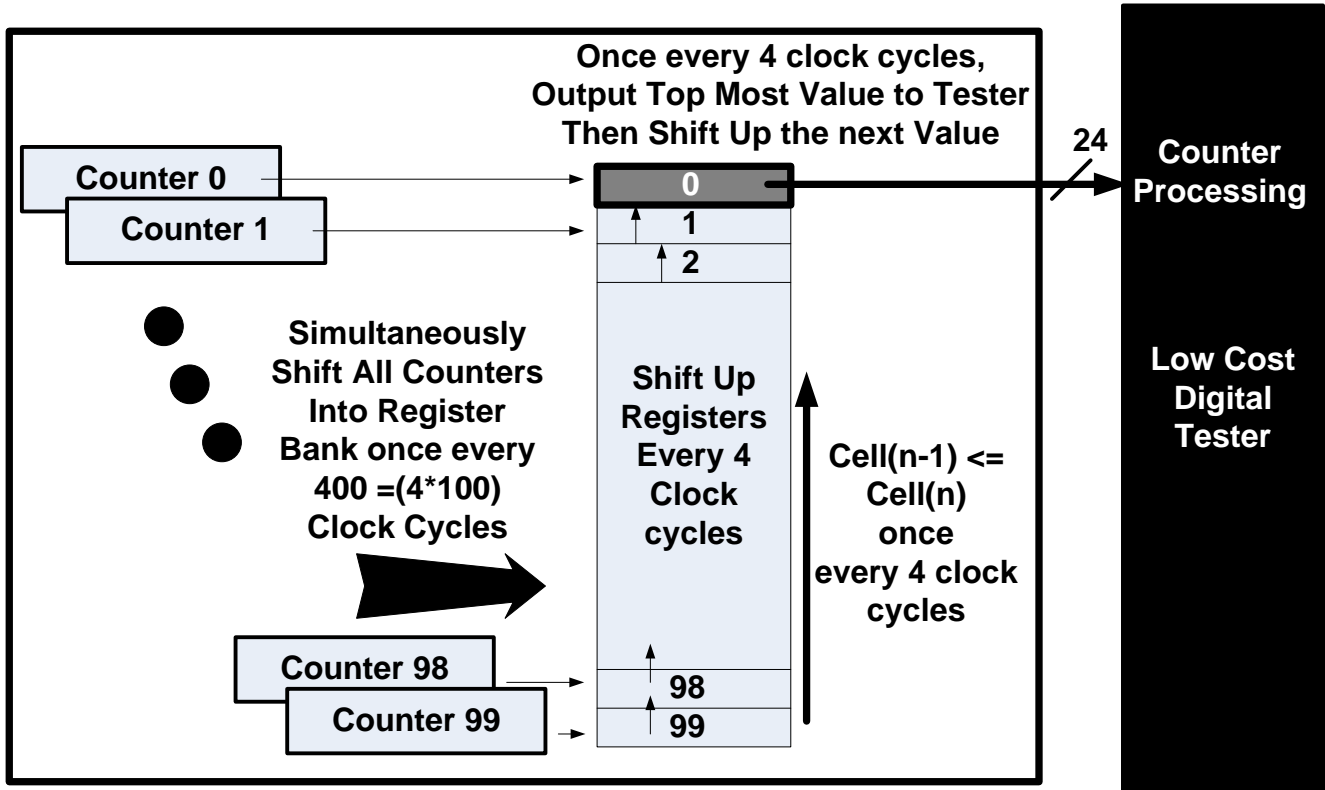


Fig 10: Schematic of the 24-bit Counters and their Output Selection Logic. In this case, the output selection logic is a shift register (Shifts up counter values to the output registers every 4 cycles)

The counter array developed by REAG is illustrated in Fig 10. The array contained 100 counters that were 24 bits wide. Because it is impossible to simultaneously output 100 by 24 bits, requiring 2400 outputs, an output scheme had to be employed that would not compromise the number or speed of the counters yet ensure that each counter is an observable node. Conventional thinking would suggest employing a multiplexer that sequences through the array and selects one of the 100 counters to be output at a time. Unfortunately, the function of the multiplexer, selecting 100 items, requires many levels of combinatorial logic which can be problematic during radiation testing and will slow down the operation of the circuit. Such a large block of logic can potentially mask the primary objective which is characterizing counter SEU susceptibility. Therefore, a novel output methodology had to be established.

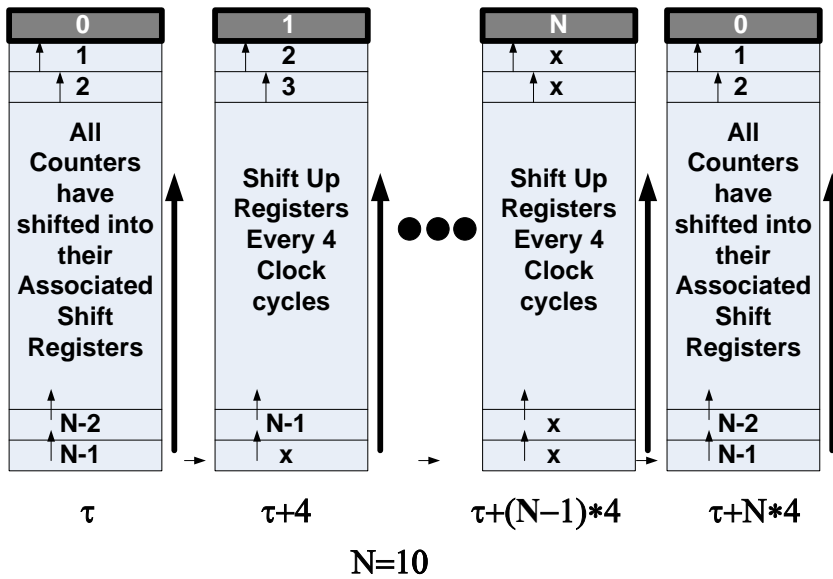


Fig 11: Counter Shift Register Cycles; Numbers in shift registers represent counter labels at a given moment in time. If there is an x with the shift register, then it is considered a “don’t-care” state

As an alternative, a “snap-shot” solution was implemented. With this methodology, each counter is captured simultaneously at a given time into a bank of registers. The number of registers is equivalent to the number of counters (i.e. each counter has its own snapshot register). This is illustrated in Fig 10. The top of the register bank (register 0) is the only register that is accessible by the tester and is 24 bits wide. Subsequently, the DUT to tester interface is simplified.

Fig 10 and Fig 11 illustrate the utilization of the snapshot shift register for each clock cycle. The nomenclature that will be used is as follows:

- n : counter label number
- k : snapshot cycle. First cycle out of reset all of the counter values are snapshot (shifted over) to the shift up register bank. k is 0 for this cycle. The next snapshot of counter values is 400 clock cycles later and k will increment to 1.
- $X_{n,k}$ is the counter- n value that was snapshot into the shift up register for snapshot cycle k .

As previously stated, coming out of reset, each counter has an initial value equal ($X_{n,0}$) to the counter label number (n), e.g. Counter 0 has a reset value of 0 ($X_{0,0}=0$) and counter 99 has a value of 99 ($X_{99,0}=99$). As the circuit comes out of reset, the counter values are loaded into its corresponding register (within the shift-up register bank – see time τ in Fig 11). The counters continue to increment simultaneously as the shift registers shift counter values up every 4 clocks cycles – illustrated in Fig 11. The purpose of the shift-up is so each counter can reach shift register 0 (the output window to the tester). After all counters have been shifted up and loaded into the tester ($\tau+4N=$ once every 400 clock cycles), all of the counters are reloaded into the shift register bank.

As a summary, the key of the snapshot output scheme, is that the shift register array has now replaced a huge multiplexer. The benefits are as follows:

1. Counter upsets can easily be identifiable
2. Counters are incrementing and changing state every cycle. Hence maximum performance is able to be tested.
3. If a counter becomes upset, it will stay upset and it will eventually be captured during the snap shot period Counters are continuous and are not interrupted due an elaborate output scheme
4. Routing complexity is exclusive to just the counter array
5. The shift register architecture allows for high speed counter testing. A large multiplexer creates long paths of combinatorial logic and significantly slows down system speed.

The state space of the DUT should be deterministic and traversable. Pertaining to equation 3, for a 24 bit counter running at 25MHz and a shift up period of once every 4 clock cycles, it will take a little less than 1s for every state to be reached for all counters. Radiation tests generally last for several minutes. Hence, counter states are considered fully traversed within each test run.

$$\frac{2^{24}}{fs} = \frac{1.67 \times 10^7}{25MHz} = 0.67s \quad (3)$$

4.3.2 Counter I/O Interface and Expected Outputs

Table 4: DUT1 Counter I/O.

I/O Name	Bits	Dir wrt to DUT	Description
PROASIC_Counter_CLK	1	OUT	Counter shift Clock
PROASIC_COUNTER	24	OUT	Counter output
CLK	1	IN	Clock to counter circuitry
CLR	1	IN	Reset to counter circuitry

The DUT receives a clock and a reset from the tester. The expected output (PROASIC_COUNTER) is purely an increment by 1 starting at value 0 as illustrated in Figure 12. The first PROASIC_COUNTER will pertain to counter 0, followed by counter 1, counter 2... up to counter 99. A new snap shot is performed and PROASIC_COUNTER will restart by outputting counter 0.

With respect to the separate counters, Counter (n) is output every 400 cycles (400 cycles= 1 snapshot cycle). Therefore the counter values that represent each counter will increment by 400 for each snapshot cycle.

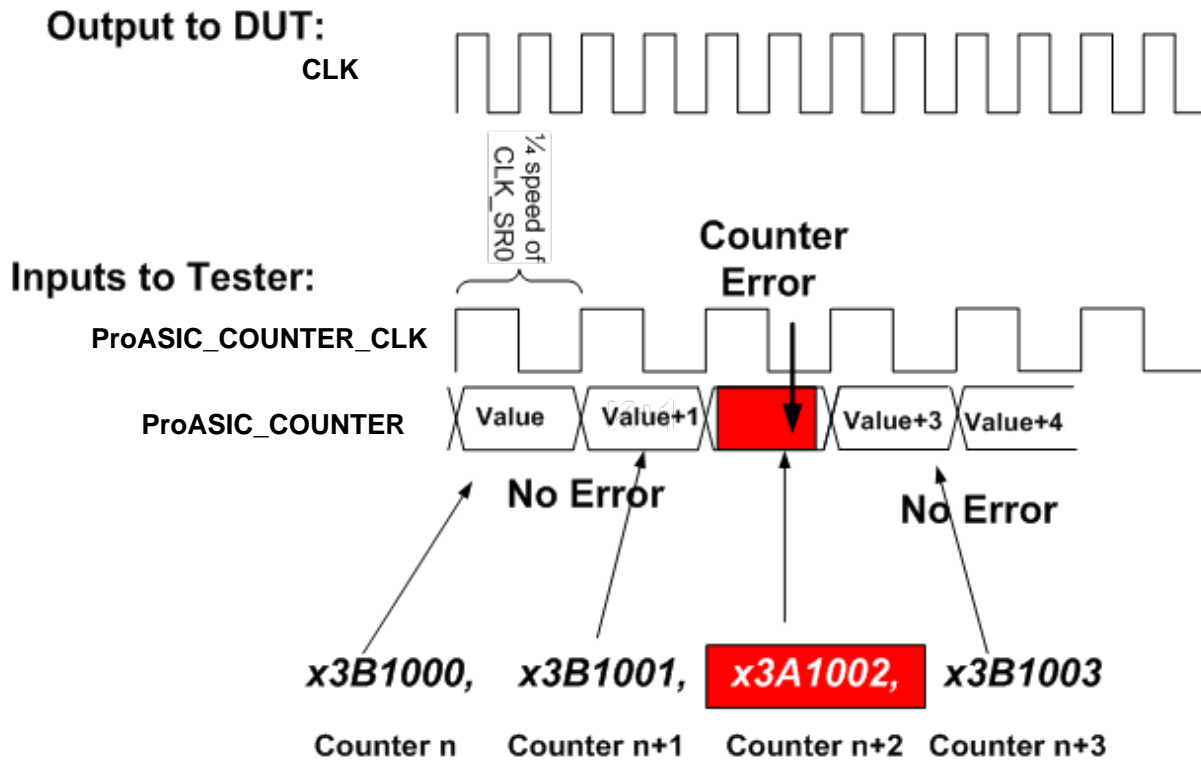


Figure 12: Typical SEE Counter Outputs. Each output represents a value from a different counter in the array. Counter selection is sequential, hence, the counter number and the counter values all increment by 1 each PROASIC_Counter_Clk cycle.

4.3.3 Counter Expected Upsets

- Bit or multiple bit upset: This example is illustrated in Figure 12. If a bit flips, it will stay flipped however, the counter will still increment.
- Broken counter: Counter stops counting and its value either stays constant or becomes complete noise
- Snap shot register:
 - A bit can get flipped while the counter value is in the snap shot register. In this case, the expected counter value will only be upset for one cycle.
 - Snap shot register can either stop shifting- in this case the output values will remain constant or become noise
 - Snap shot register can skip a cycle, in this case the counter values will be off the number of skipped shift cycles from their expected values
- Global routes: An upset can occur in the clock or reset circuitry

4.3.4 Summary of Counter Array Test Evaluations

The objectives of testing the counter array strings are to:

- Determine the impact to the error cross section when implementing complex structures with fanout. SET filtration and propagation will be heavily analyzed
- Frequency effect evaluation
-

4.4 Hamming Code 3 Finite State Machine (H3FSM)

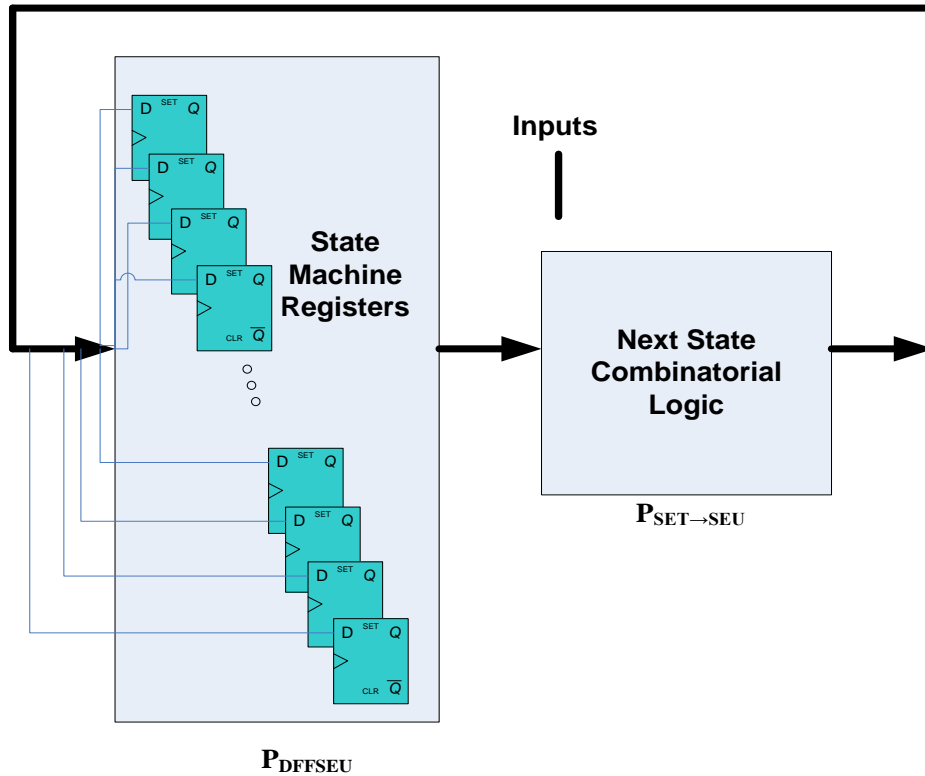


Figure 13: Finite State Machine (FSM) Schematic. FSMs consist of a bank of registers that feed combinatorial next-state logic.

Finite State Machines (FSMs) consist of registers (DFFs) that hold its current state and next state logic that determine the transition state for the next clock cycle. Figure 13 is an illustration of a typical FSM. Next state logic can either determine to retain the current state or transition to another defined state. FSMs are used to control various functions. Its premise is to define procedures to be performed at each state.

A FSM transitioning to an alternate state at an unexpected time can be detrimental to control, the physical device, or other devices within the system. Hence, attention is given to hardening techniques to FSM structures. A FSM with no mitigation has P_{DFSEU} , $P_{SET \rightarrow SEU}$, and P_{SEFI} components. [Mentor Graphics Precision Rad-Tolerant](#) has implemented a hamming code 3 (H3FSM) Error Detection and Error Correction (EDAC) scheme as an alternate mitigation strategy to TMR. The H3FSM structure is depicted in Figure 14. It is important to note that the unique components of a H3FSM are:

- (1) The encoding values of each state are complex schemes implementing hamming distance of 3
 - (2) The additional circuitry to perform EDAC. It relies on the premise that: $P_{DFSEU} \gg (P_{SET \rightarrow SEU} + P_{SEFI})$.
- However, because there is a significant amount of combinatorial logic per DFF, the premise may not be satisfied hence compromising mitigation. REAG has performed a preliminary study on the H3FSM state machines.

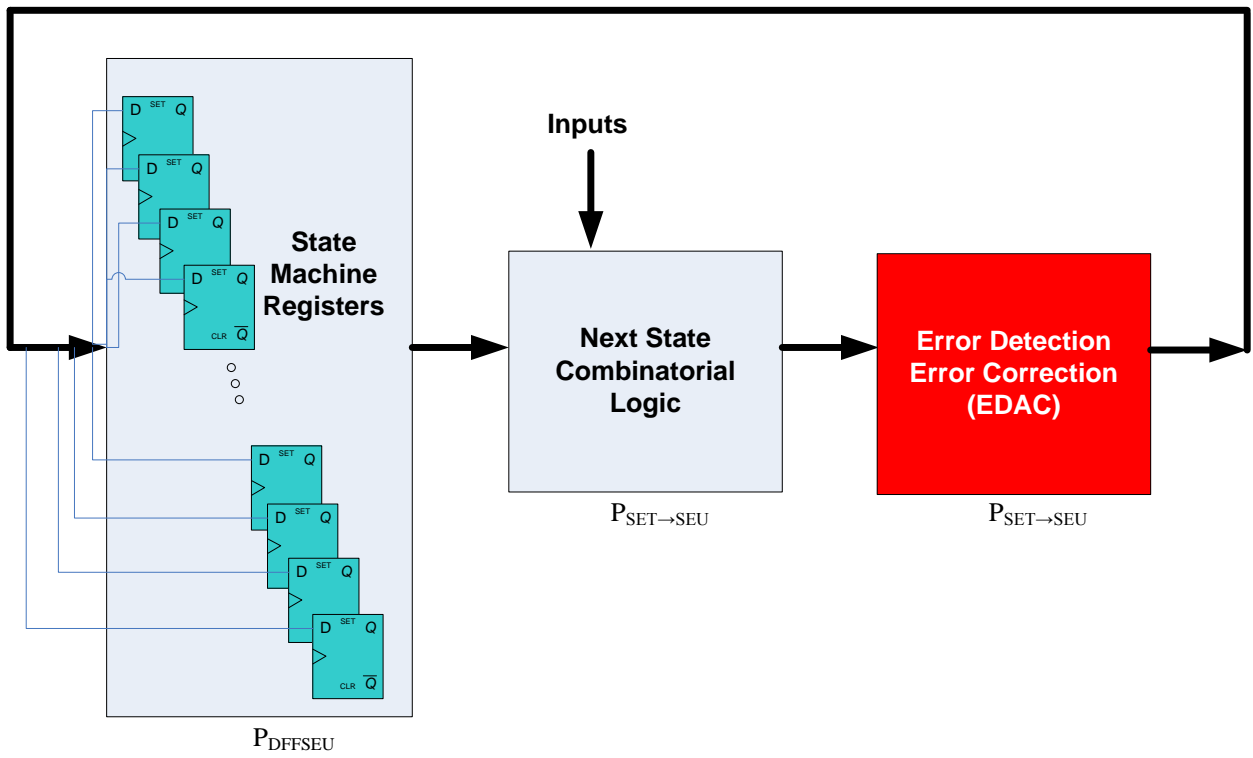


Figure 14: Finite State Machine (FSM) Schematic with Hamming Code 3 Error Detection and Error Correction.

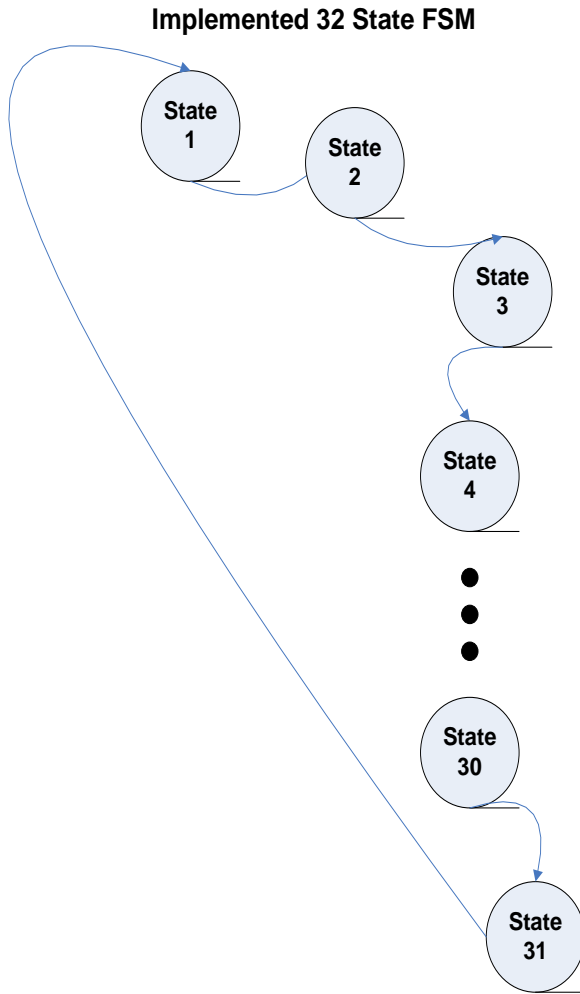


Figure 15: 32 bit Finite State Machine. Encoding used was Hamming Code 3.

Table 5: Hamming Code 3 Encoding of a 32 state FSM. 5bit Column+4bit Column comprise the entire state encoding = 9 bits.

9-bit Binary State Encoding	Decimal 5 bit Count	4-bit Parity
000000000	0	0000
000011001	1	1001
000100111	2	0111
000111110	3	1110
001000110	4	0110
001011111	5	1111
001100001	6	0001
001111000	7	1000
010000101	8	0101
010011100	9	1100
010100010	10	0010
010111011	11	1011

011000011	12	0011
011011001	13	1010
011100100	14	0100
011111101	15	1101
100000011	16	0011
100011010	17	1010
100100100	18	0100
100111101	19	1101
101000101	20	0101
101011100	21	1100
101100010	22	0010
101111011	23	1011
110000110	24	0110
110011111	25	1111
110100001	26	0001
110111000	27	1000
111000000	28	0000
111011001	29	1001
111100111	30	0111
111111110	31	1110

The implemented state machine is illustrated in Figure 15. It is a FSM that progresses to the next state every clock cycle. Although the encoding is hamming code 3, the FSM can be considered a counter with 4 redundant parity bits. Hence, 5 bits will cycle through the 32 binary counter states. The 4 additional bits are there to guarantee that the 32 states are at least a hamming code 3 distances apart. Subsequently there are a total of 9 bits per H3FSM. 242 H3FSMs were implemented. A snapshot architecture similar to the counter arrays (242 deep) was implemented to output the H3FSMs.

The 5 binary counter bits of the H3FSM cycle through the 32 states. At reset each H3FSM starts at state 0. Hence all state machines are the same coming out of reset (this is different than the counter array). At cycle n each H3FSM is expected to have:

(4)

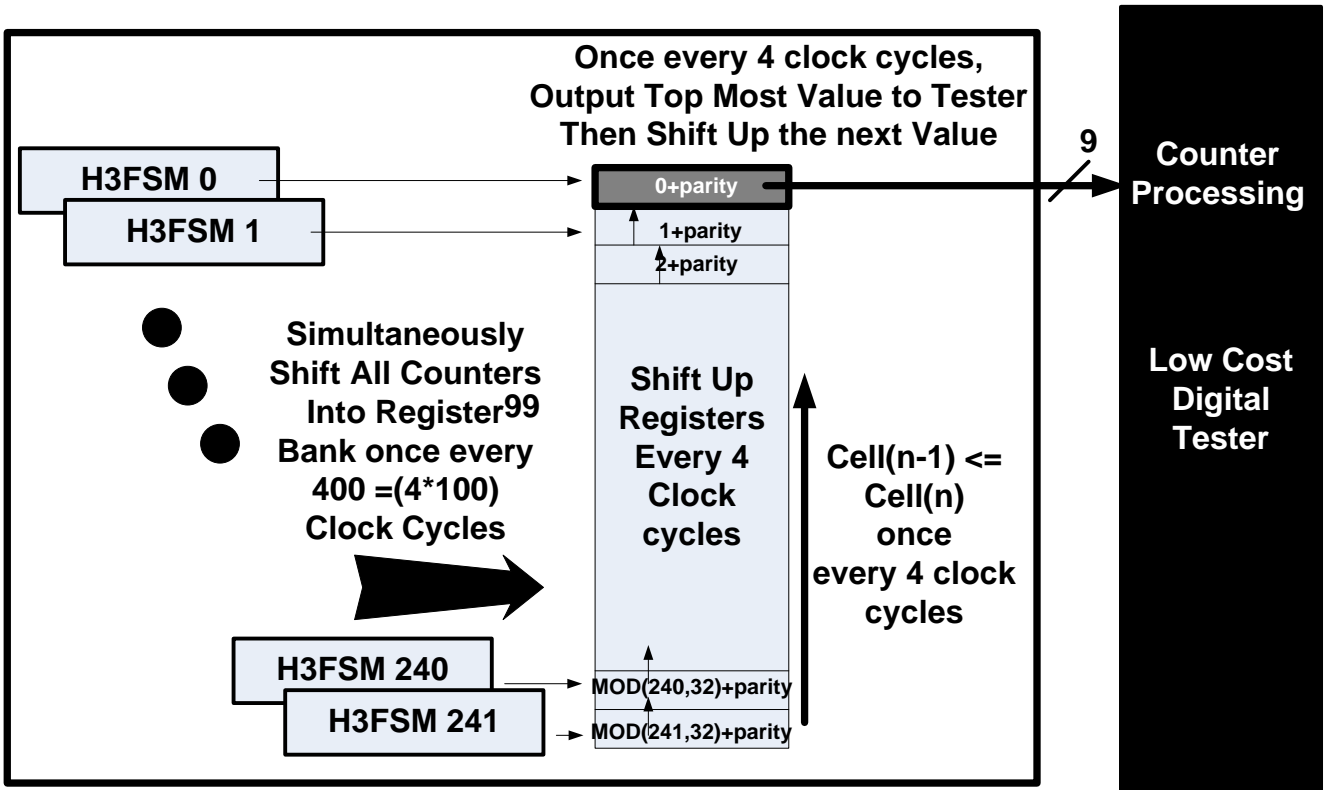


Figure 16: Hamming Code 3 Finite State Machine (H3FSM) Array Interface to Tester. Each H3FSM has a total of 9 bits. Subsequently, the interface to the tester is also 9 bits wide.

4.4.1 H3FSM Interface to Tester

Table 6: Hamming Code 3 Finite State Machine I/O

I/O Name	Bits	Dir wrt to DUT	Description
PROASIC_H3FSM_CLK_TM0	1	OUT	H3FSM shift Clock
PROASIC_H3FSM_CLK_TM1	1	OUT	H3FSM shift Clock
PROASIC_H3FSM_CLK_TM2	1	OUT	H3FSM shift Clock
PROASIC_H3FSM_TMR0	9	OUT	H3FSM output
PROASIC_H3FSM_TMR1	9	OUT	H3FSM output
PROASIC_H3FSM_TMR2	9	OUT	H3FSM output
CLK_TMR0	1	IN	Clock to counter circuitry
CLK_TMR1	1	IN	Clock to counter circuitry
CLK_TMR2	1	IN	Clock to counter circuitry
CLR_TMR0	1	IN	Reset to counter circuitry
CLR_TMR1	1	IN	Reset to counter circuitry
CLR_TMR2	1	IN	Reset to counter circuitry

4.4.2 H3FSM Expected Upsets

- Bit or multiple bit upset: This example is illustrated in Figure 12. If a bit flips, it will stay flipped however, the H3FSM will still progress to the next state. Errors will only be reported if the 5 state bits are upset. The redundant 4 bits are don't care elements.
- Broken FSM: FSM stops progressing and its value either stays constant or becomes complete noise
- Snap shot register:

- A bit can get flipped while the FSM value is in the snap shot register. In this case, the expected FSM value will only be upset for one cycle.
- Snap shot register can either stop shifting- in this case the output values will remain constant or become noise
- Snap shot register can skip a cycle, in this case the counter values will be off the number of skipped shift cycles from their expected values
- Global routes: An upset can occur in the clock or reset circuitry

4.4.3 Summary of H3FSM Array Test Evaluations

The objectives of testing the H3FSM array strings are to:

- Determine the impact to the error cross section when implementing complex structures with an alternate mitigation strategy than the common TMR approach.
- Frequency effect evaluation

5. HIGH SPEED DIGITAL TESTER (LCDT) TEST VEHICLE

The following sections describe the construction of the Low Cost Digital Tester (LCDT) including communication interfaces with the DUT and user PCs. Figure 17 is a picture taken at the Texas A&M Heavy Ion Cyclotron Facility illustrating the LCDT connected to the DUT ready to be irradiated.

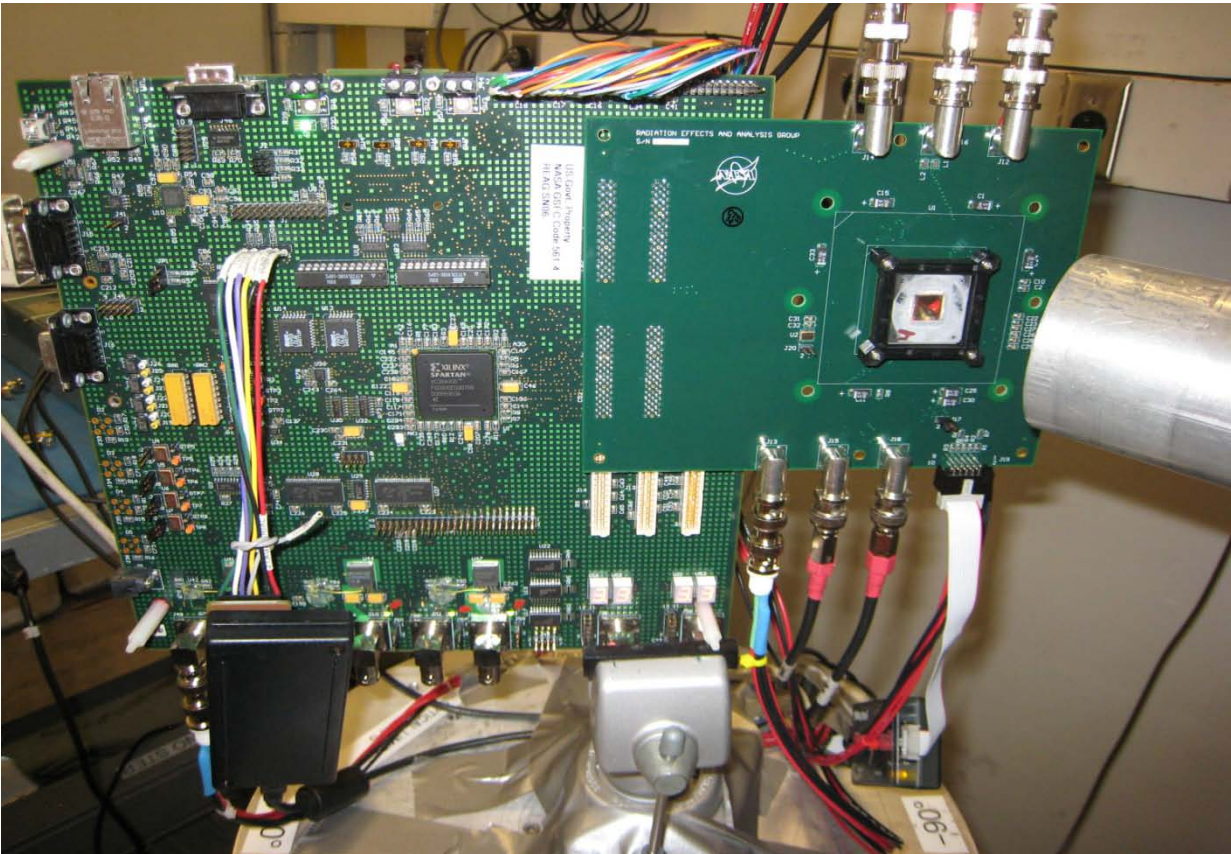


Figure 17: Picture of Low Cost Digital Tester (LCDT) connected to Device Under Test (DUT) at Texas A&M Heavy Ion Facility

5.1 Architectural Overview

The PROASIC controller/processor is instantiated as a sub component within LCDT. The test set-up consists of a Mother Board (FPGA Based Controller/Processor) and a daughter board (containing DUT and its associated necessary circuitry). The socket within the DUT Daughter board accommodates the ProASIC

devices. The objective of this DUT Controller/processor is to supply inputs to the ProASIC ACTEL Device and perform data processing on the outputs of the ProASIC. The LCDT communicates with a user controlled PC. The user interface is LAB-VIEW. It will send user specified commands to the mother board and receive information from the mother board. Please see Documents: "LCDT" and "General Tester" for further information concerning the LCDT functionality. A Picture of the tester and DUT at Texas A&M Cyclotron Facility is in Figure 17. The test setup schematic is shown in Figure 18.

Labview GUI Connected to Memory Processing in HSDT. Commands are also sent (and echoed) to the HSDT through this RS232 interface

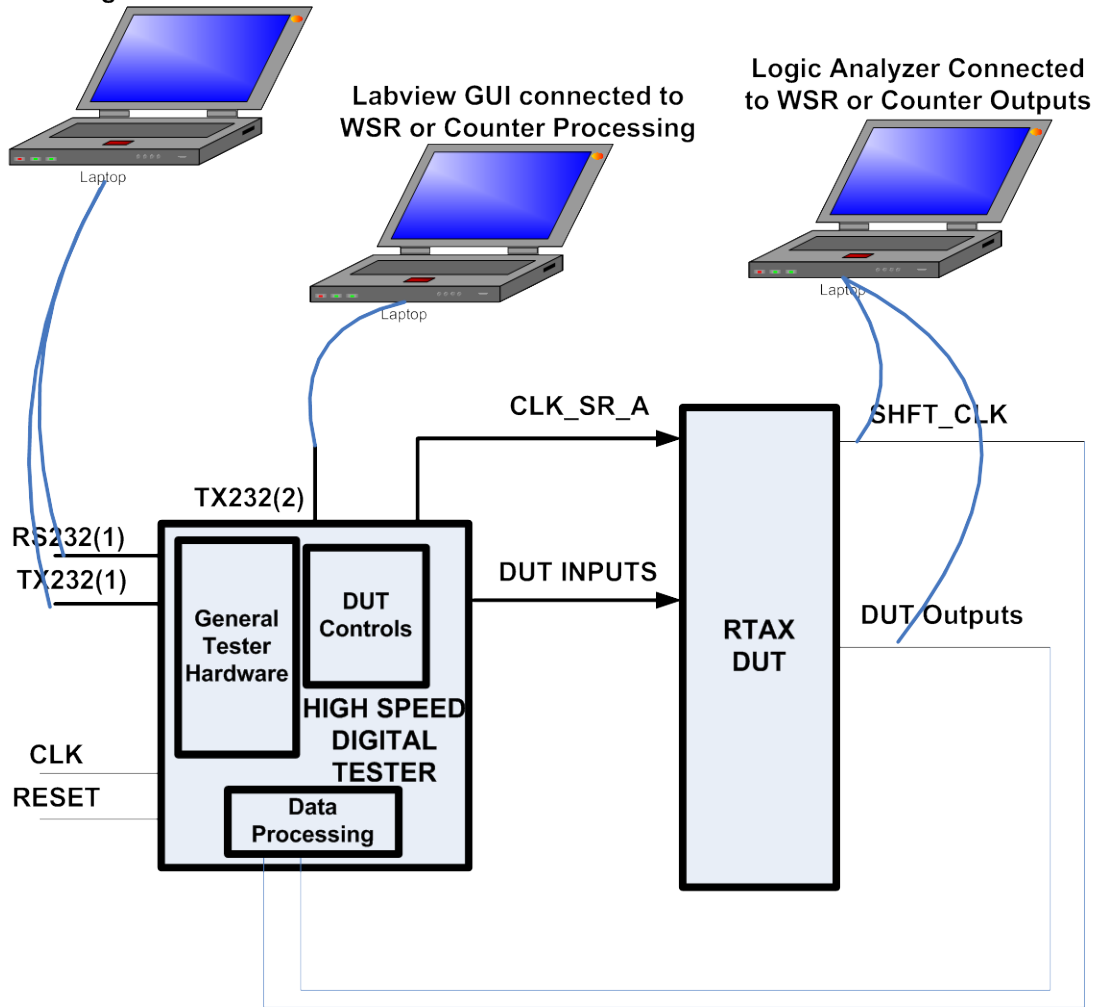


Figure 18: System Level Tester Architecture. Two PCs are running a Labview GUI. One PC is running a logic analyzer to have real time processing of the DUT WSR or Counter Array outputs. The PC connected to RS232(1) sends commands to the LCDT that set test parameters and starts test operations

5.1.1 I/O List and Definitions

Interface tables were supplied in the previous sections for all of the designs. The same I/O for each of the 2 DUTs are presented with respect to the LCDT in this section.

Table 7: I/O for Shift Register Tester.

Input Name	bits	Direction	Wrt	Description
------------	------	-----------	-----	-------------

to LCDT			
CLK	1	IN	System clock of the LCDT from Board crystal
RESET	1	IN	LCDT system reset from Power supply
RX232(1)	1	IN	Serial receive input from Host PC. Used for PC to send commands to the LCDT
TX232(1)	1	OUT	Serial transmission line to Host PC. Used to Echo commands and to send back either Shift Register or Counter Error Data
TX232(2)	1	OUT	Serial transmission line to Host PC. Used to send back memory error data
PROASIC_SHFT_CLK0	1	IN	Chain 0 shift Clock
PROASIC_SHFT_CLK1	1	IN	Chain 1 shift Clock
PROASIC_SHFT_CLK2	1	IN	Chain 2 shift Clock
PROASIC_SHFT_CLK3	1	IN	Chain 3 shift Clock
PROASIC_SHFT_CLK4	1	IN	Chain 4 shift Clock
PROASIC_SHFT_CLK5	1	IN	Chain 5 shift Clock
PROASIC_SHIFT_STRING0_TMR0	4	IN	4-bit windowed output of chain 0
PROASIC_SHIFT_STRING1_TMR0	4	IN	4-bit windowed output of chain 1
PROASIC_SHIFT_STRING2_TMR0	4	IN	4-bit windowed output of chain 2
PROASIC_SHIFT_STRING3_TMR0	4	IN	4-bit windowed output of chain 3
PROASIC_SHIFT_STRING4_TMR0	4	IN	4-bit windowed output of chain 4
PROASIC_SHIFT_STRING5_TMR0	4	IN	4-bit windowed output of chain 5
PROASIC_PATTERN_TMR0	2	OUT	Each chain gets bits from this vector: The bits select which data input pattern (all 0's, 1's, or checker board). See Table 3 for selection details
PROASIC_EN0_TMR0	1	OUT	Sent to chain enable or disable chain 3
PROASIC_EN1_TMR0	1	OUT	Sent to chain enable or disable chain 4 and 5
CLK_SR_TMR0	1	OUT	Clock to all of the WSR circuitry
CLR_SR_TMR0	1	OUT	Reset to all of the WSR circuitry

Table 8: I/O For Counter Tester.

Input Name	bits	Direction	Wrt	Description
------------	------	-----------	-----	-------------

to LCDT			
CLK	1	IN	System clock of the LCDT from Board crystal
RESET	1	IN	LCDT system reset from Power supply
RX232(1)	1	IN	Serial receive input from Host PC. Used for PC to send commands to the LCDT
TX232(1)	1	OUT	Serial transmission line to Host PC. Used to Echo commands and to send back either Shift Register or Counter Error Data
TX232(2)	1	OUT	Serial transmission line to Host PC. Used to send back memory error data
PROASIC_COUNTER_CLK0	1	IN	Counter shift Clock
PROASIC_COUNTER	24	IN	24 bit counter output
CLK	1	OUT	Clock to counter circuitry
CLR	1	OUT	Reset to counter circuitry

Table 9: : I/O For H3FSM Tester

Input Name	bits	Direction Wrt to LCDT	Description
CLK	1	IN	System clock of the LCDT from Board crystal
RESET	1	IN	LCDT system reset from Power supply
RX232(1)	1	IN	Serial receive input from Host PC. Used for PC to send commands to the LCDT
TX232(1)	1	OUT	Serial transmission line to Host PC. Used to Echo commands and to send back either Shift Register or Counter Error Data
TX232(2)	1	OUT	Serial transmission line to Host PC. Used to send back memory error data
PROASIC_H3FSM_CLK0	1	IN	H3FSM shift Clock
PROASIC_H3FSM	9	IN	9 bit H3FSM output
CLK_H3FSM	1	OUT	Clock to counter circuitry
CLR_H3FSM	1	OUT	Reset to counter circuitry

5.2 RS232 communication from the LCDT to the Host PC

All RS232 communication from the LCDT to the host PC is prefaced with a header. . Information from the LCDT to the Host PC is one of the following listed in Table 10: an alive-timer, a command echo, or an Error Record.

Table 10: A list of the LCDT to Host PC RS232 Header bytes. Only the LCDT uses header information. The host PC sends pure commands to the LCDT without headers.

Header	Description
00 FA F3 20	Alive Header No data bytes follow (i.e. only the header is sent from the LCDT to the PC)
00 FA F3 22	Command Echo. 4 data bytes follow that represent the command that was previously sent from the Host PC to the LCDT.
00 FA F3 21	Data Error Record: 23 bytes follow.

5.3 RS232 communication From the Host PC to the LCDT

Communication from the host PC to the LCDT does not contain a header. Information sent from the host PC to the LCDT are commands and are all 4 bytes in length The interface is controlled by a user GUI designed with LabView software.

5.3.1 User GUI

Commands are sent by typing specific values into Labview fields or controlling Labview on/off buttons listed on the screen. Figure 19 is the Labview Graphical User Interface (GUI) used to control and test the WSR designs. Figure 20 is the Labview GUI used to control and test the Counter designs. Figure 21 is the Labview GUI used to control and test the H3FSM designs.

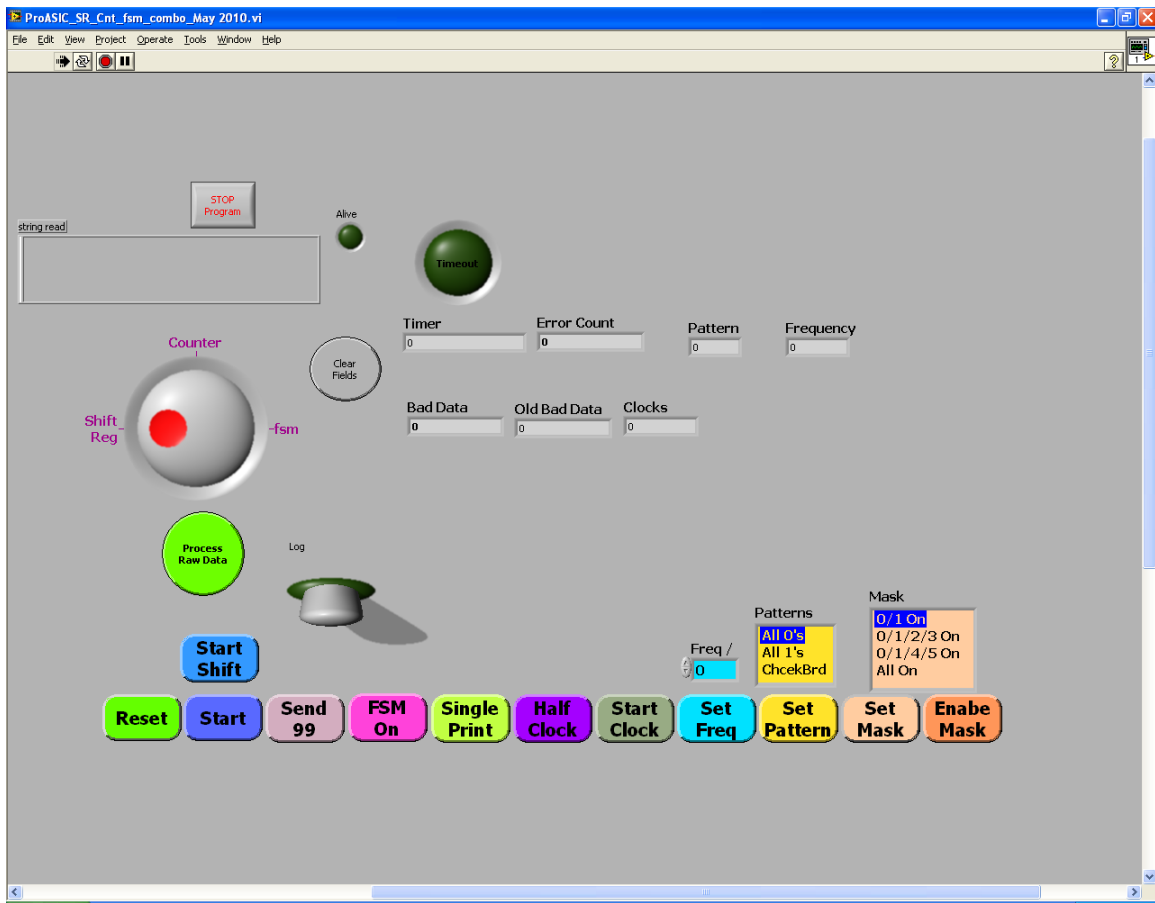


Figure 19: WSR GUI. Communicates with the Tester via the RS232(1) and TX232 (1) Interfaces. Commands are sent using this GUI. Command Echoes and WSR error reports are sent to this GUI from the Tester

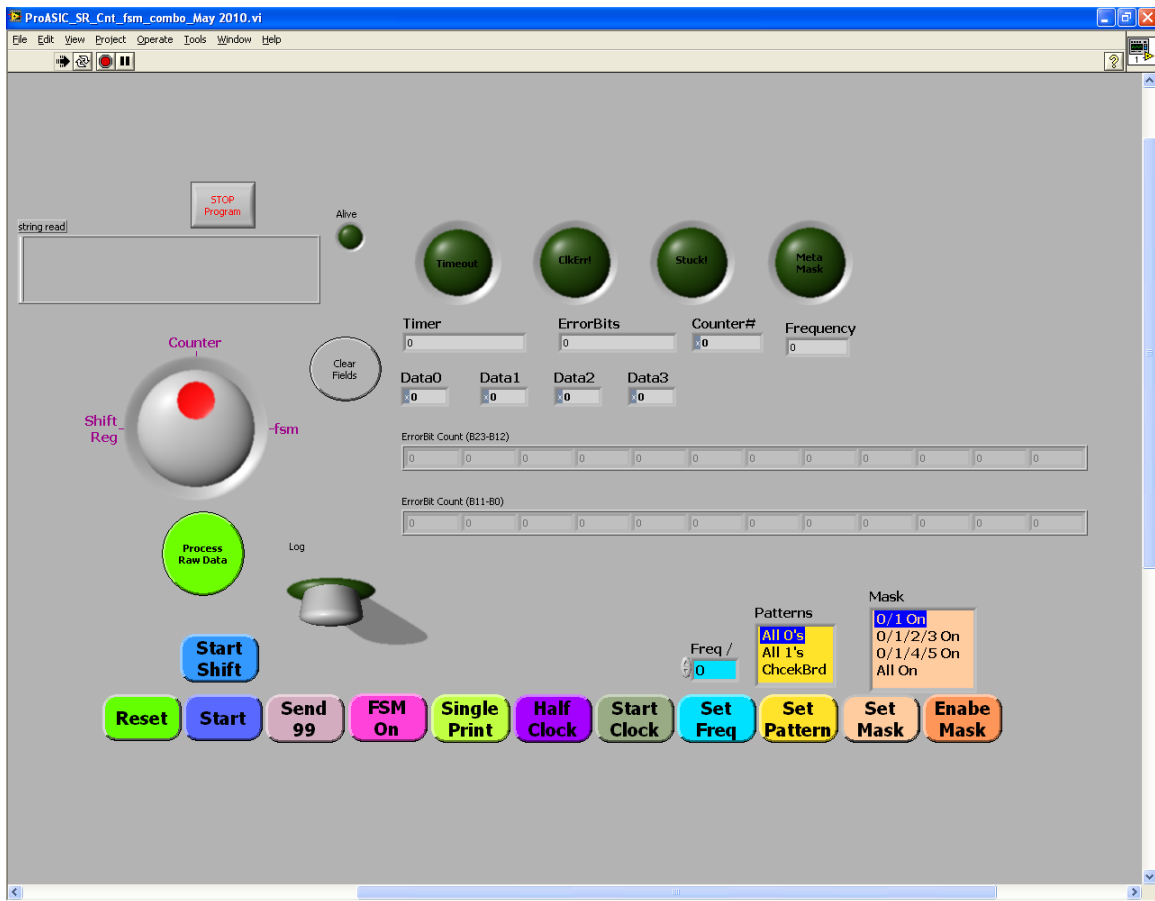


Figure 20: Counter Labview Interface. Communicates with the Tester via the TX232 (1) Interfaces. Counter error reports are sent to this GUI from the Tester.

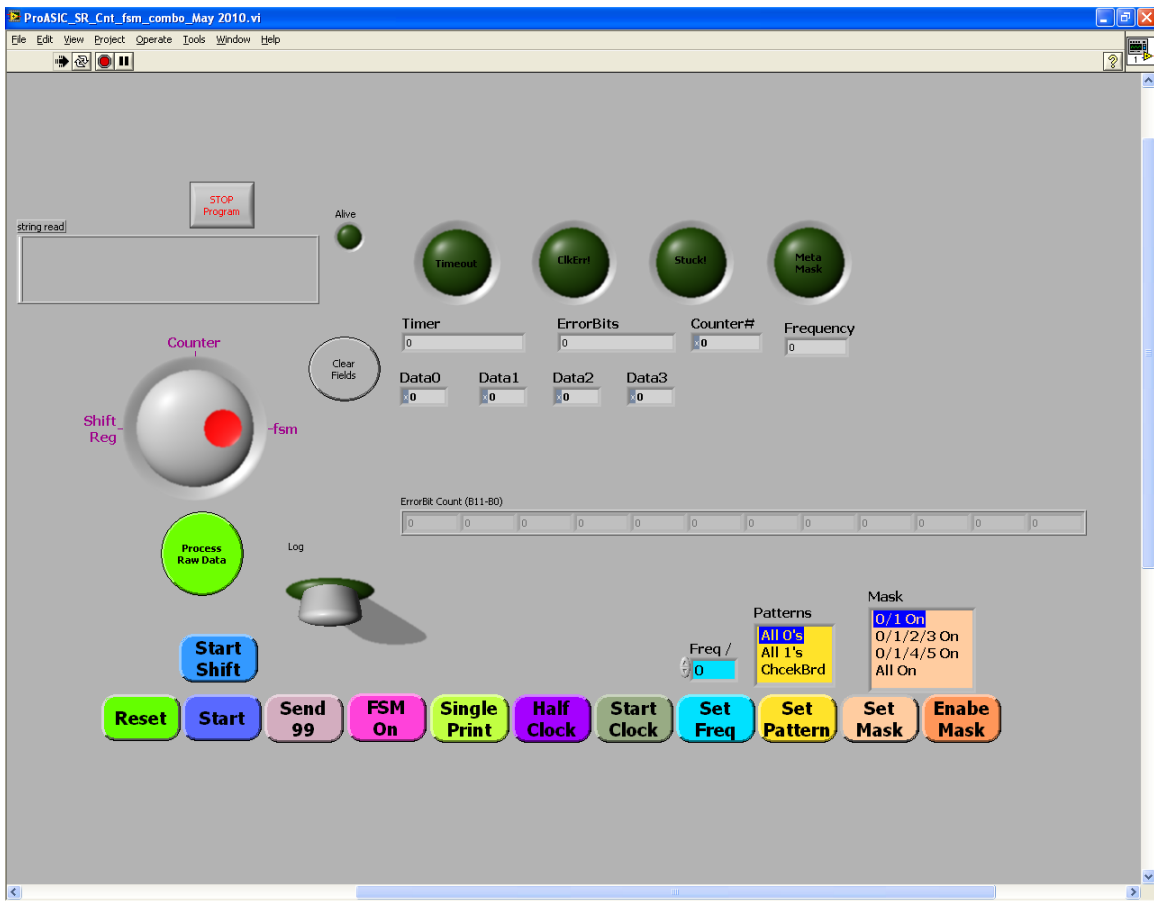


Figure 21:H3FSM GUI. Communicates with the Tester via the RS232(1) and TX232 (1) Interfaces. Commands are sent using this GUI. Command Echoes and H3FSM error reports are sent to this GUI from the Tester

The following section describes the commands sent from the Host PC to the LCDT.

5.3.2 User Interface and Command Control

The User controls the tests via a LABVIEW interface running on a PC. The PC communicates with the LCDT with a RS232 serial link. The format of communication is a command/Data 4 byte word.

Table 11 : Summary of Commands Used in PROASIC Tester

Command #	Command	D0	D1	D2	Description
01	Reset LCDT	n	n	n	Resets PROASIC
99	Reset DUT Memory	N	N	N	Resets ProASIC Memory control logic (address counters, read and write enables).
03	Reset of Shift register or counter tests	N	N	n	Places either the WSRs or the counter arrays in reset mode
04	Reset + start of Shift register or counter	N	N	n	Sends a reset pulse and then starts the WSRs or counter array tests

	tests				
90	WSR Pattern Number	y	n	n	D0 = 00... data input to WSR is a constant 0 01... data input to WSR is a constant 1 02 or 03... data input to WSR is checker board (i.e. changes every WSR clock cycle)
A1	WSR or Counter Array Clock Frequency	y	n	n	D0 is the Clock frequency divider of 100mhz. The synthesized clock will be sent to the DUT WSR or counter Arrays as their system clock.

6. DUT TEST PROCEDURES

6.1 DUT1 WSR Testing

The objectives of testing are to determine:

- Bit upset rates
- Frequency effects to SET capture rates
- Data Pattern effects to SET capture rates
- Global route effects

In order to obtain WSR objectives, tests were performed varying several parameters as listed in Table 12.

Table 12: WSR Test Parameter Variation

	Frequenc y Effects	Data Patter n Effects	Global Routes	General Bit Upsets	No TMR	LTM R	DTM R
<u>Architectural:</u> Variation in combinatorial blocks between DFFs in WSR	x	x	x	x	x	x	x
Frequency variation from test to test	x	x	x	x	x	x	x
Data pattern variation from test to test	x	x	x	x	x	x	x
LET variation	x	x	x	x	x	x	x
<u>Architectural:</u> Some chains contain global enable routes, some do not	x	x	x	x	x	x	x

6.1.1 Dynamic: Evaluate susceptibility of WSR

1. Bias the device, turn on clocks and toggle reset
2. Let WSR run and compare with expected DUT output pattern (verify no errors)
3. Irradiate DUT
4. Tester reads DUT and compares to expected value
 - o If error during read, then the LCDT records that an error has occurred and sends the data value

- with timestamp to the PC
 - o Goto 4 if not done with test else goto 5
- 5. Stop Beam
- 6. Reset Tester and DUT to prepare for next test

6.2 Counter Array Tests

6.2.1 Dynamic: Evaluate susceptibility of Counter DFF cells in biased-dynamic states

7. Bias the device, turn on clocks and toggle reset
8. Let counter logic run and compare with expected counter pattern (verify no errors)
9. Irradiate DUT
10. Tester reads DUT and compares to expected value
 - o If error during read, then the LCDT records that an error has occurred and sends the data value with timestamp to the PC
 - o If not done with test Goto 4, else goto 5
11. Stop Beam
12. Reset Tester and DUT to prepare for next test

Table 13: Counter Array Test Parameter Variation

	Frequenc y Effects	Data Pattern Effects	Global Routes	General Bit Upsets	No TMR	LTM R	DTM R
Frequency variation from test to test	x	x	x	x	x	x	x
LET variation	x	x	x	x	x	x	x

6.3 H3FSM Array Tests

6.3.1 Dynamic: Evaluate susceptibility of H3FSM DFF cells in biased-dynamic states

13. Bias the device, turn on clocks and toggle reset
14. Let counter logic run and compare with expected counter pattern (verify no errors)
15. Irradiate DUT
16. Tester reads DUT and compares to expected value
 - o If error during read, then the LCDT records that an error has occurred and sends the data value with timestamp to the PC
 - o If not done with test Goto 4, else goto 5
17. Stop Beam
18. Reset Tester and DUT to prepare for next test

Table 14: H3FSM Array Test Parameter Variation

	Frequenc y Effects	Data Pattern Effects	Global Routes	General Bit Upsets	No TMR	LTM R	DTM R
Frequency variation from test to test	x	x	x	x	x	x	x
LET variation	x	x	x	x	x	x	x

7. PROCESSING THE DUT OUTPUTS

The outputs of the DUT are fed to the tester for data processing. The objective of the data processing is to capture data from the DUT, compare to an expected value, and report to the host PC if there is an error. The DUT system clock and reset signals are generated in the LCDT.

7.1 WSR, Counter, H3FSM SHIFT_CLK Processing

Regarding the SHIFT_CLK, it is used to alert the tester that the DUTs are alive. The SHIFT_CLKs are always $\frac{1}{4}$ of the speed of the DUT system clock.

Due to the interface delays and device latencies and in order to consequently decouple the DUT to tester timing restrictions, the DUT SHIFT_CLK is considered asynchronous to the tester and is sampled using the LCDT system clock. Thus, the tester's sampling clock will always be 4 times as fast as SHIFT_CLK. The SHFT_CLK is fed into a metastability filter and an edge detect. This process takes 1 to 2 clock cycles of the sampling clock (detection will be delayed by 1 to 2 sampling clock cycles of the actual edge).

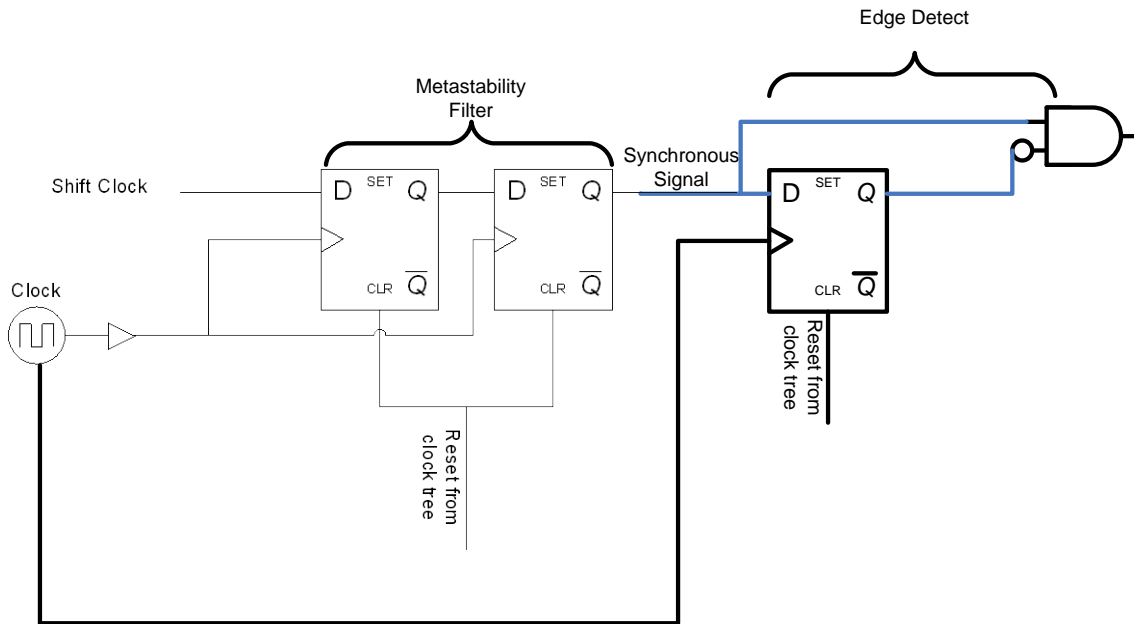


Figure 22: Shift_CIK Capture consists of a Metastability Filter and a Edge Detect

The SHIFT_CLK edge is expected to come at a frequency that is $\frac{1}{4}$ of the LCDT clock. If the edge is stopped, glitched, or missing, the event is reported to the host PC by the LCDT.

7.2 WSR Data Processing

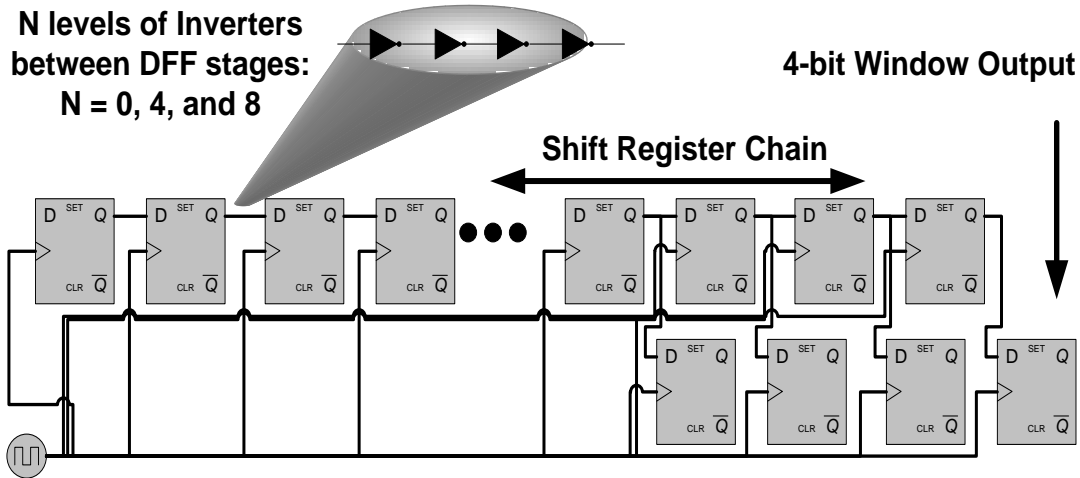


Figure 23: Another look at a WSR string. The 4-bit window is the registers in the illustration that are underneath the shift register string. The last 4-bits of the string are shifted into window, once every 4 clock cycles.

In order to avoid metastable events due to an error in the output, data is registered twice before evaluation. As illustrated in Figure 8, the four bit WSR output window stays constant for all of the tested data patterns (all 0's, all 1's, and checkerboard). Therefore determining bit flips are simple... it's merely checking for a change in data input (does the data from the current cycle(n) equal the data from the last cycle(n-1)?).

One must take caution because the tester is always reporting changes in data. This requires a record report when data changes from expected to corrupted, and then from corrupted to expected. In other words, not every report from the tester suggests data is bad. On the good note is with the inclusion of the timestamp, this gives the user the ability to post process and to determine how long data is in error and if recovery is possible.

Table 15: Types of Errors with Detection and Reporting Scheme

Error	LCDT Detect
Bit flip in shift register string	Has the data changed from the previous cycle?
Global Route glitch: Clock, reset, or enable	There will be a burst of bad data in the data string. A reset has a strict error signature because of how the string comes out of reset... e.g. the string is set to all 0's for a certain number of cycles. A clock or enable glitch can cause a burst of noise data. All cases are eventually recoverable for a WSR.
Bit flip in window or I/O	Has the data changed from the previous cycle and is only corrupt for less than the 4 clock cycle window.

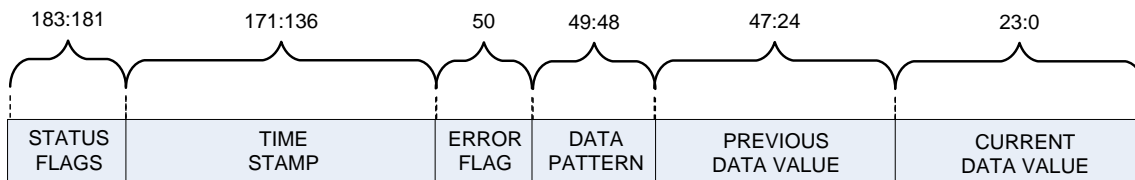


Figure 24: WSR Error Record Data Fields. Each error record is prefaced with an error header (00 FA F3 21) when being sent from the LCDT to the Host PC.

Figure 24 demonstrates an error record that is sent from the LCDT to the user PC regarding a WSR behavior. The relationship between the previously captured and currently captured WSR windowed values explains the error signatures and event occurrences within the WSR structure during testing.

Table 16 WSR Field Table. Yellow shading indicates fields that are generated from DUT inputs. White Fields are sourced from either tester settings or tester logic

Field	# of Bits	Description
Current Value	24	Currently captured data (cycle N)
Previous Value	24	Previously captured data (cycle N-1)
Data Pattern	2	2 bit data pattern set by commands (00 or 01 or 10)
Error Flag	1	Not used
Freq	8	DUT frequency set by user command
Error Count	16	Unused
Time Stamp	32	Cycle counter. Must multiply by the DUT frequency to convert to time. Used to determine error burst sequences
Status	3	Indicates type of error record: "001" is a timeout – one of the shift clocks not detected "011" Out of timeout – all shift clocks are recovered "000" Error or non-error – current value does not equal previous value "010" Debug check – command was sent to check value settings

Table 17: WSR Current Value and Previous Value Processing

Previous Value and Current Value	Indication
Previous = good and Current=bad	WSR just reported an error
Previous = bad and Current=good	WSR has recovered from error
Previous= bad and Current = bad	WSR is in a burst of error
Previous = exact inversion of Current	Either an enable was struck or there was a shift clk hit (e.g. previous = hex 5 and current = hex A).

7.2.1 WSR SEU Cross Section Calculations

Generally, calculating the SEU Cross Section for WSR chains is a simple process. Count the number of upsets and divide by the reported particle fluence. The WSR cross section is then normalized to the number of bits within the chain that is being analyzed. When there is a burst of data, the total fluence must be adjusted because upsets are not being captured during the inoperable burst period. Hence, the number of particles that the device experienced during said period is subtracted off of the total.

(5)

7.3 Counter Array Data Processing

The DUT output to the tester will increment once every 4 cycles. The increment is a continuous sequence of counts. It is important to note the difference between the terms counter number and counter value. The counter number is a tag (or name) given to a counter. The counter value is the actual data that is stored in the counter (or the current state of the counter). The tester keeps a local copy of the expected counter number with respect to the incoming counter value in order to keep track of the integrity of each counter.

7.3.1 Counter Array Data Capture and compare

In order to avoid metastable events due to an error in the output, data is registered twice before evaluation. Both the data and the counter number are expected to increment every 4 cycles and will wrap around at its boundaries as listed in Table 18.

Table 18 Counter Value and Counter Number Wrap around Boundaries

Bits Wrap Around Value

Counter values	24	$2^{24}-1$ (after $2^{24}-1$ next value is 0)
Counter Number	8	99 (after 99 next value is 0)

Regarding DUT value comparisons, any change in DUT output counter value must be an increment of 1 from the previous DUT counter value. If not, then an error record is sent to the LCDT. One must take caution because this will require at least 2 records per upset. The first record will be the counter that is in upset and the next record will be the following counter that is not in upset. This is because neither of the two counters will be an increment of 1 apart.

Post processing of the output records will help to determine if the upset occurred within the snap shot register or in the counter. This is done by understanding that if a counter is upset, it will stay upset. However, an upset in the snapshot register will only be upset for 1 snapshot cycle.

Upsets in the counter shift_clk value (should be a signal that is 1/4 the DUT clock) was described in Section 7.1.

7.3.2 Counter Array Error Record

A significant amount of post processing is expected to be performed on this data. Subsequently, the error record should contain enough information to comprehend and differentiate between events.

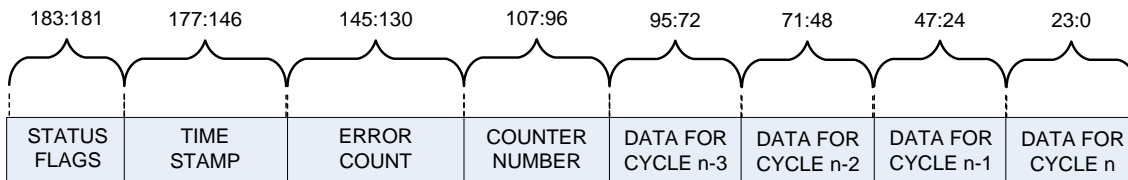


Figure 25: DUT2 Counter Array Error Record. Cycle n represents capture cycles. Capture cycles are once every 4 LCDT tester clock cycles.

Table 19: Counter Array Error Record Fields: Yellow indicates Fields generated from DUT Inputs

Field	Bits	Description
Data Cycle N	24	Current DUT output. Error: If it is not an increment of 1 from the previous counter value and not an increment of 2 from the data value received cycle (n-2) No error (recover from error): If it is not an increment of 1 from the previous value but is an increment of the data value received cycle (n-2) Otherwise: DUT is in a burst of error
Data Cycle N-1	24	Capture cycle n-1 DUT output (capture cycle n-1 is actually 4 LCDT clock cycles from Data cycle N)
Data Cycle N-2	24	Capture cycle n-2 DUT output (capture cycle n-2 is actually 8 LCDT clock cycles from Data cycle N)
Data Cycle N-3	24	Capture cycle n-1 DUT output (capture cycle n-3 is actually 12 LCDT clock cycles from Data cycle N)
Counter Number	8	Tester local copy of expected counter number. (0 through 99)
Error Count		
Time Stamp	32	
Status flags	3	

7.3.3 Counter Array SEU Cross Section Calculations

Because each of the 24 DFFs within each counter has different logic feeding into its data pin, all 24 bits of the counter are analyzed separately. Hence, there are 24 different cross sections. As an example, given that there are 100 counters total, the Cross section for bit0 will be made up of upsets on the 100 different bit0 per

counter. Counter Array SEU cross sections in this document will be reported as binned cross sections. Each bin is an average of 4 counter bit SEU cross sections. The first bin in the average cross section for bit0, bit1, bit2, and bit3. There are six bins in total because there are 24 bits in a counter. There are 400 bits within each bin.



7.4 H3FSM Array Data Processing

The DUT output to the tester operates in the same fashion as the counter array. The difference is that the H3FSM is 9 bits while the Counter Array is 24 bits. In addition, the expected increment is based off of the 5 bits that represent the binary count. An upset in the parity (redundant bits) is not reported because the state machine stays intact.

7.4.1 H3FSM Array Data Capture and compare

In order to avoid metastable events due to an error in the output, data is registered twice before evaluation (metastability filter). Both the data and the H3FSM number are compared to their expected values once every 4 cycles and will wrap around at its boundaries as listed in Table 18.

Table 20: H3FSM Wrap Around

	Bits	Wrap Around Value
Counter values	9	2^9-1 (after 2^9-1 next value is 0)
Counter Number	8	241 (after 241 next value is 0)

Regarding DUT value comparisons, any change in DUT output counter value must be an increment of 1 from the previous DUT counter value. If not, then an error record is sent to the Tester. One must take caution because this will require at least 2 records per upset. The first record will be the H3FSM that is in upset and the next record will be the following counter that is not in upset. This is because neither of the two counters will be an increment of 1 apart.

Post processing of the output records will help to determine if the upset occurred within the snap shot register or in the counter. This is done by understanding that if a counter is upset, it will stay upset. However, an upset in the snapshot register will only be upset for 1 snapshot cycle.

Upsets in the counter shift_clk value (should be a signal that is ¼ the DUT clock) was described in Section 7.1.

7.4.2 H3FSM Array Error Record

A significant amount of post processing is expected to be performed on this data. Subsequently, the error record should contain enough information to comprehend and differentiate between events. The H3FSM uses the same format as the counter, however, within the lower 4 fields (24 bit wide), the H3FSM only utilizes the lower 9 bits of each field.

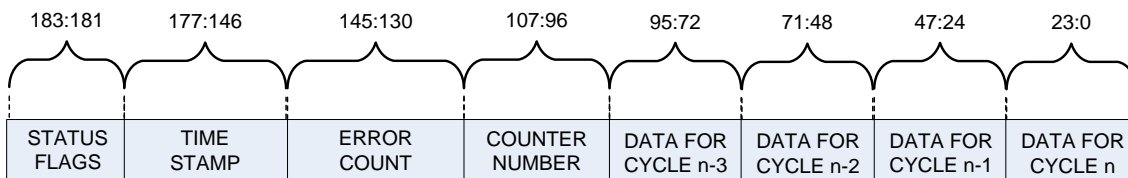


Figure 26: Counter Array Error Record. Cycle n represents capture cycles. Capture cycles are once every 4 LCDT tester clock cycles.

Table 21: DUT2 Counter Array Error Record Fields: Yellow indicates Fields generated from DUT Inputs

Field	Bits	Description
Data Cycle N	24	Current DUT output. However, Output only occupies the lower 9 bits of this field. Error: If it is not an increment of MOD(4*(242),32) from the same H3FSM of the previous snap shot cycle No error (or recover from error): Recover from error if value = MOD(4*(242),32) from the same H3FSM of the previous snap shot cycle Otherwise: DUT is in a burst of error
Data Cycle N-1	24	H3FSM number-1 value from previous snapshot cycle. Output only occupies the lower 9 bits of this field.
Data Cycle N-2	24	H3FSM number-2 value from previous snapshot cycle. Output only occupies the lower 9 bits of this field.
Data Cycle N-3	24	H3FSM number-3 value from previous snapshot cycle. Output only occupies the lower 9 bits of this field.
H3FSM Number	8	Tester local copy of expected H3FSM number. (0 through 241)
Error Count		
Time Stamp	32	
Status flags	3	

7.4.3 Counter Array SEU Cross Section Calculations

Because each of the 5 DFFs within each H3FSM has different logic feeding into its data pin, all 5 bits of the counter are analyzed separately. Hence, there are 5 different cross sections. As an example, given that there are 242 H3FSMs total, the Cross section for bit0 will be made up of upsets on the 242 different bit0 per H3FSM. Although most of the processing for H3FSM is similar to the counter, H3FSM cross sections are not binned in this document.

(8)

8. HEAVY ION TEST FACILITY AND TEST CONDITIONS

Facility: Texas A&M University Cyclotron Single Event Effects Test Facility, 15 MeV/amu tune).

Flux: 1.0×10^4 to 2.0×10^5 particles/cm²/s

Fluence: All tests were run to 1×10^7 p/cm² or until destructive or functional events occurred.

Table 22: LET Table

Ion	Energy (MEV/Nucleon)	LET (MeV*cm ² /mg) 0°	LET (MeV*cm ² /mg) 45°	LET (MeV*cm ² /mg) 60°
Ne	15	2.8	5.6	
Ar	15	8.5	12.6	
Cu	15	20.3		
Kr	15	28.7	40.73	57.6
Xe	15	52.7	75.09	106

Test Conditions:

Test Temperature: Room Temperature

Operating Frequency: 15 MHZ to 160MHZ

Power Supply Voltage: 3.3v I/O and 1.5V Core.

The PROASIC devices were irradiated with Argon, Krypton, and Xenon beams at normal incidence, 0, 45 and 60 degrees (yielding effective LETs values listed in Table 22: LET Table) at the Texas A&M University Cyclotron Single Event Effects Test Facility.

The PROASIC devices were monitored for Single Event latchup under the above conditions. Each part was placed in the beam until a Single Event latch (SEL) event occurred or 10^6 ions/cm² – the beam fluence was then recorded. During our experiment, no Single Event latchup events occurred, yielding a threshold SEL LET for latchup of $> 10^6$ MeV•cm²/mg.

The PROASIC devices were also tested to measure the error cross section under the above conditions. Each part was placed in the beam until 10^6 ions/cm² was reached. An average cross section per bit was determined for a given LET as the number of fault events observed divided by the total fluence of the associated run at that LET.

It is important to note that it has been shown that the ProASIC device has a low tolerance to Total Dose. Hence it was necessary to not exceed 10Mrad per device during testing. Because a large number of tests were expected to be performed and only 11 devices were available, particle fluence per test run had to be kept at a minimum and total dose had to be recorded per run. The consequence to testing with low fluence is the inability to observe upsets at low LETs or at low Frequencies. As a result, the test report is considered preliminary and follow-up tests are suggested in order to increase SEU statistics.

9. PRELIMINARY WSR HEAVY ION TEST RESULTS

9.1 Testing Difficulties

Initial test plan was to place a device within the socket on the DUT test board. The device will be programmed with the target design (e.g. No TMR WSRs, LTMR Counter, DTMR WSR, etc...). The target design is tested at various frequencies and data patterns. On completion of the tests, a new design is programmed into the DUT and it is tested with various frequencies and data patterns. The reprogramming only occurs while the heavy ion beam is turned off. This procedure of testing and reprogramming is performed until the device has acquired 10Mrad. At this point, a new DUT replaces the old in the DUT board socket and test procedure recommences.

9.1.1 First Test Trip 05/2010

During the first test trip, the initially planned procedure was not able to be successfully performed. Tests started at high LET values (Xe = 53 MeV•cm²/mg). At these LET values, we found that we would not be able to reprogram the devices after approximately 3 to 7 test runs (about 5Mrads). In other words, after testing a LTMR WSR, a DTMR WSR was downloaded to the ProASIC and the reprogram failed. It is important to note that (1) reprogramming only occurred when the beam was off and (2) the devices were fully operational prior to reprogramming. Hence, it was only the reprogramming circuitry that would fail. In addition, once the attempt to reprogram was made, because of the configuration flash erase cycle, the device was no longer functional. Once again, reprogramming was only performed while the heavy ion beam was turned off. REAG is currently investigating whether this effect will anneal such that the destructive failures no longer exist.

9.1.2 Second Test Trip 08/2010

Due to the difficulties first observed with reprogramming the device, the test procedures were changed. Each device had a dedicated design – no reprogramming was performed.

Only lower LET values were used during heavy ion testing (2.8MeV•cm²/mg – 28MeV•cm²/mg). Although no attempts to reprogram were performed on test-site, once the devices were returned to the REAG lab, reprogramming tests were performed. The objective was to further investigate potential programming damage after heavy-ion beam exposure. No upsets were observed.

9.1.3 Summary of Reprogramming issue

Device reprogramming failures were observed with LET > 53 MeV•cm²/mg. Attempts to re-program the dead devices have been performed to determine if there are annealing effects. It is interesting to note that, devices that will not reprogram with one particular design may reprogram with another (and be fully functional). However, this is not always the case, some devices are not able to be reprogrammed at all. This reprogramming effect was not observed at lower LET values.

It is suggested that this phenomena be further investigated at another time.

9.2 No-TMR

No TMR 100MHz WSR Strings

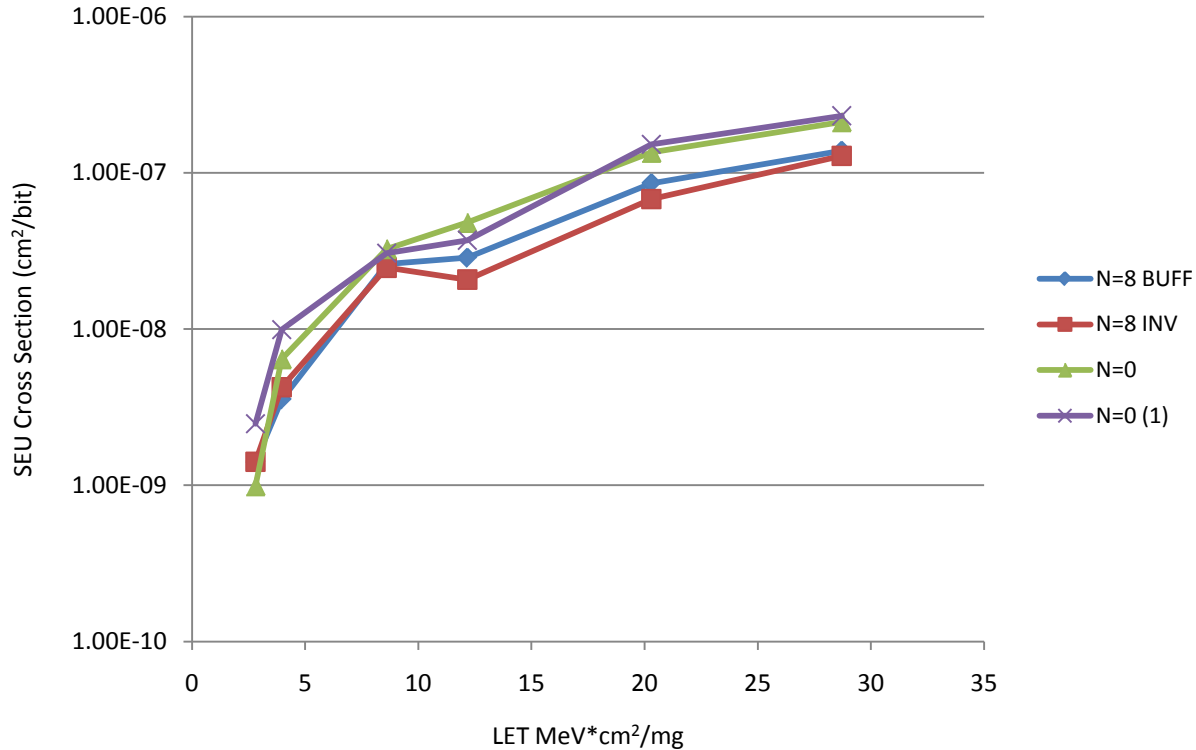


Figure 27: No-TMR Log-linear curves at 100MHz for N=0 and N=8 Strings

No-TMR designs do not contain mitigation. Hence the SEU cross section (σ_{SEU}) is based on DFFs, combinatorial logic, and global routes (SEFIs).

(9)

(10)

As shown in Figure 35, (No TMR N=0) SEU cross sections are slightly higher than (No TMR N=8) SEU cross sections.

(11)

This suggests that the DFFs are more susceptible than combinatorial logic blocks. By definition, N=0 WSRs only contain DFFs. Hence, based on the number of particles and types of particles, all upsets will incur in DFFs (or global routes) and are not frequency dependent. N=8 chains have DFFs and combinatorial logic. Hence, for the same number of particles, upsets are shared between DFFs and combinatorial logic (and global routes). Due to the frequency dependence of $P_{SET \rightarrow SEU}$ not all transients will get captured – however some will get captured. Because the difference across all σ_{SEU} (N=0 and N=8) is insignificant at lower LETs, it can also be deduced that $P_{SET \rightarrow SEU} \ll P_{DFFSEU}$. Although N=0 chains have a slightly higher cross section than N=8, the difference is insignificant such that one can assume the cross sections to not be design dependent and have similar values.

(12)

This hypothesis can be observed by comparing Figure 28 through Figure 32. No-TMR Counter Array Cross sections are almost flat across all bits and have similar values to the WSR cross sections.

(13)

9.2.1 No-TMR WSR String

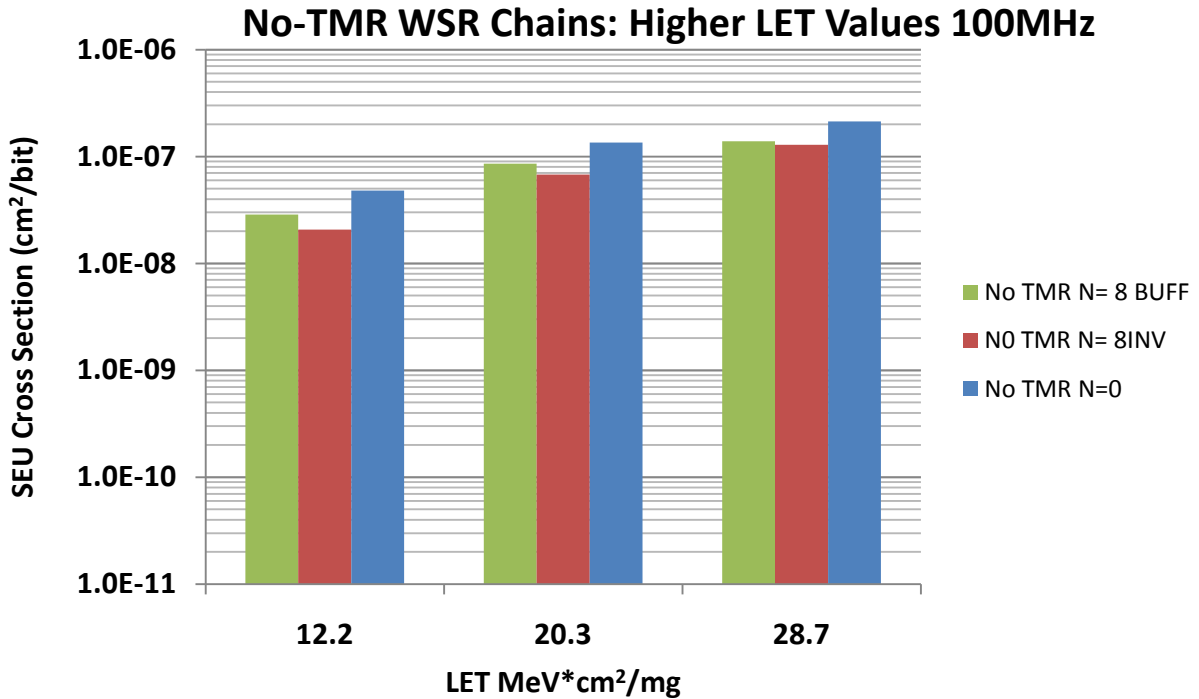


Figure 28: No-TMR at 100MHz with LET values > =12 MeV*cm²/mg. N=0 chains

Figure 31 illustrates the dominance of $P_{DFE_{SEU}}$ at low LETs. Examining the SEU cross section across counter bits, there is an insignificant difference between upsets. If $P_{SET \rightarrow SEU}$ had a significant impact then data pattern, frequency, and fanin/fanout would cause the various bits to incur significantly different cross sections.

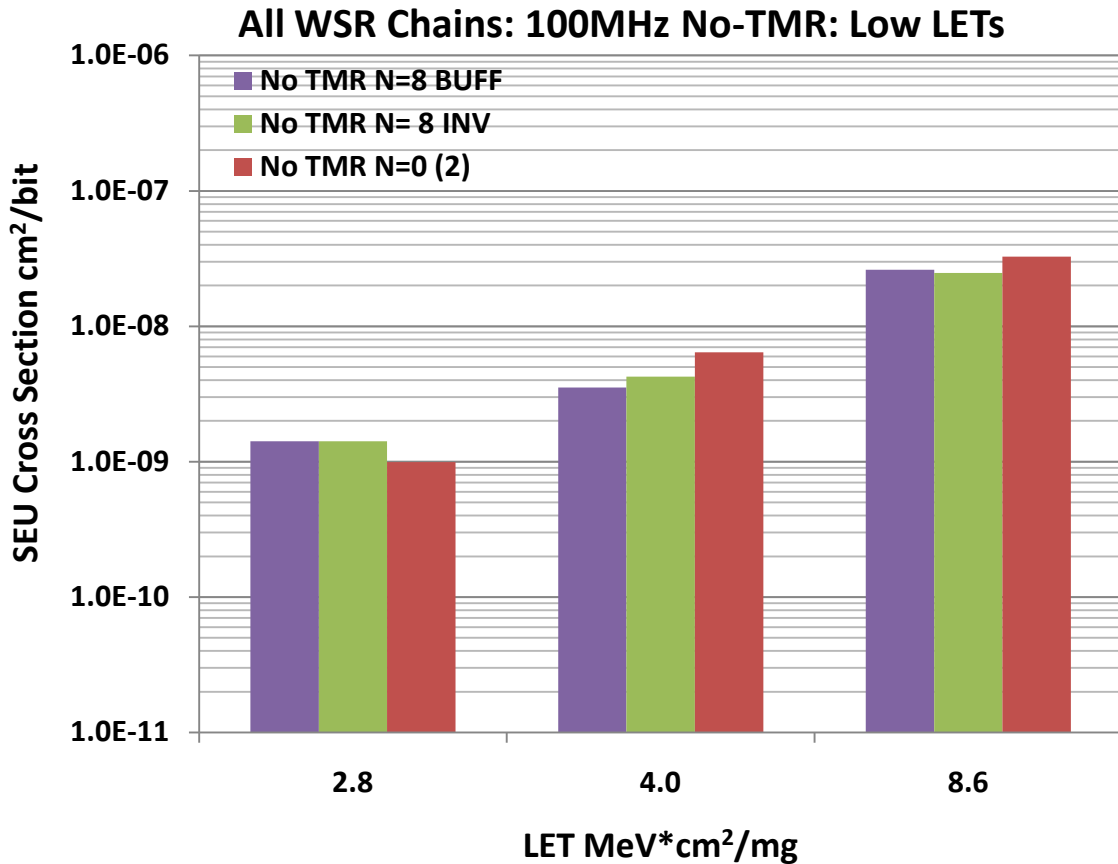


Figure 29: No TMR for WSR chains N=0 to N=8. 100MHz Operational Frequency Checkerboard Pattern. N=16 WSRs not in graph because they cannot operate at 100MHz.

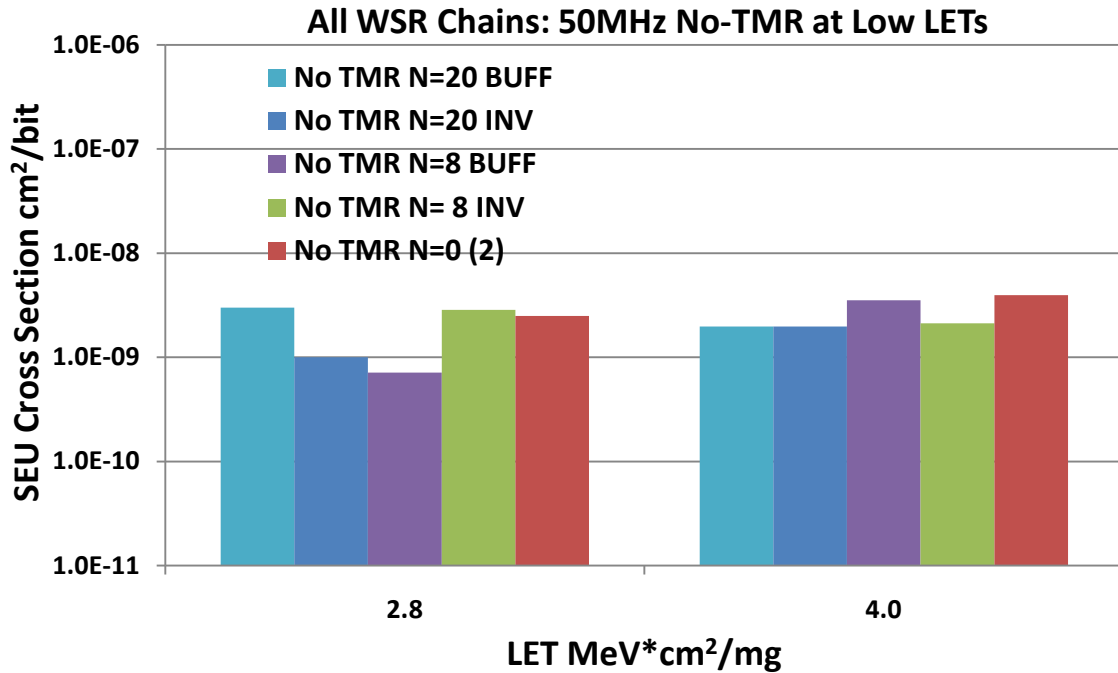


Figure 30: No TMR for all WSR chains: 50MHz Operational Frequency with Checkerboard Pattern. Due to time limitation, No TMR designs at LET > 4.0 MeV*cm²/mg were not tested.

9.2.2 No-TMR Counters

SEU Cross sections are calculated in order to perform error rate predictions per orbit. As previously mentioned in the introduction, Shift-register strings have been the traditional FPGA design used for radiation testing. It is well understood that implemented designs for space-flight projects are significantly more complex than shift registers strings. Hence, the purpose of testing counters is to investigate how design-complexity affects SEU Cross sections. Such a study assists in extrapolating SEU cross sections from tests circuits to target space-flight designs.

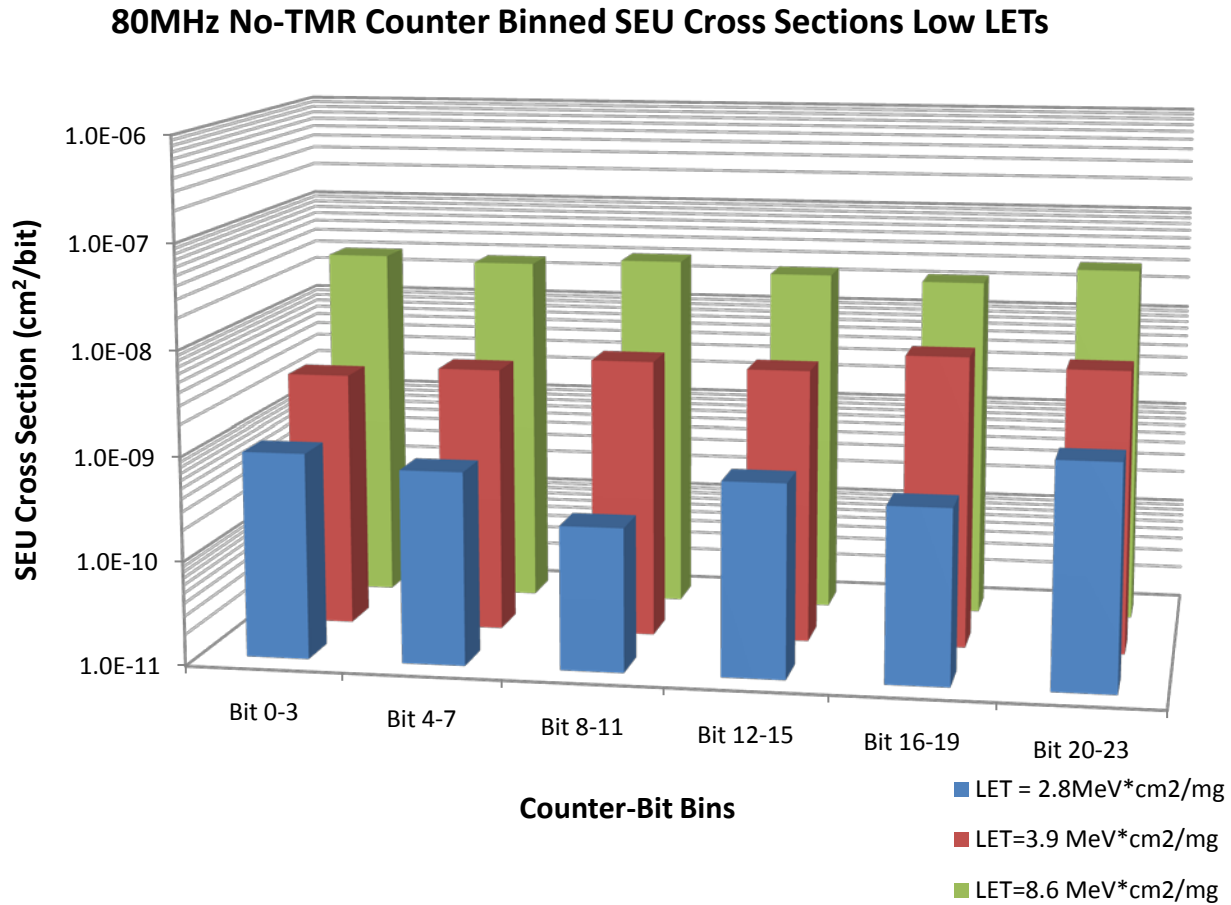


Figure 31: No TMR Counter bits. Bits are placed into bins of 4. Average Cross Section per bit-bin is illustrated

80MHz No TMR Counter Binned SEU Cross Sections Higher LETs

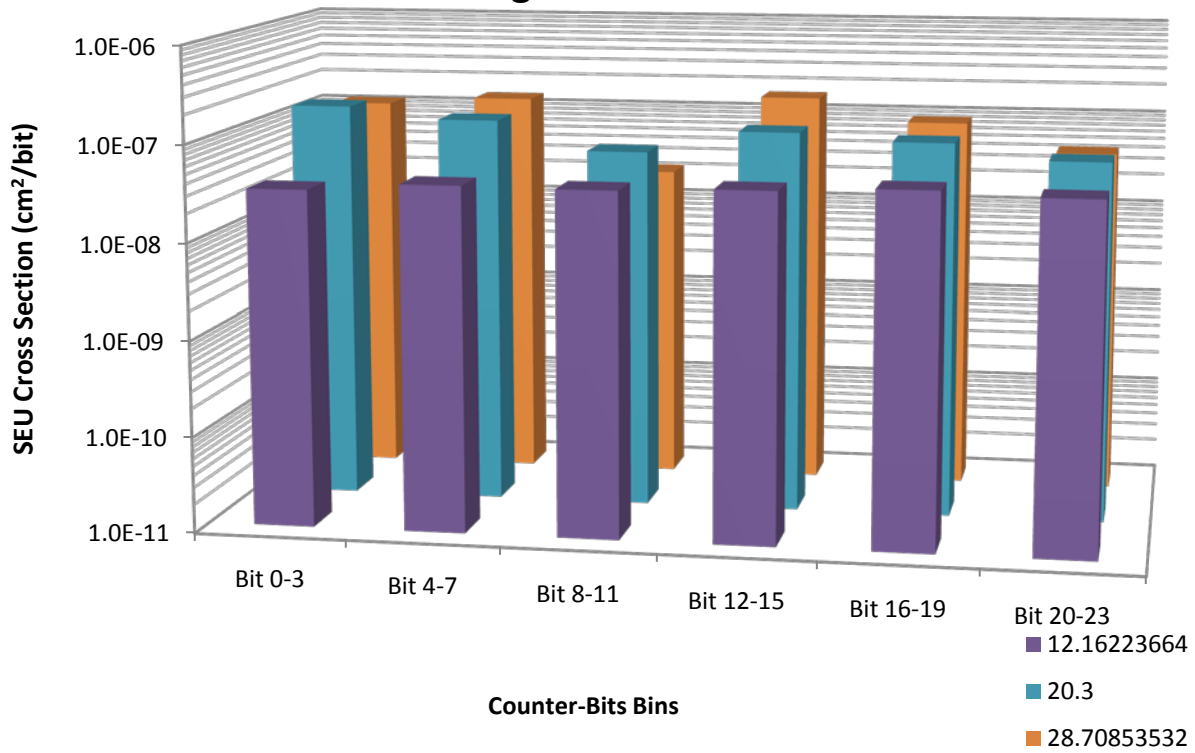


Figure 32: No TMR Counter bits. Bits are placed into bins of 4. Average Cross Section per bit-bin is illustrated. At higher LET values, $P_{SET \rightarrow SEU}$ is more prevalent as shown by the variation in SEU cross sections across bits.

9.3 LTMR SEU Cross Sections

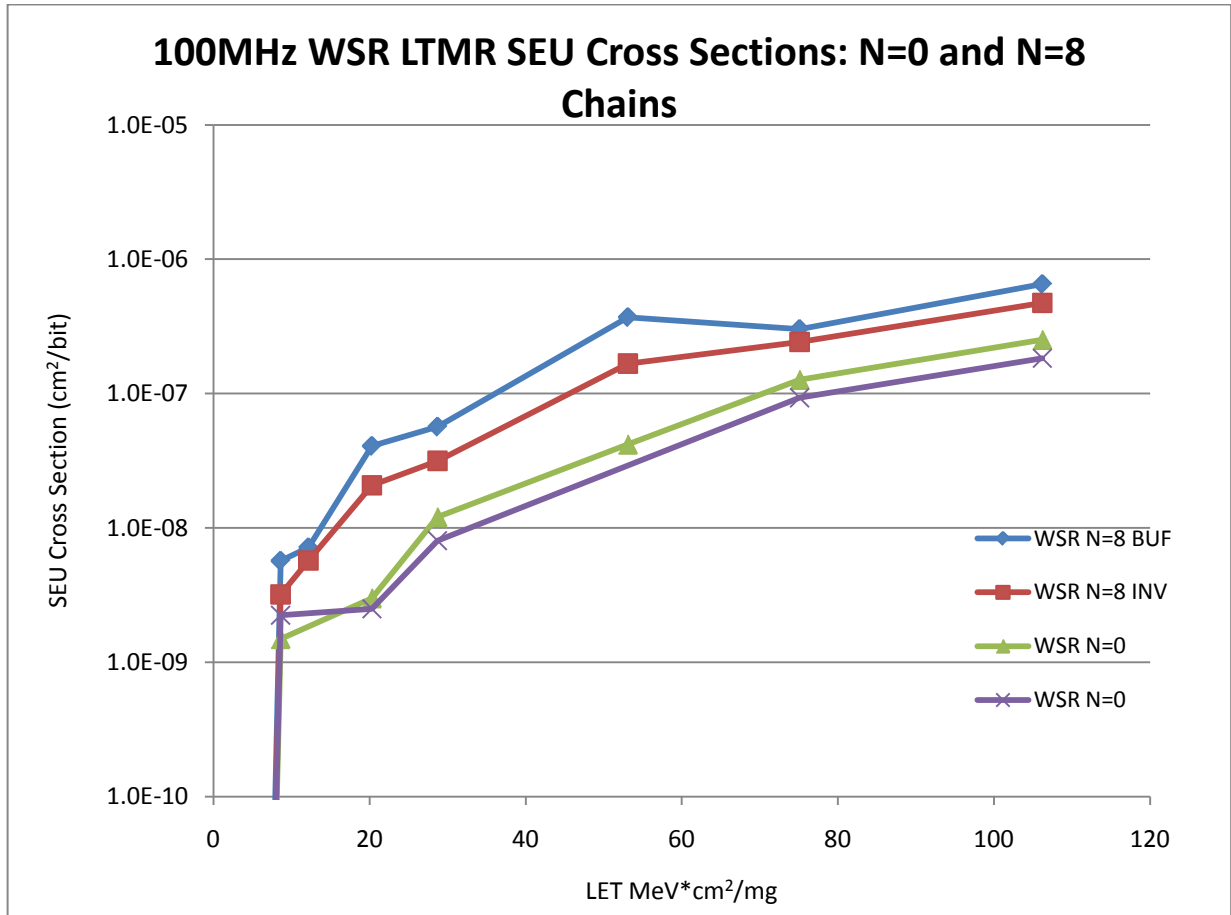


Figure 33: 100MHz Checkerboard pattern LTMR SEU Cross Sections per LET. WSR N=0 and N=8 Chains are shown.

Figure 33 shows the SEU Cross Sections of the LTMR N=0 and N=8 WSR chains across the full range of tested LET values. Chains including Buffers tend to have a slightly higher cross section than chains with inverters. N=0 chains include voters hence they are no longer purely DFF chains, i.e. there is a combinatorial logic block between each DFF. It is expected that the voter masks upsets from the triplicated DFF, however has its own contribution to the SEU cross section in the form of the combinatorial $P_{SET \rightarrow SEU}$. The following sections further investigate LTMR SEU cross Sections across variations in design state space.

9.3.1 Effectiveness of LTMR

Shared Data Input. SET can not be voted out and can cause upset in all three DFFs

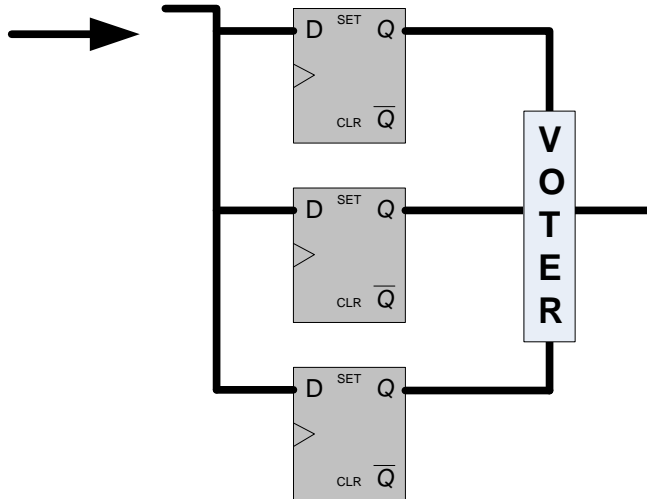


Figure 34: ProASIC LTMR Details. Mitigation is susceptible to SETs because it contains non-redundant combinatorial logic prior to the data input of the LTMR DFFs.

Figure 34 represents LTMR at a DFF. Because the data path is shared by the triplicated DFFs, a transient in the data path will not be able to be voted out. In other words, LTMR is effective for single upsets that occur in DFFs, but does nothing for transients that can get caught by their destination DFFs. Hence the effect of P_{DFFSEU} is reduced.

Capture is essential for a transient to become an upset if the transient is in the data path of a synchronous circuit. Hence, the faster the clock period, the more probable the upset can get caught. In addition, increasing combinatorial logic, should (but not always due to capacitive loading) increase the potential sources of SETs and would in turn increase the SEU cross section. The following investigates such theories.

As shown in Figure 35, using LTMR significantly decreases the cross section for N=0 WSRs (15X decrease). As previously mentioned, one must be aware that once the LTMR is inserted into the N=0 circuit, combinatorial blocks exist (voting is a combinatorial block: (A and B) or (B and C) or (A and C)). For the LTMR N=8 WSRs, the combinatorial logic upsets are the majority and although the LTMR decreased the cross sections, it was not as effective as with the N=0 strings. This is as expected because LTMR only reduces P_{DFFSEU} . Hence as the number of combinatorial logic gates grow, LTMR becomes less effective. Reinforcing LTMR effectiveness, as shown in Figure 35, No-TMR N=0 has a higher SEU cross section than N=8. Post LTMR insertion, N=0 has a lower cross section than N=8.

(14)

ProASIC3E No TMR vs. LTMR Improvement

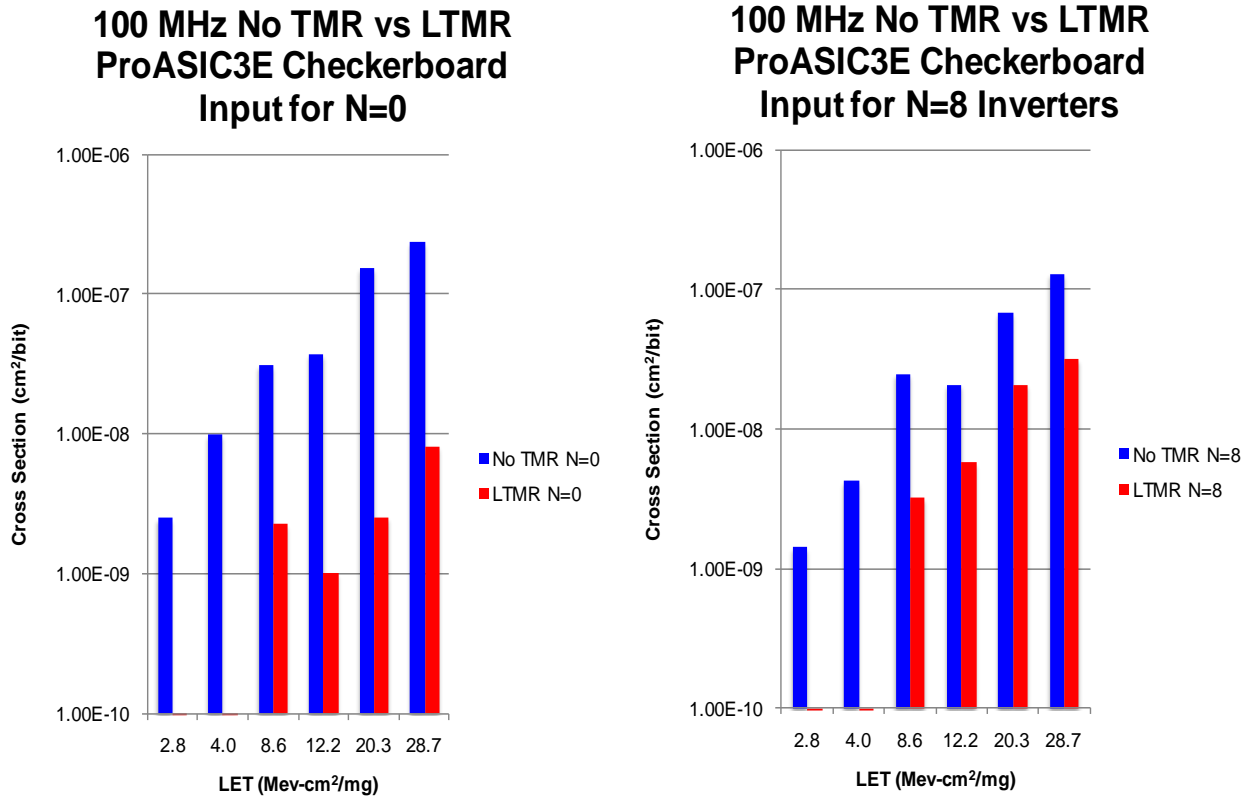


Figure 35: Comparison of 100MHz Checkerboard N=0 and N=8 WSRs with No TMR and LTMR

9.3.2 LTMR and LET Threshold

Using LTMR has successfully increased the LET threshold (LET_{TH}). This in turn reduces the overall error rate. With No-TMR significant upset rates existed at an LET of 2.8. After the insertion of LTMR there is an observable LET_{TH} at 8.6. This is true for all architectures (WSRs and Counters) and is illustrated in Figure 35 and Figure 36.

80 MHz LTMR Counter Cross Sections at Low LET Values

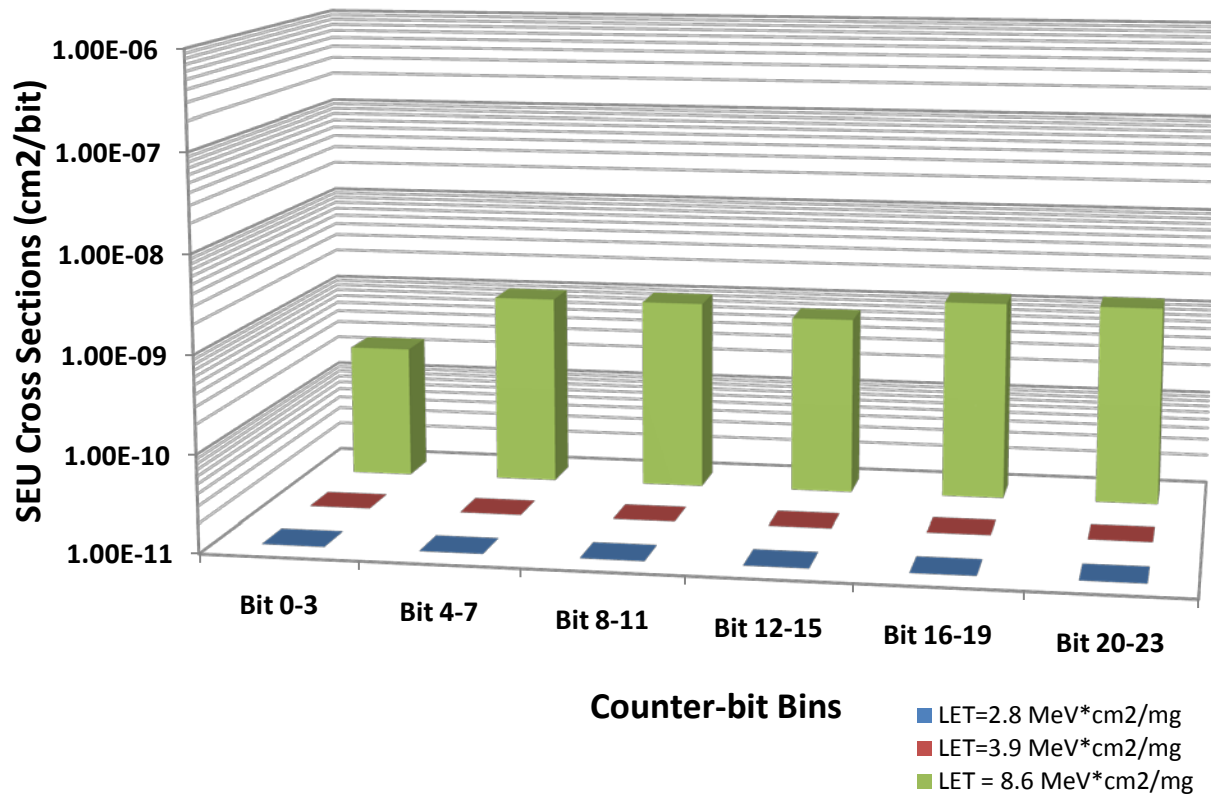


Figure 36: LTMR Counter Array SEU Cross Sections at 80MHz operational frequency. Low LET values are observed. On-set or LET threshold has been increased over No-TMR.

9.4 SET Propagation with LTMR Designs

As previously mentioned, $P_{SET \rightarrow SEU}$ is the most significant source of error in a LTMR circuit. More specifically, due to the global routes (P_{SEFI}), $P_{SET \rightarrow SEU}$ is most significant at lower LET values, P_{SEFI} has more of a contribution at higher LETs.

In order for $P_{SET \rightarrow SEU}$ to occur, a transient must be generated, propagate, and then be captured by the destination DFF. Capacitive loading due to routes or gate transfer functions can reduce the size of the transient and hence reduce $P_{SET \rightarrow SEU}$. Research in ASIC devices have shown the possibility of transient widths increasing, hence increasing $P_{SET \rightarrow SEU}$. The following figures (Figure 38, Figure 39, and Figure 40) show SET propagation in LTMR circuits.

9.4.1 LTMR: SET Generation, Propagation, and Capture in RTAX-S Devices

As previously mentioned, with LTMR insertion at each DFF, the most prevalent upset probability factor is $P_{SET \rightarrow SEU}$. In order for $P_{SET \rightarrow SEU}$ to exist, a transient with a width of τ_{width} must be generated (with a probability of $P_{generate}$), propagated (with a probability of $P_{propagate}$), and then captured by a destination DFF. In order for the SET to be captured, it must arrive during the destination DFF's clock edge with a probability proportional to the width of the transient multiplied by f_s . Consequently, $P_{SET \rightarrow SEU}$ is frequency dependent and has operational frequency (f_s) as a parameter ($P(f_s)_{SET \rightarrow SEU}$).

(15)

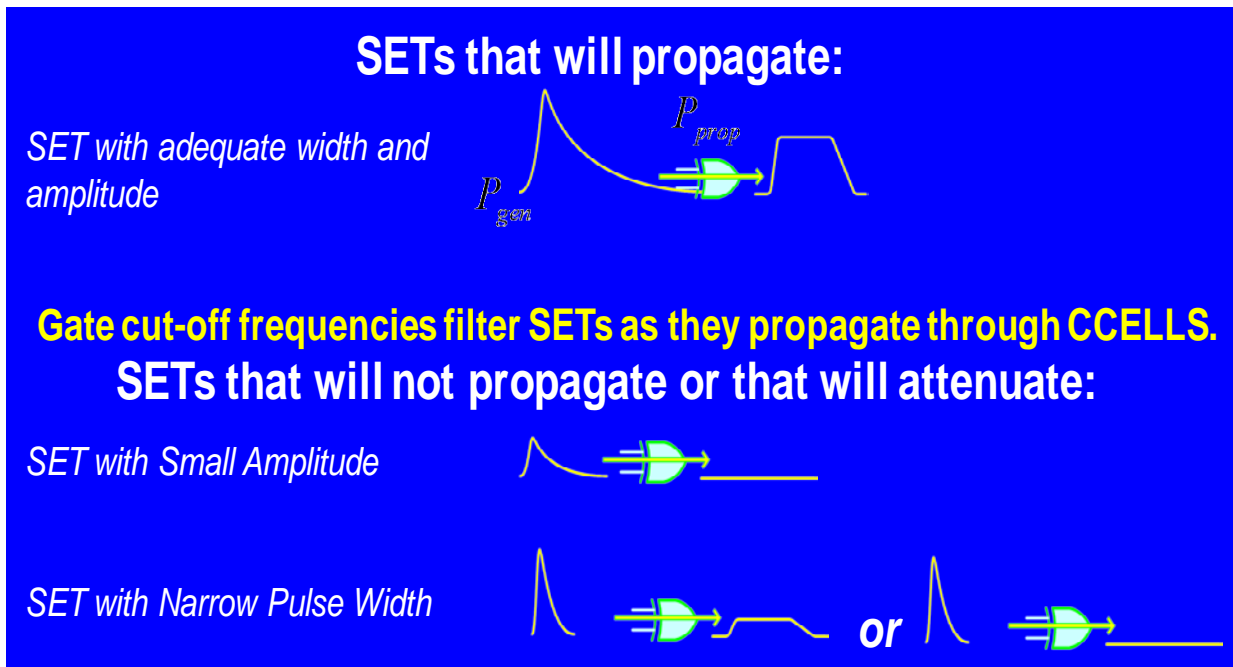
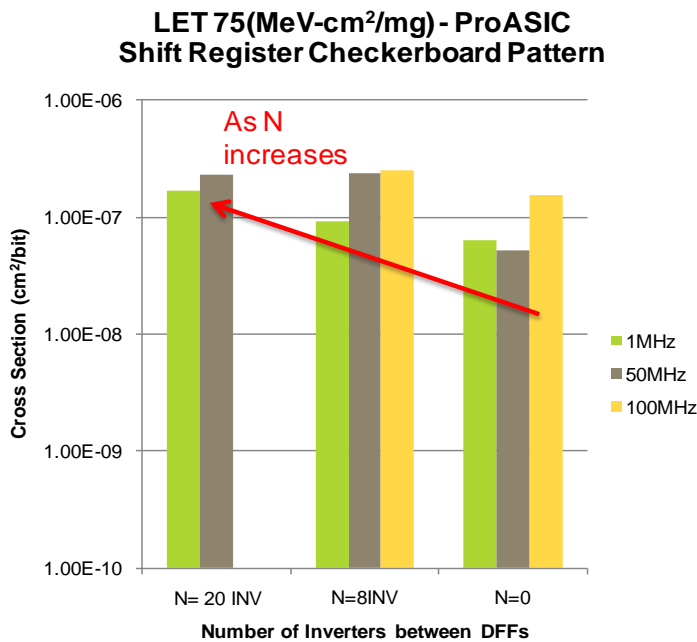


Figure 37: SET Propagation through Combinatorial logic gates

If transients are not filtered during propagation, see Figure 37, they have potential to reach their destination DFF and be active during the clock edge. In addition, as the number of combinatorial logic blocks increase, so does $P_{generate}$. If $P_{propagate}$ is high, then it follows that as the number of combinatorial blocks increase so will $P_{SET \rightarrow SEU}$. For LTMR circuits $P_{SET \rightarrow SEU}$ is the dominant source of error and hence an increase in $P_{SET \rightarrow SEU}$ essentially increases the overall cross section of the design.

Figure 38 through Figure 40 illustrate for WSRs with a given frequency and given LET, as N (number of combinatorial logic blocks between DFFs) increases so does the overall SEU cross section.



- As Frequency increases, Cross section increases
- As N increases, Cross section increases

Figure 38: $P_{SET \rightarrow SEU}$ with LET=75 for Various WSR Strings Containing Different Levels of Combinatorial Logic Cells. (N=0: No Combinatorial logic; N=8: 8 Levels of Combinatorial logic blocks between DFFs; N=20: 20 Levels of Combinatorial Logic Blocks between DFFs)

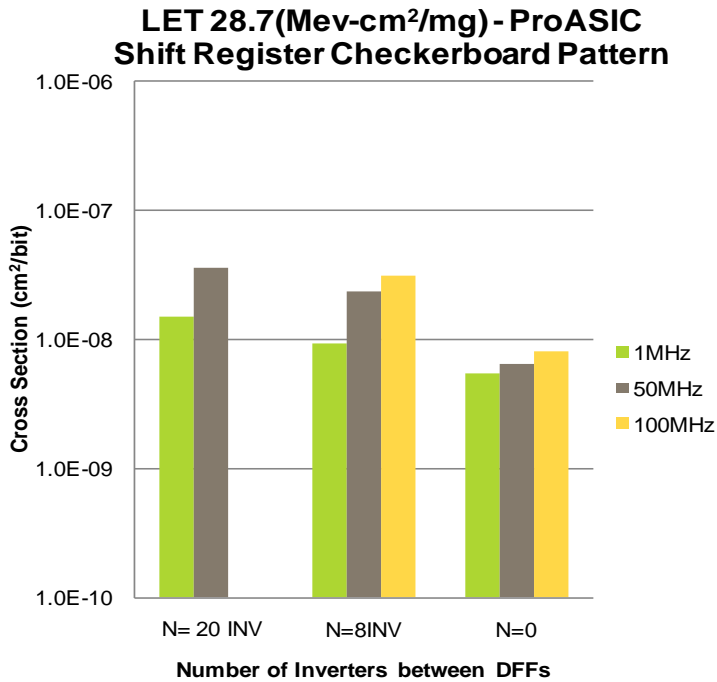


Figure 39: $P_{SET \rightarrow SEU}$ with LET=28.7 for Various WSR Strings Containing Different Levels of Combinatorial Logic Cells. (N=0: No Combinatorial logic; N=8: 8 Levels of Combinatorial logic blocks between DFFs; N=20: 20 Levels of Combinatorial Logic Blocks between DFFs)

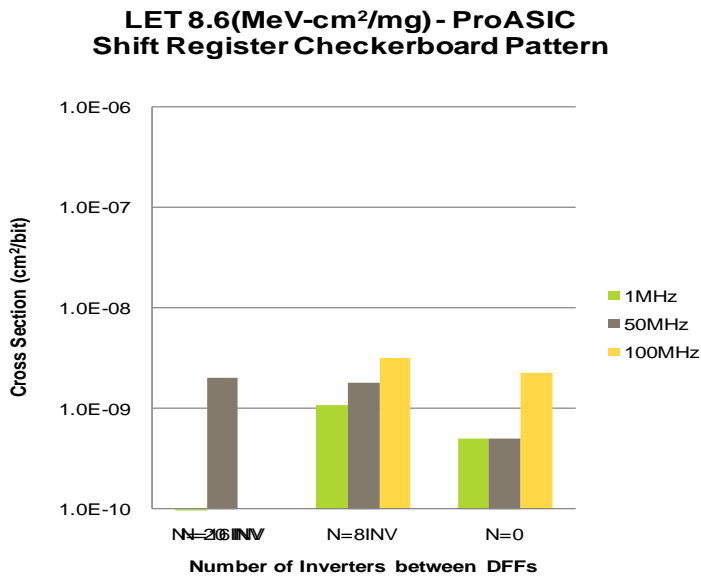


Figure 40: $P_{SET \rightarrow SEU}$ with LET=8.6 for Various WSR Strings Containing Different Levels of Combinatorial Logic Cells. (N=0: No Combinatorial logic; N=8: 8 Levels of Combinatorial logic blocks between DFFs; N=20: 20 Levels of Combinatorial Logic Blocks between DFFs)

9.4.2 LTMR Data Pattern Effects

In order to investigate data pattern effects, comparisons are made between static data inputs (such as all 0's) and checker board (data changing every clock cycle). Static data input yielded significantly lower error cross sections than the alternating data pattern for all N=0 WSRs at higher frequencies as shown in Figure 41. However, N=8 WSR SEU Cross sections did not significantly differ when comparing static and checkerboard patterns. It should be noted that WSR N=8 contain enables while N=0 do not. The enable logic adds another dimension of upsets to the strings. It is possible that with LTMR (P_{DFFSEU}) has been decreased. However, with N=8 enables they are global routes that cannot be mitigated and increase P_{SEFI} . As P_{SEFI} is increased it becomes closer to $P_{\text{SET} \rightarrow \text{SEU}}$, and hence contributes more so to the cross section. In this P_{SEFI} can mask the data pattern effects as in Figure 42.

It should be noted that due to time limitations, no static 0 data patterns were tested. Hence the data only represent LETs from $53 \text{MeV} \cdot \text{cm}^2/\text{mg}$ to $106 \text{MeV} \cdot \text{cm}^2/\text{mg}$. More analysis should be performed for better statistics.

In summary:

N=0 WSRs do not contain enables. Hence $P_{\text{SET} \rightarrow \text{SEU}}$ is most significant and data pattern effects exist: checkerboard has a higher rate than static.

N=8 WSRs contain enables. Hence P_{SEFI} can mask the data pattern effects of $P_{\text{SET} \rightarrow \text{SEU}}$. In this case, there is not a significant difference between checkerboard and static.

More of the effects of global enable routes and P_{SEFI} will be discussed in the DTMR and Burst sections.

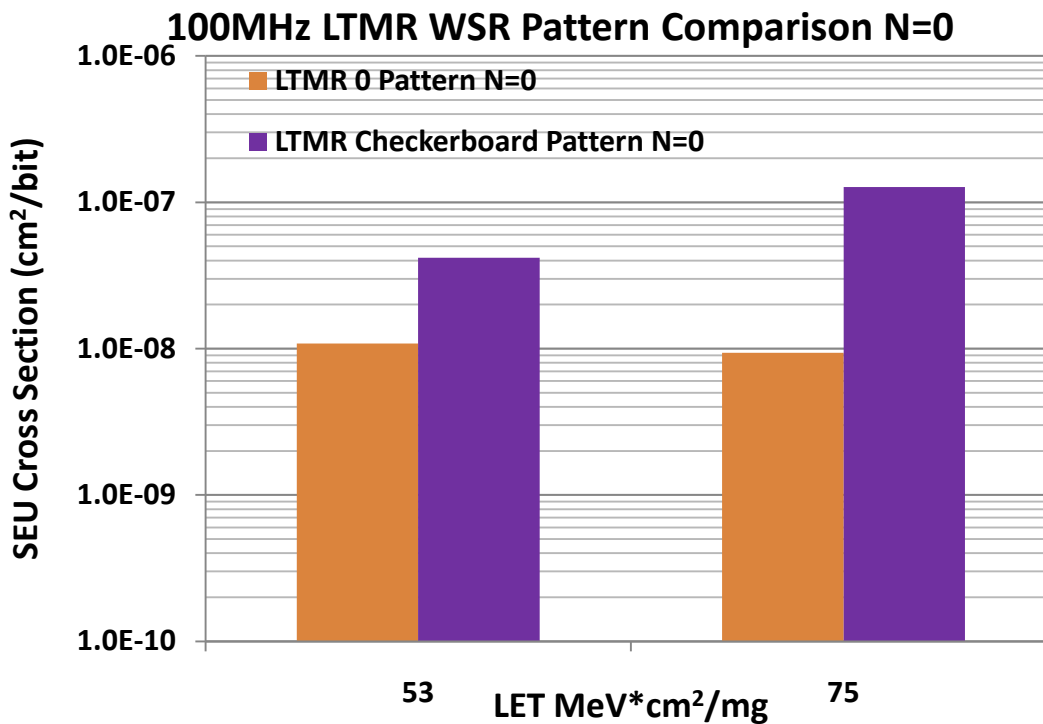


Figure 41: 100MHz LTMR WSR Data Pattern Comparisons. Checkerboard versus Static 0 Pattern for N=0 Chains. Checkerboard has a significantly higher cross section over LET values.

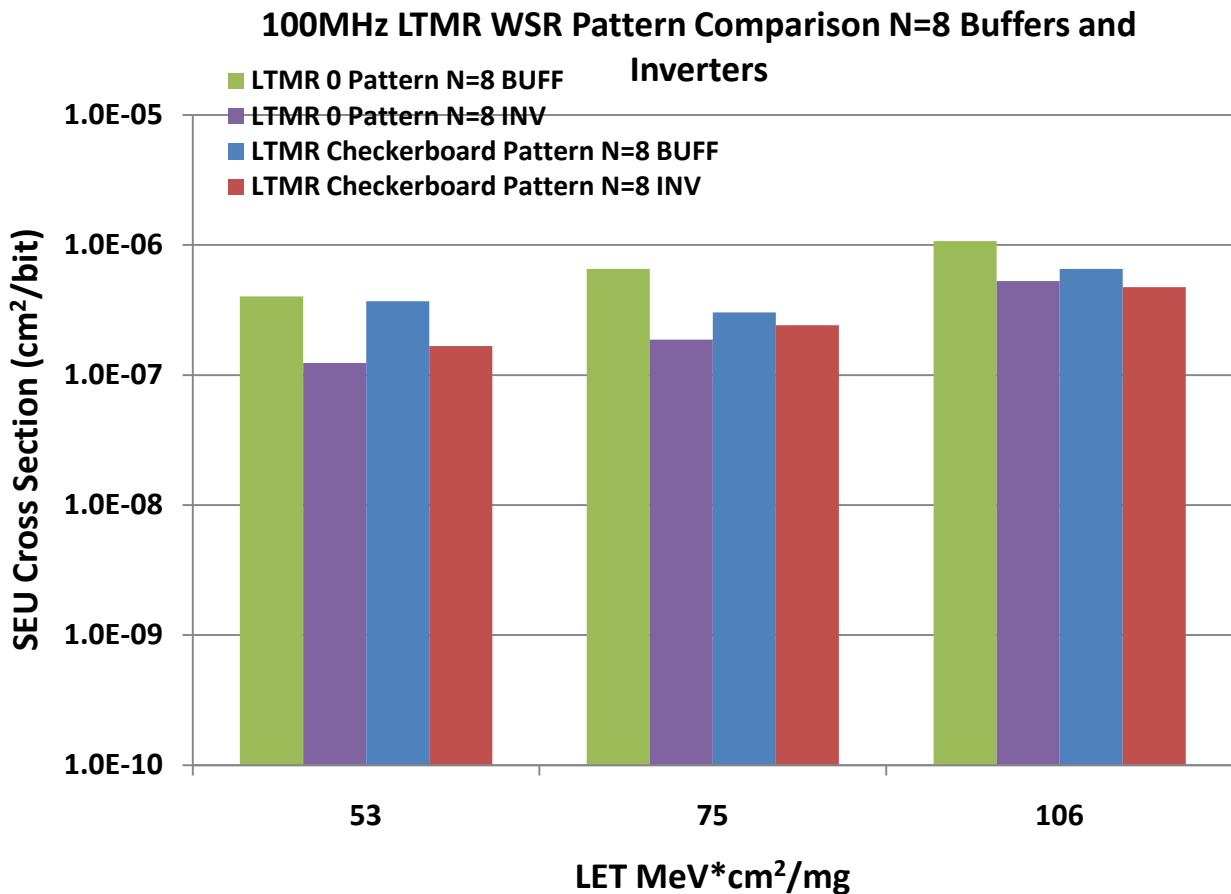


Figure 42:100MHz LTMR WSR Data Pattern Comparisons. Checkerboard versus Static 0 Pattern for N=8 Chains. Cross sections are not significantly different, however, 0-pattern with buffers seem to consistently have higher cross sections than other strings across LET.

9.4.3 LTMR Architectural Effects: Counters

It is shown in Figure 43 through Figure 45, that at low (8.6 MeVcm²/mg) to mid LET values (20.3 MeVcm²/mg), data pattern and combinatorial block effects exist. This is expected because, as previously mentioned, LTMR at low LET is dominated by P_{SET→SEU}. Explanation of data pattern and combinatorial logic blocks:

Data pattern: With the counter, the lower order bits (LOB) have the fastest data pattern as opposed to the upper order bits. Hence, the LOB have the highest cross sections. It is shown in in Figure 43 through Figure 45 that the SEU cross sections decrease as the bin bit orders increase.

Combinatorial: With the counter, the Higher order bits (HOB) has an increase in combinatorial logic blocks. Towards the mid-range of the counter bits, it is shown in in Figure 43 through Figure 45, that the SEU cross sections begin to increase. In addition, it is shown in in Figure 43 through Figure 45 for the WSRs that as N increases so does the SEU cross sections.

At the higher LET values, Figure 46 and Figure 47, The SEU cross sections begin to approach each other over all counter bins and WSR strings. This is due to PSEFI beginning to have more of a contribution to the SEU cross sections.

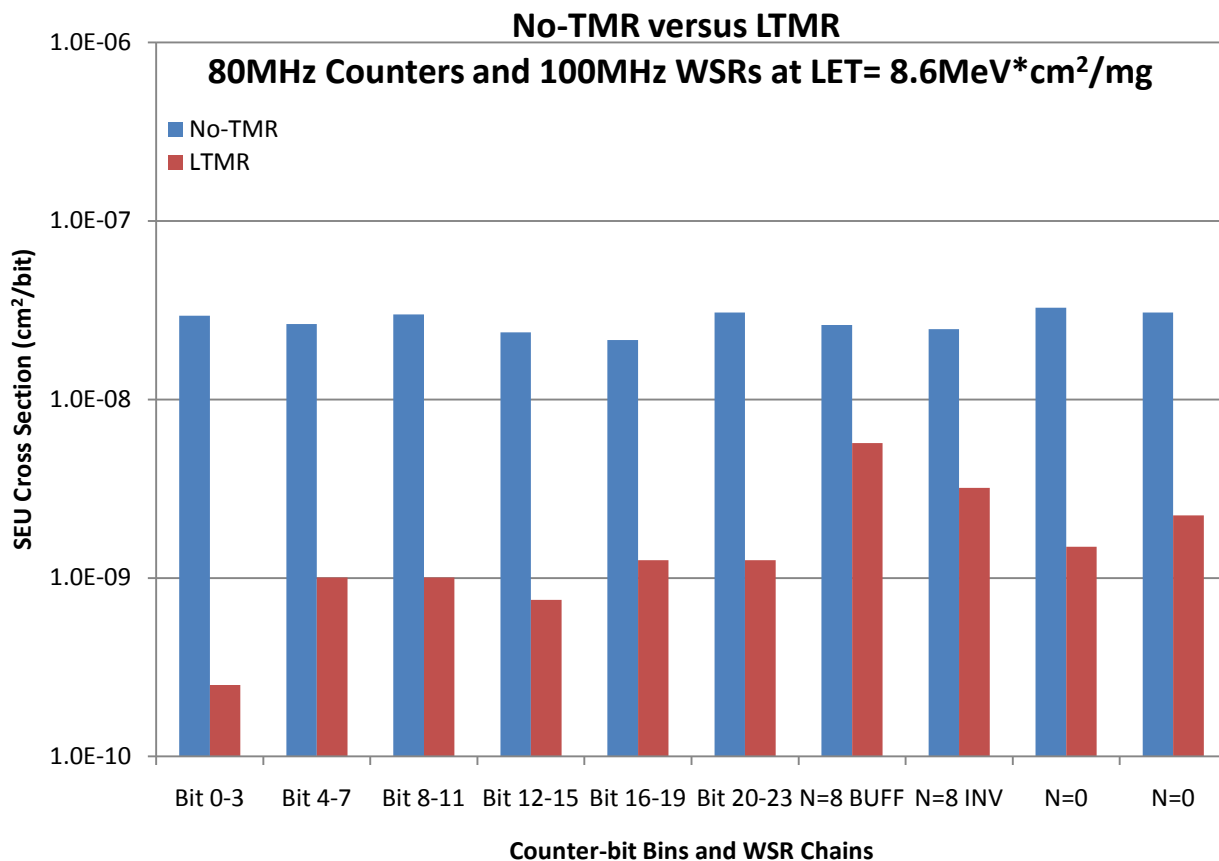


Figure 43: Comparing No-TMR versus LTMR. Left portion of figure shows 80MHz Counters. Right most portion of figure shows 100MHz WSRs.

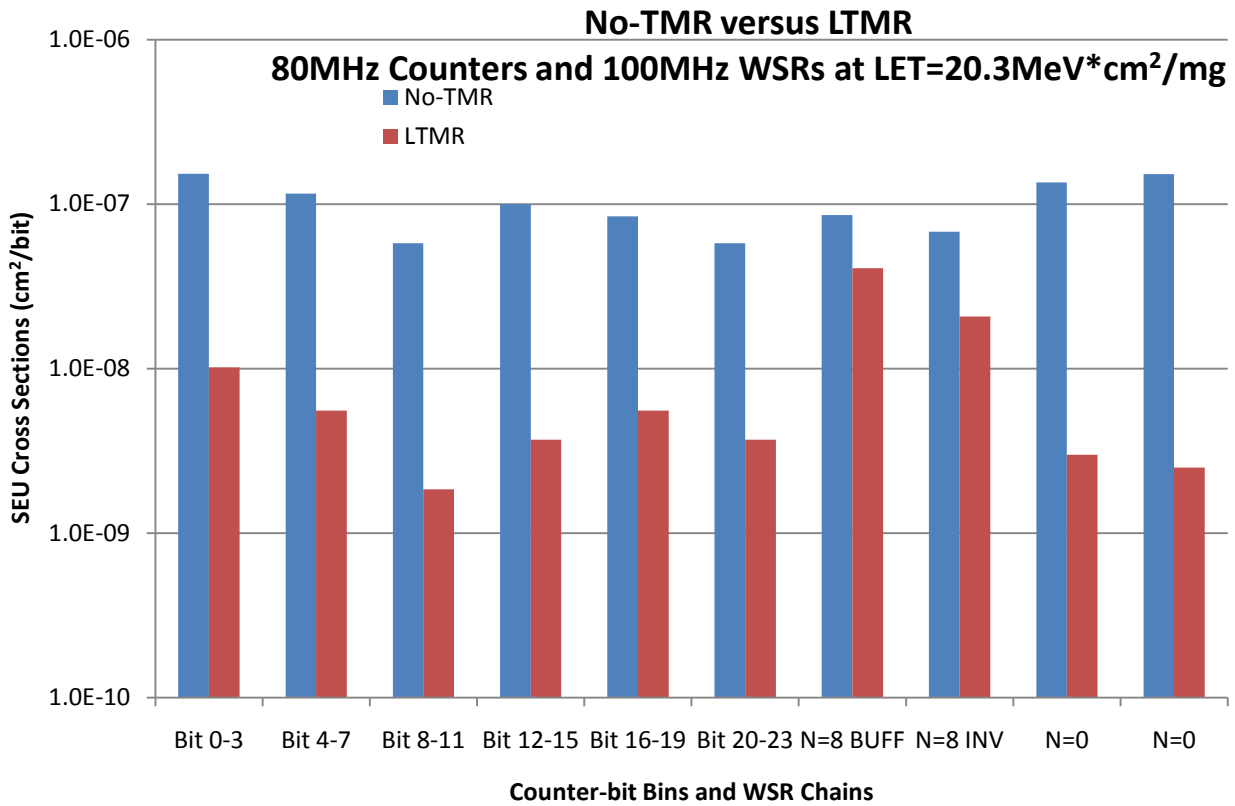


Figure 44: No-TMR Versus LTMR: 80MHz Counters and 100MHz WSR Chains at LTMR LET_{TH}=20.3 MeV*cm²/mg. Counter SEU Cross sections are within the N=0 to N=8 SEU Cross section range

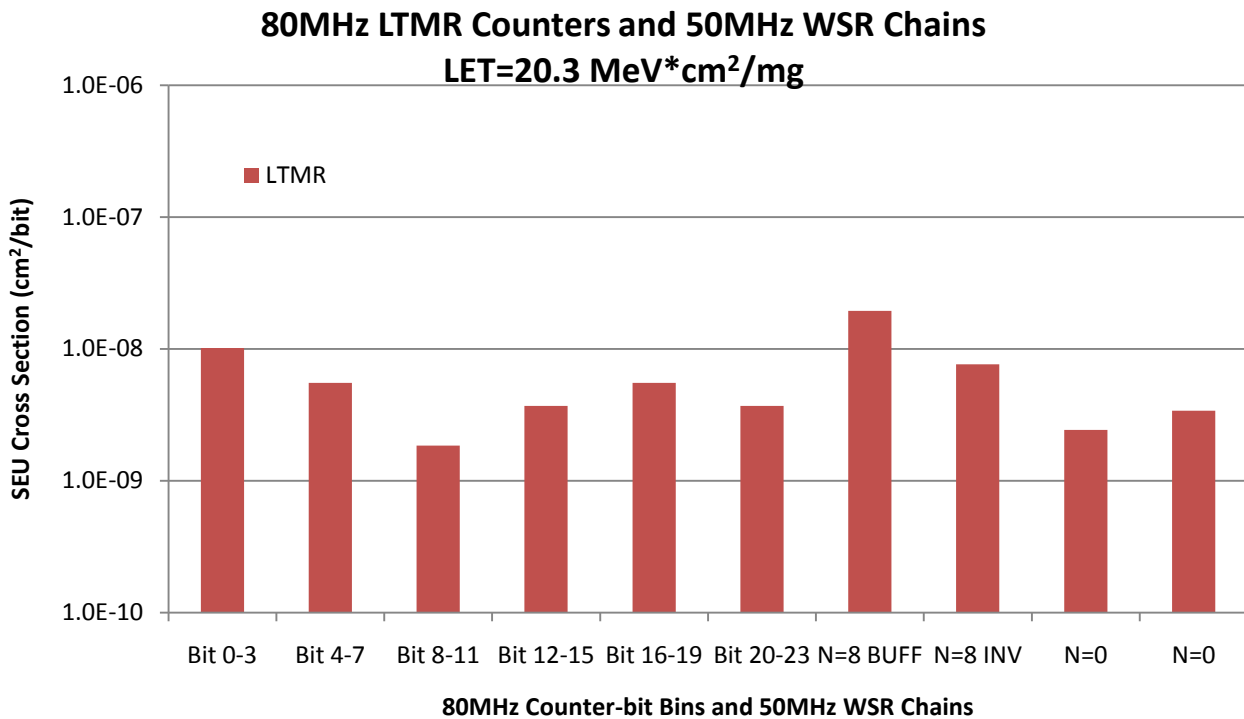


Figure 45: LTMR: 80MHz Counters and 50MHz WSR Chains at LTMR LET_{TH}=20.3 MeV*cm²/mg. SEU Cross Sections of counters and WSR strings begin to approach each other.

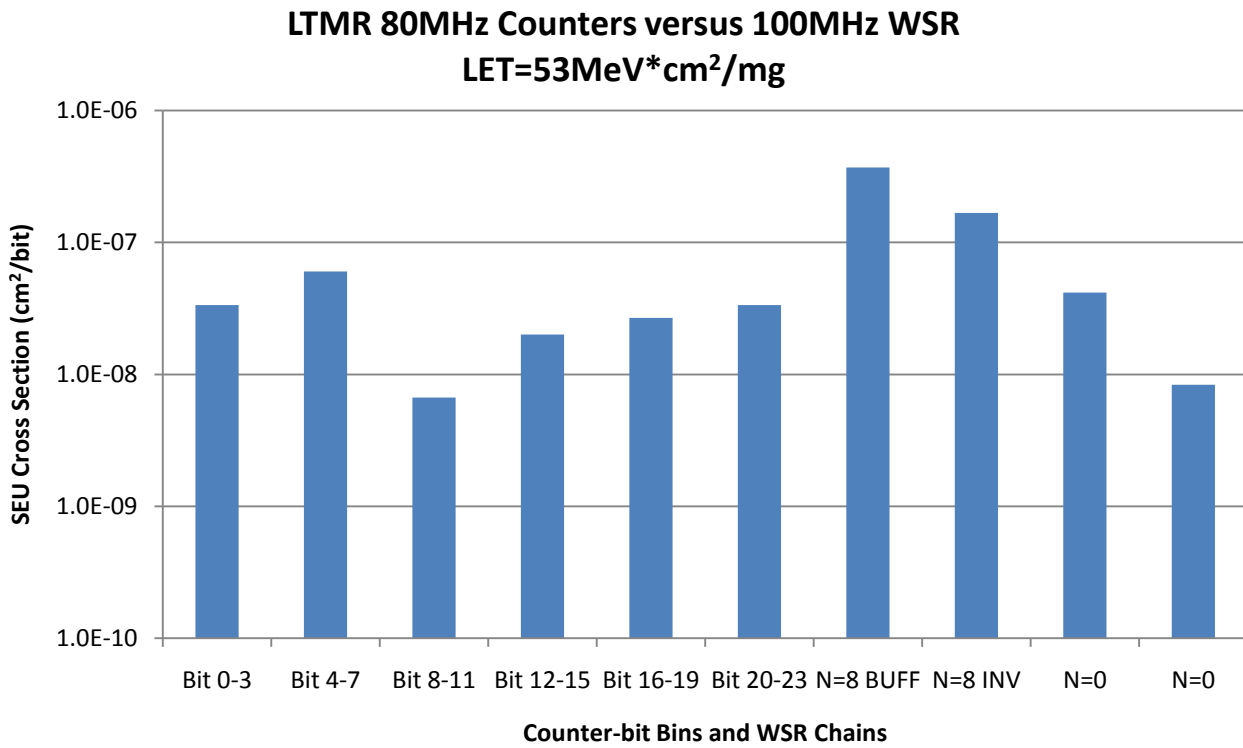


Figure 46:LTMR: 80MHz Counters and 100MHz WSR Chains at LTMR LET_{TH}=53.1 MeV*cm²/mg. Counter SEU Cross sections are within the N=0 to N=8 SEU Cross section range

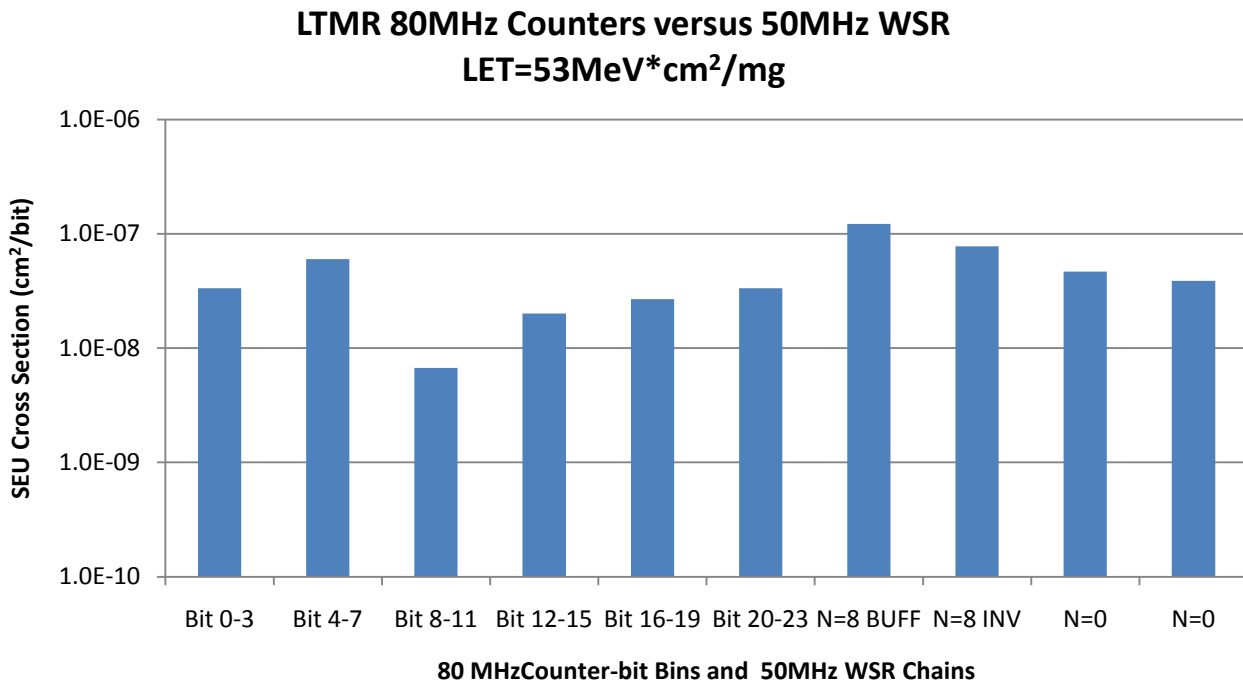


Figure 47: LTMR: 80MHz Counters and 100MHz WSR Chains at LTMR LET_{TH}=53.1 MeV*cm²/mg. SEU Cross Sections of counters and WSR strings begin to approach each other.

9.5 Global Routes: DTMR and P_{SEFI}

As previously mentioned, the ProASIC FPGA is a commercial device. Hence, no RHBD was placed in the

device by the manufacturer. This study concentrated on the effectiveness of designer implemented RHBD within a commercial FPGA. Two of the RHBD methods investigated is LTMR and DTMR. The dilemma with LTMR and DTMR is that each scheme has portions of the design un-mitigated. GTMR, (although expensive), is a proven method for masking and correcting most single-bit upsets that can incur within a design. Difficulty arises because GTMR requires 3 separate clock domains that are logically treated as one synchronous domain. Voter comparisons are performed across clock domains and feed DFFs in all three clock domains. This is a direct violation of synchronous design rules, hence requires special attention to successfully implement. In order to implement GTMR, the FPGA (or ASIC) must contain clock trees that have minimal skew within each tree and minimal skew between each tree. In more detail, the clock skew (CLK_{skew}) must be less than the shortest path delay (τ_{delay}) from DFF to DFF which includes DFF feedback paths.

(16)

The clock trees within the ProASIC device have minimal clock skew within a tree. However, between trees, the clock skew is too large to implement GTMR. Subsequently, only LTMR and DTMR have been implemented. All designs use clocks, resets, and some use global enable pins. Global routes are not mitigated in LTMR and DTMR, hence they are single points of failure. In addition, because of the global nature of the routes, a single transient can cause many gates (or DFFs) to fail and can cause the entire system to be inoperable for consecutive cycles.

A burst in this document is defined by the outputs of the design under-analysis being in a faulty state for consecutive cycles. It follows that, any upset that is not in burst, corresponds to a single cycle upset.

9.5.1 DTMR Single Cycle Upsets

As previously mentioned, tests were also performed with DTMR insertion. With LTMR, although all DFFs are triplicated circuits the Data inputs (D) are shared throughout the designs and are hence single points of failure. It is also important to note that the global routes such as Clock (C), Enable (E), and Reset (R) were also shared and are single points of failure. Global routes in this document contribute to P_{SEFI} . With DTMR, C, E, and R are shared... i.e. all global routes are shared. However, the data paths are not. Because the DFFs are triplicated and data paths are not shared, P_{DFFSEU} and $P_{SET \rightarrow SEU}$ are both reduced. Hence, the most dominant source of SEUs stem from P_{SEFI} .

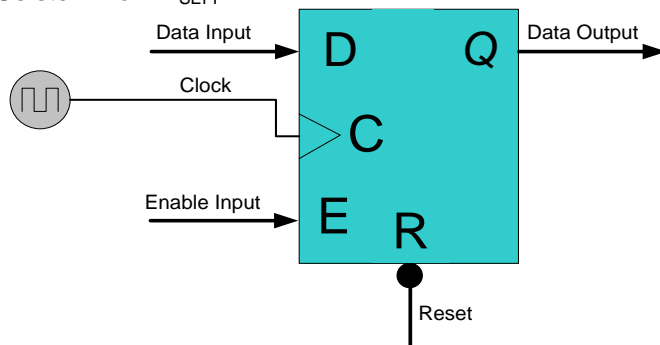


Figure 48: DFF with Data Input (D), Clock Input (C), Enable Input (E), Reset $\text{\textcircled{R}}$, and Data Output (Q).

(17)

Designs that minimize the amount of global routes in the ProASIC and that use DTMR will reduce P_{SEFI} and subsequently reduce the overall SEU cross section.

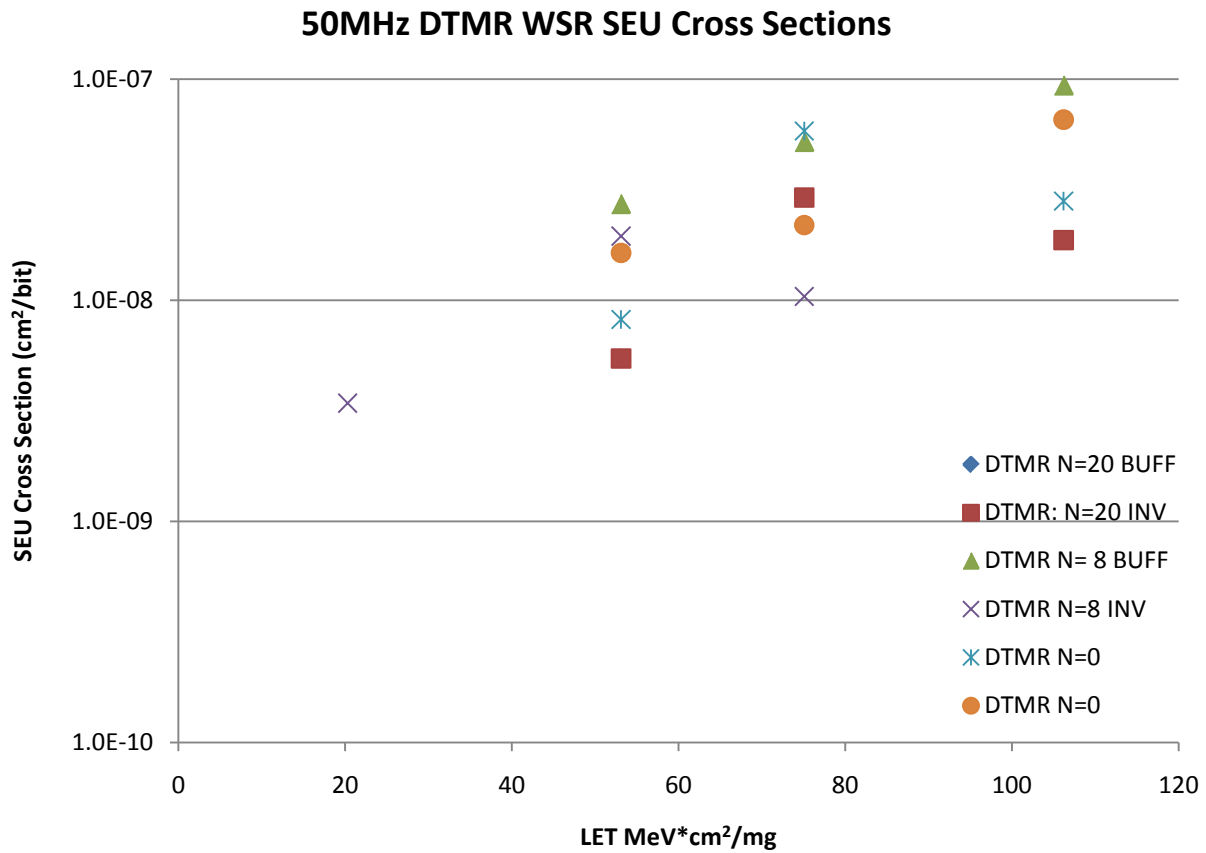


Figure 49: 50MHz DTMR WSR SEU Cross Sections per LET.

Figure 49 illustrates DTMR SEU Cross Section of the WSRs operating at 50MHz. It is interesting that the N=20 strings generally have smaller cross sections. This is due to the fact that the particles that are causing transients in the N=20 data paths are not contributing to the SEU Cross sections due to the DTMR masking of P_{DFFSEU} and $P_{SET \rightarrow SEU}$.

9.5.1.1 DTMR versus LTMR

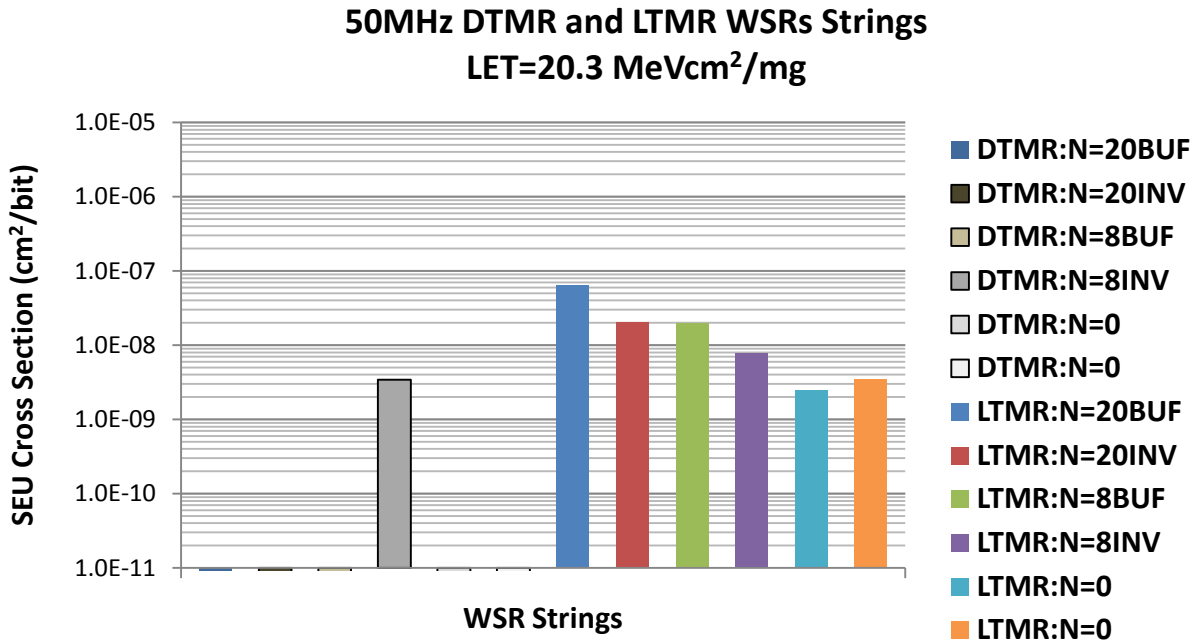


Figure 50: 50MHz Checkerboard WSR DTMR and LTMR Cross Sections at LET = 20.3MeV*cm²/mg. DTMR LET_{TH} near 20MeVcm²/mg for checkerboard data pattern and for a design containing global Clocks, Resets, and Enables.

At higher LET values, DTMR SEU Cross section values approach LTMR SEU Cross section values as illustrated in Figure 51 and Figure 52. This is a result of the cross sections being dominated by global routes or P_{SEFI}.

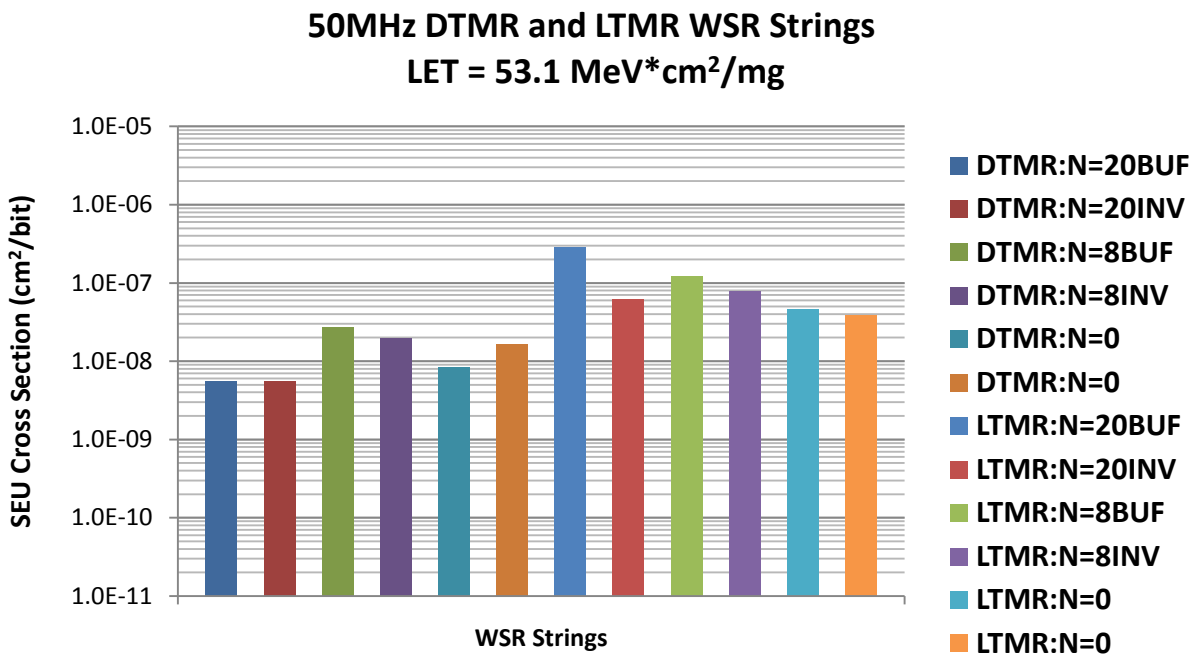


Figure 51: 50MHz Checkerboard WSR DTMR and LTMR Cross Sections at LET = 53.1MeV*cm²/mg. DTMR WSR SEU Cross Sections occupy the left portion of the graph. LTMR occupy the right.

50MHz DTMR and LTMR WSR Strings LET = 106.2MeV*cm²/mg

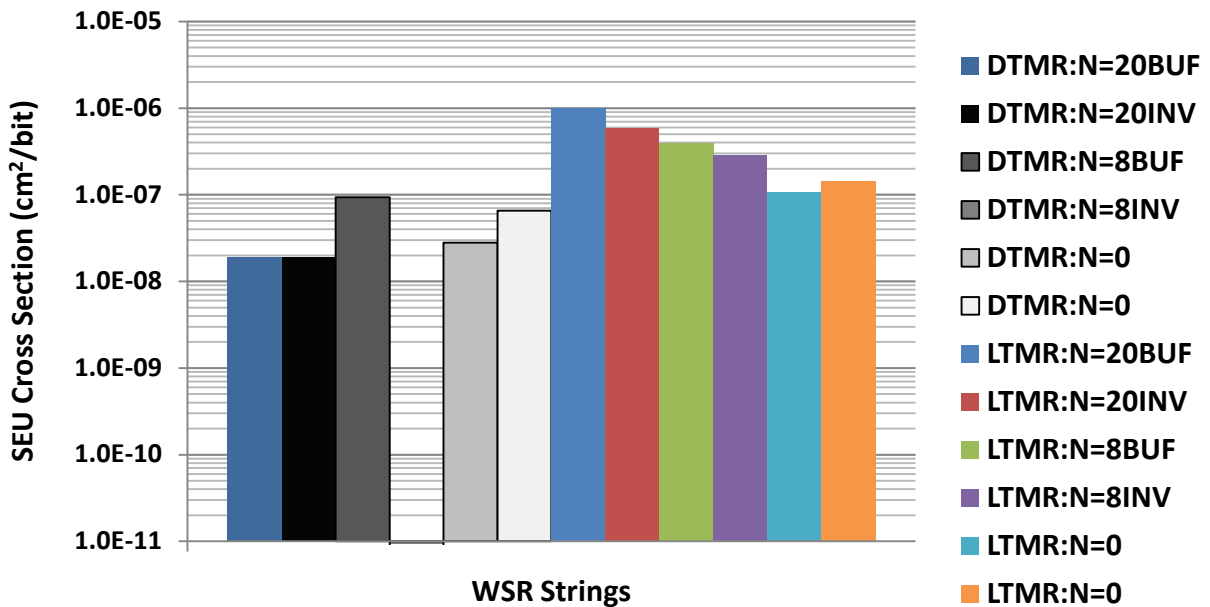


Figure 52:50MHz Checkerboard WSR DTMR and LTMR Cross Sections at LET = 106.2MeV*cm²/mg.

9.5.1.2 DTMR Data Pattern and Frequency Effects

Figure 53 is an illustration of WSR DTMR SEU Cross sections with a static 1 input. It is important to note that the upsets in this figure are only single cycle upsets (no bursts are included in the cross sections). Input is static 1, hence an upset is when an output has gone to a “0” state. Because all combinatorial logic paths and DFFs are mitigated in DTMR, then the upsets can only come from the shared routes (single points of failure) or I/O. The only shared points that exist in the DTMR circuits within the REAG tests are C, R, and E (if E is used). For a static “1” input, a glitch on the clock pin would most likely NOT cause an upset because the same pattern is sampled (data paths are static). The same theory is true for the Enable. Hence the only significant source of upset occurs from the R. If the R incurs a transient, it is possible for the path to go to 0.

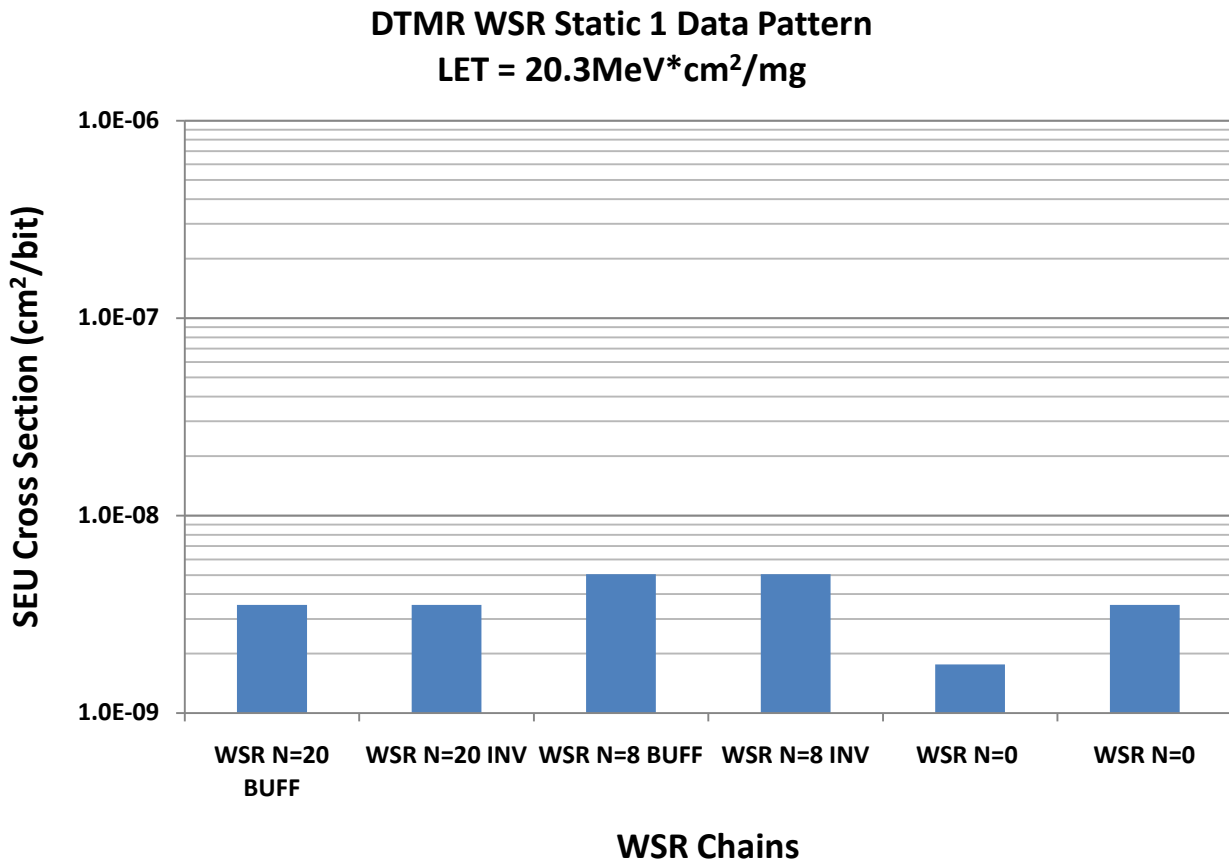


Figure 53: 50MHz WSR DTMR Static 1 Data Pattern at LET_{TH} = 20.3MeV*cm²/mg. The graph represents mostly upsets on Reset Pin. LET_{TH} was observed at 20.3MeV*cm²/mg for DTMR Static 1 pattern.

LET_{TH} for DTMR static-0 data pattern is significantly higher than the LET_{TH} for a DTMR static-1. LET_{TH} is near 75MeV*cm²/mg as illustrated in Figure 54. When analyzing the output data upset signatures, it seems as though the upsets are occurring at the I/O pins with potential cross talk. This assumption is based on the fact that output data should only change once every 4 clock cycles. Any change that occurs within the 4-cycle window, is an upset that originated from either the 4 data output-registers or transients in the I/O buffers/pins.

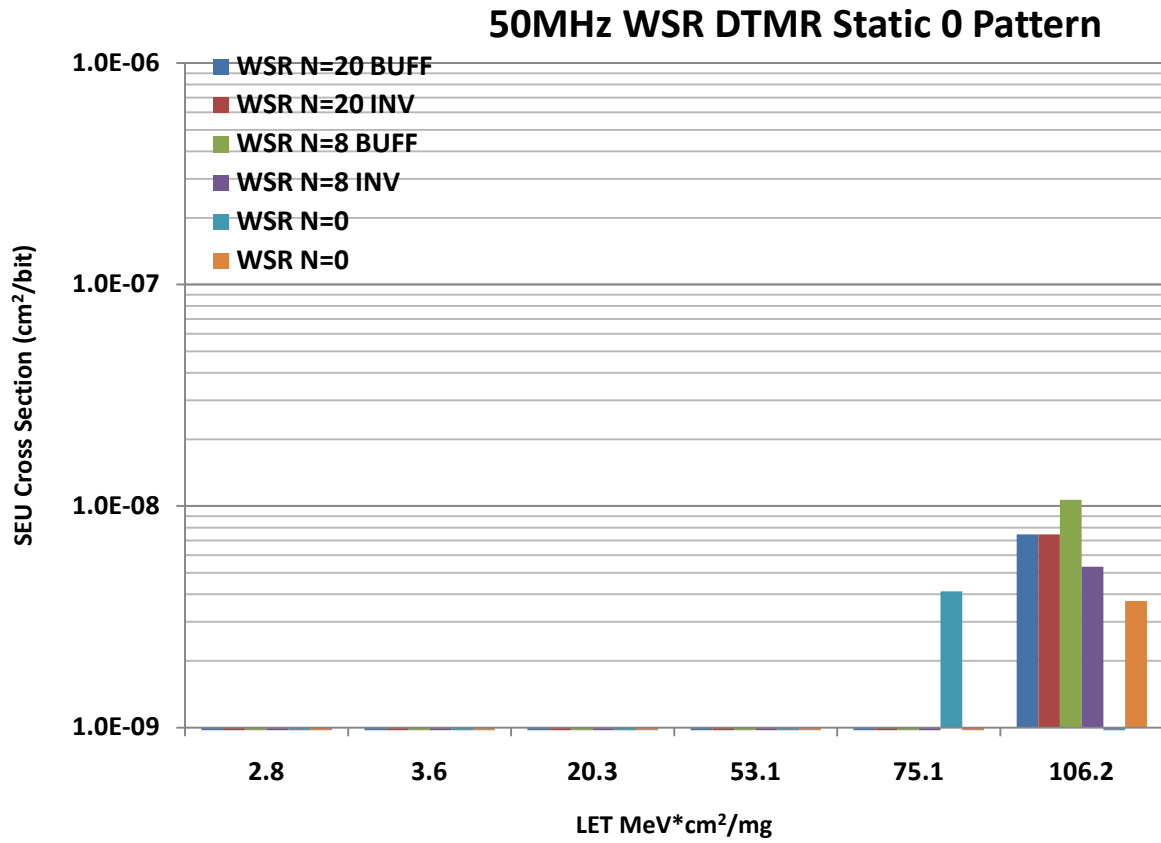


Figure 54: 50MHz WSR DTMR Static 1 Data Pattern. Represents mostly upsets in I/O

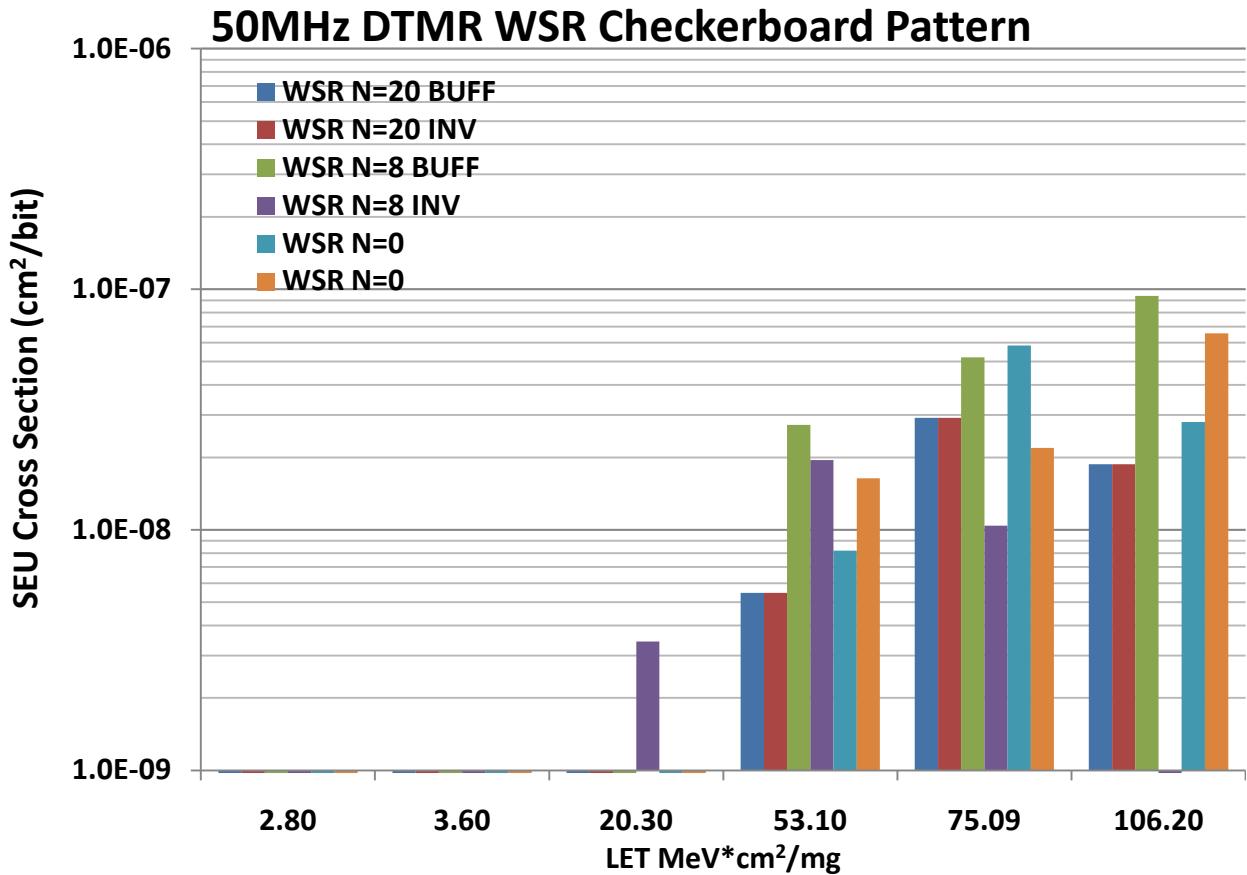


Figure 55: DTMR SEU Cross Sections at 50 MHz. LET_{TH} is close to 20MeVcm²/mg. This is a great improvement over LTMR and No-TMR

Table 23: Global Routes Connected to WSR Chains

	Clock Input: C	Enable: E	Reset: R
N=0 (1)	yes	No	yes
N=0 (2)	yes	No	yes
N=8 INV	yes	yes	yes
N=8 BUFF	yes	yes	yes
N=20 INV	yes	yes	yes
N=20 BUFF	yes	yes	yes

Because the N=0 chains do not utilize the global enable to the DFFs, it should follow that the N=0 chains will have the lowest DTMR cross sections. However, as illustrated in Figure 55, this is not the case. This suggests that the contribution that the E has to P_{SEFI} is minimal.

1MHz DTMR WSR Checkerboard Pattern

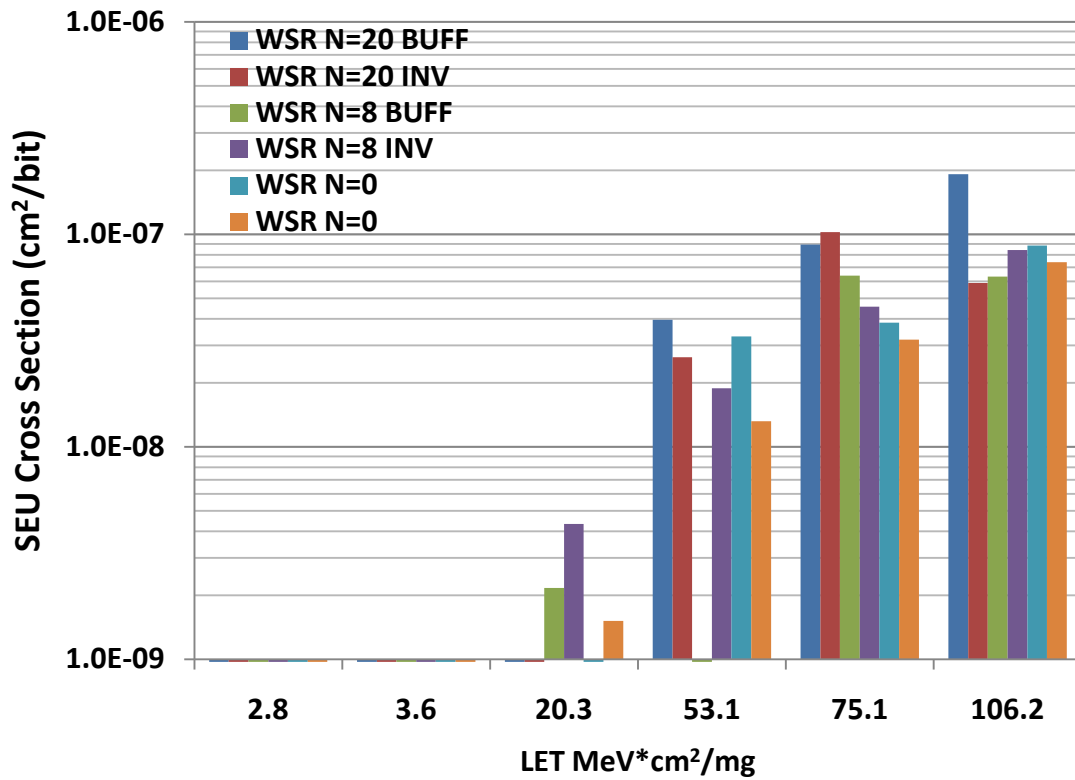


Figure 56: 1MHz DTMR WSR Checkerboard Pattern SEU Cross sections. SEU cross Sections seem to be slightly higher than 50MHz. LETTH for the WSRs remains at approximately 20MeV*cm²/mg.

DTMR WSR Data Pattern Comparisons
 $LET_{TH} = 20.3 \text{ MeV} \cdot \text{cm}^2/\text{mg}$

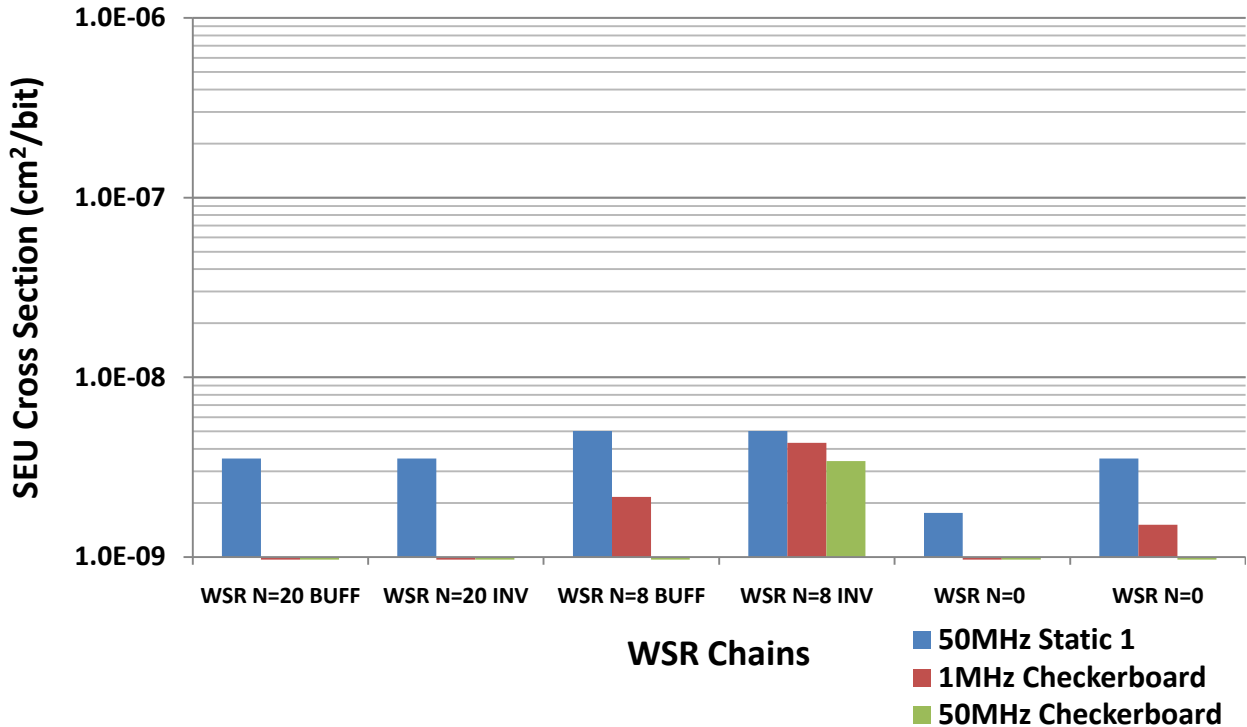


Figure 57: DTMR Data Pattern Comparison at $LET_{TH} = 20.3 \text{ MeV} \cdot \text{cm}^2/\text{mg}$. 50MHz Static 0 Pattern is not shown in graph because no upsets were observed until an $LET = 75 \text{ MeV} \cdot \text{cm}^2/\text{mg}$

9.5.1.3 DTMR Counters

The counters within the counter-array circuit do not contain enable pins. Hence, it is expected for the counters to have similar cross sections as the N=0 WSR chains. It should be noted that the counter snapshot registers do contain global enables. The snap shot registers are not analyzed in this study. One is reminded that the WSRs N>0 do contain enables and they are analyzed and reported. Counter LET_{TH} was observed near 53 $\text{MeV} \cdot \text{cm}^2/\text{mg}$. Although no upsets were observed at $LET = 20.3 \text{ MeV} \cdot \text{cm}^2/\text{mg}$ (as with the WSR strings - Figure 57), it is assumed that LET_{TH} for the counters is also at $LET = 20.3 \text{ MeV} \cdot \text{cm}^2/\text{mg}$. The assumption is being made because fluence was limited during testing and, as shown in Figure 58, the counter cross sections are statistically equivalent to the WSR SEU Cross sections.

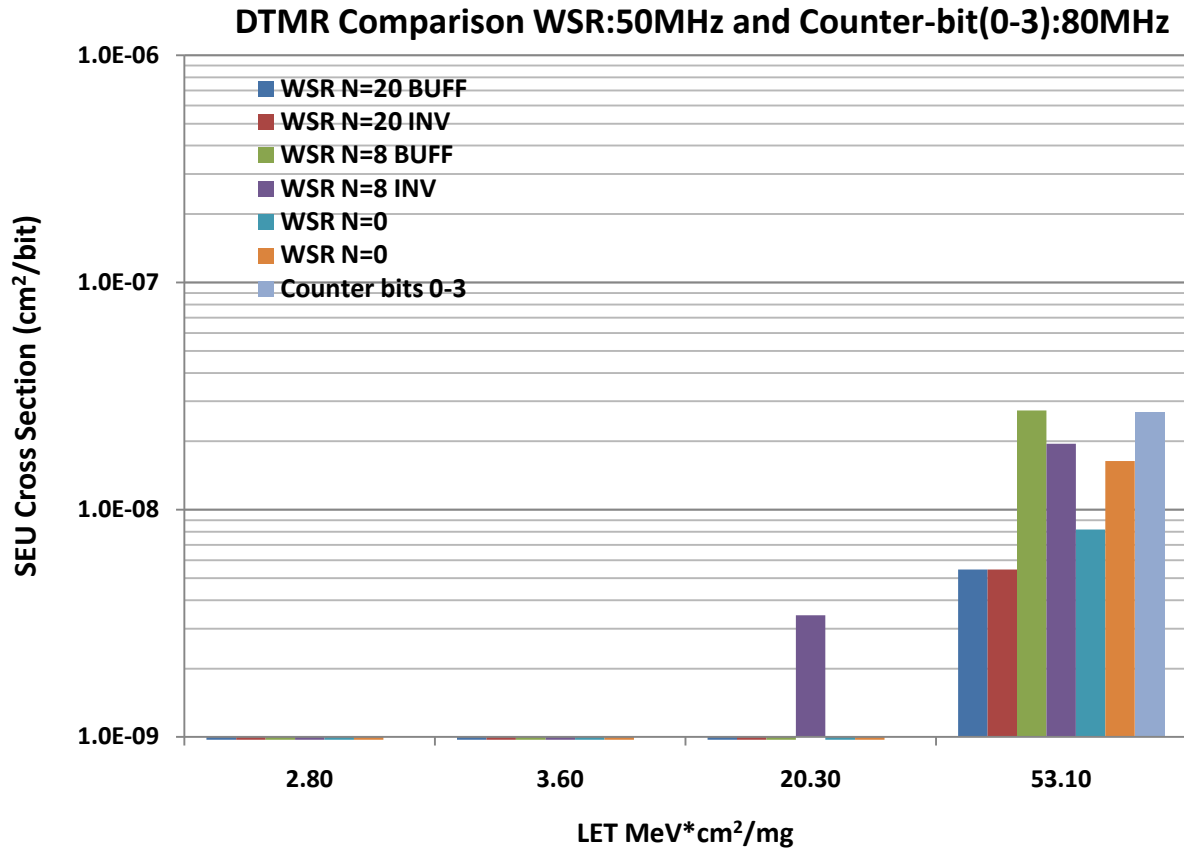


Figure 58: Comparison between Checkerboard WSR and LSB bits0-3 of the Counters. SEU Cross Sections are not significantly different with DTMR insertion. DTMR upsets rely on global nets: Clocks, Resets, and global enables.

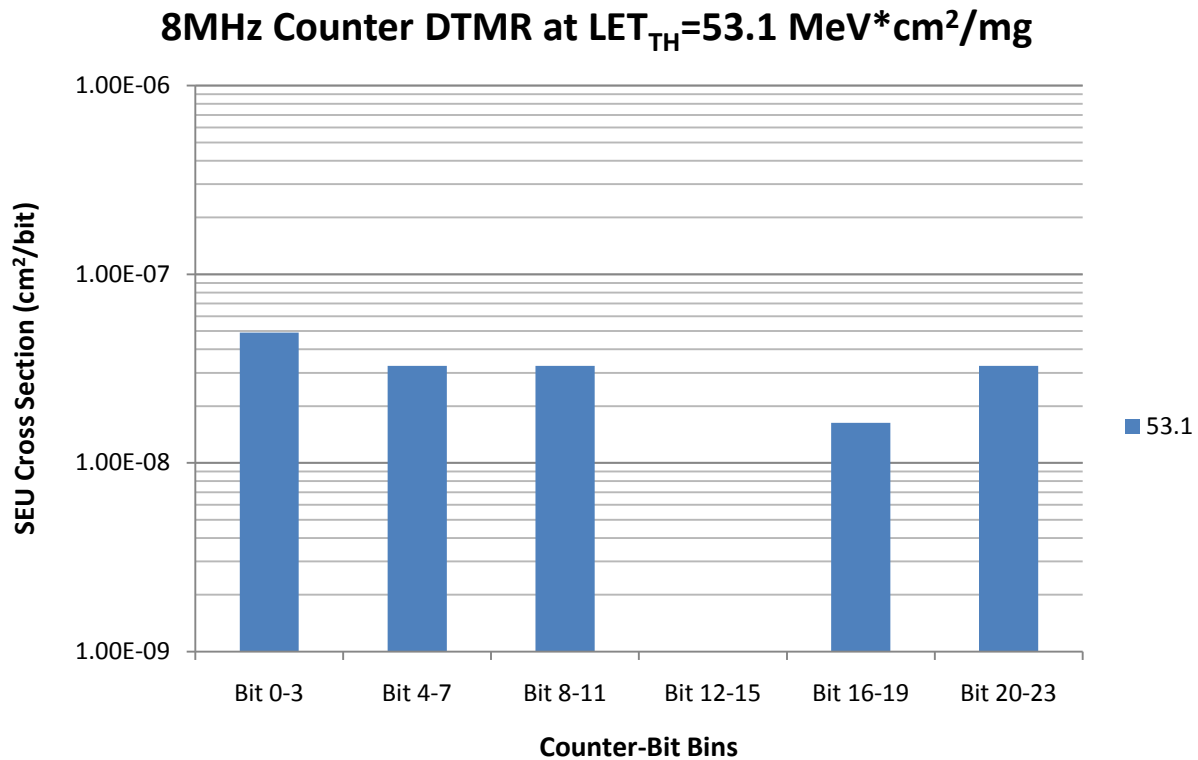


Figure 59: 8MHz Counter-bit DTMR Cross Sections. LET_{th} for the Counters was at 53MeV*cm²/mg. There is no enable pin connected to the counters. Hence upsets are local resets or clocks

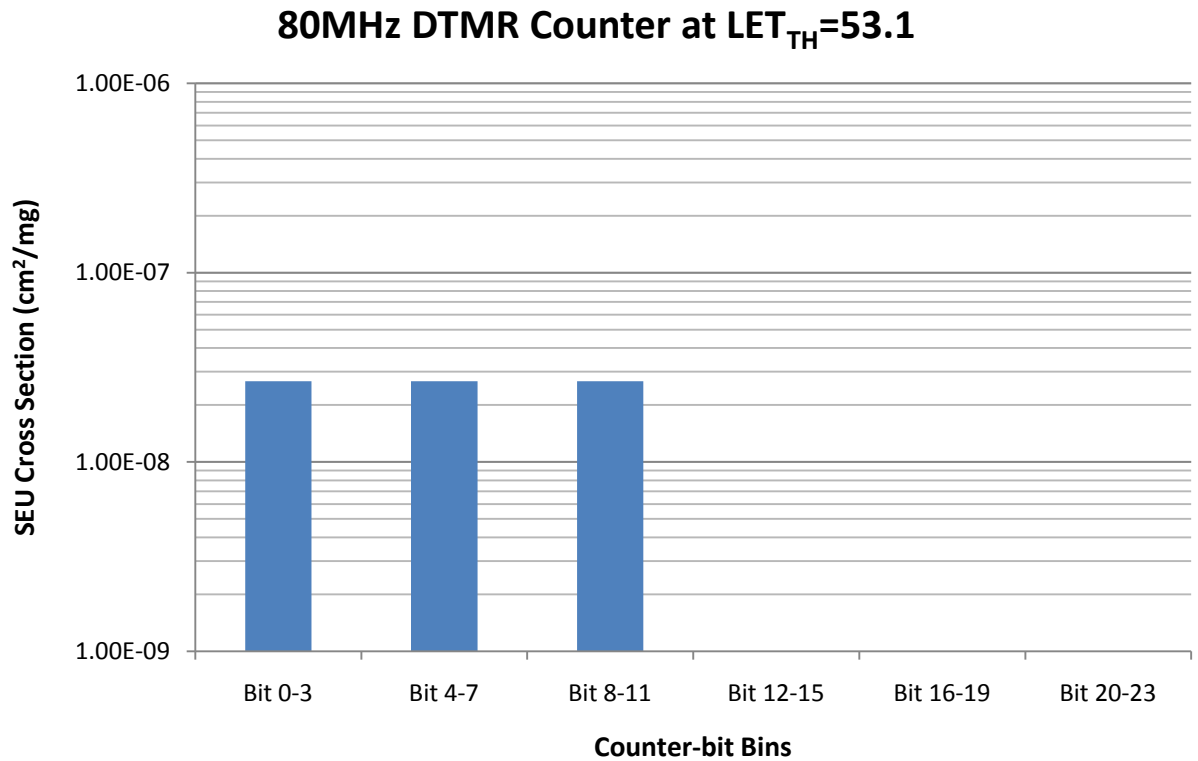


Figure 60:80MHz Counter-bit DTMR Cross Sections. LET_{th} for the Counters was at 53MeV*cm²/mg. There is no enable pin connected to the counters. Hence upsets are local resets or clocks

9.5.2 Bursts

As previously mentioned, if the outputs are erroneous for consecutive clock cycles, then the event is reported as a burst. With the DUT designs under investigation in this study, bursts are due to global route circuitry. The DTMR SEU Cross sections reported in Section 9.5.1 pertain to localized (to a DFF) single cycle upsets and are normalized per bit. Bursts are not localized errors – i.e. many DFFs become upset at once. As an example, if a reset glitches due to a transient, the portion of the circuitry affected by the reset will go to a 0-state. This will cause the outputs to be in erroneous states for several clock cycles – i.e. the outputs will be in error until the circuit can return to the expected state.

Most importantly, when analyzing the SEU Cross Sections pertaining to bursts, recognize that the cross sections are higher because they are per device and are not normalized per-bit as all of the other cross sections in this document.

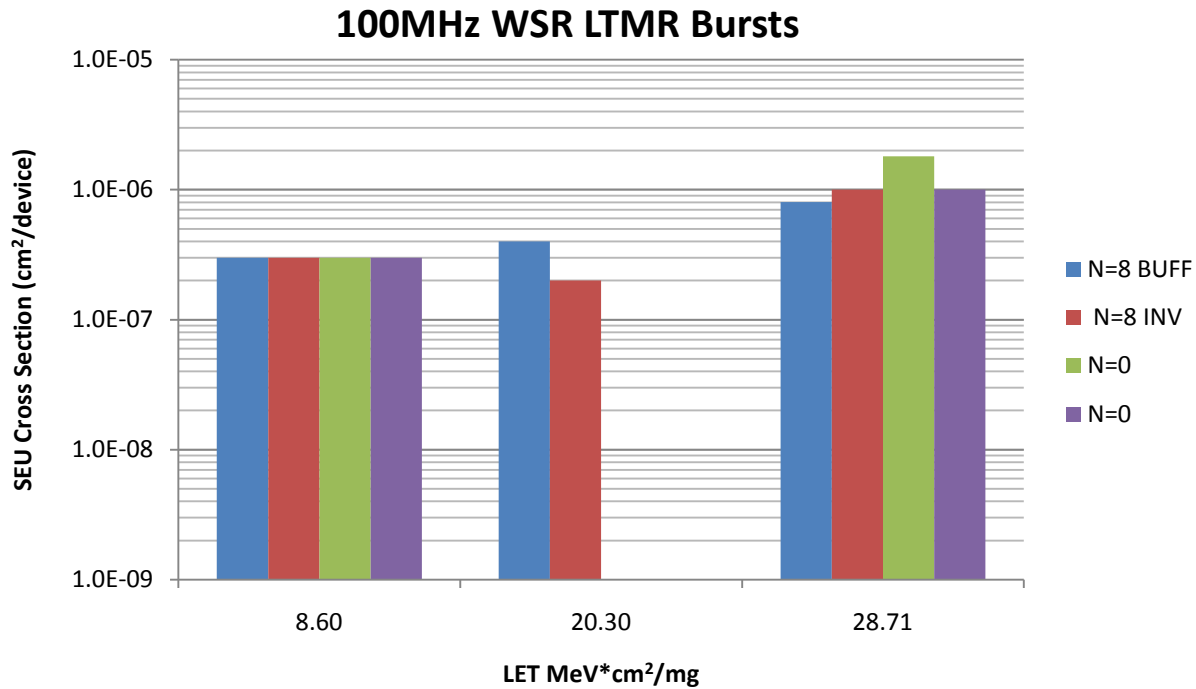


Figure 61: 100MHz WSR LTMR Bursts per Chain. Differences in Burst Cross Sections are insignificant

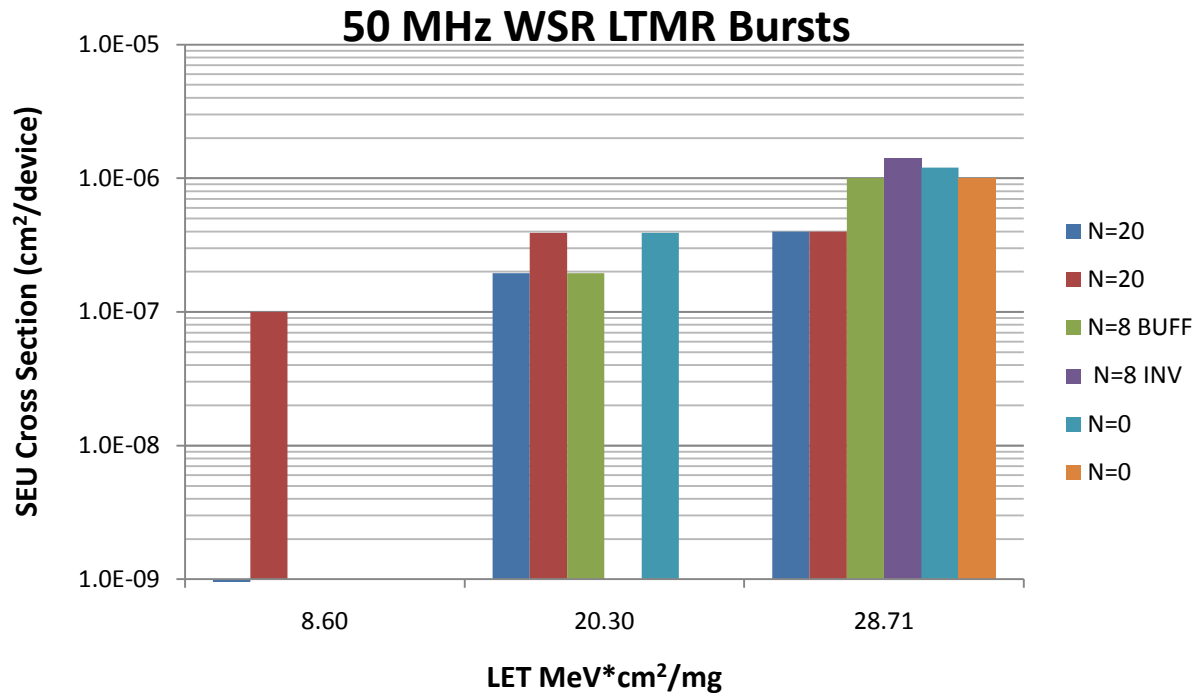


Figure 62: 50MHz WSR LTMR Bursts per Chain. Differences in Burst Cross Sections are insignificant

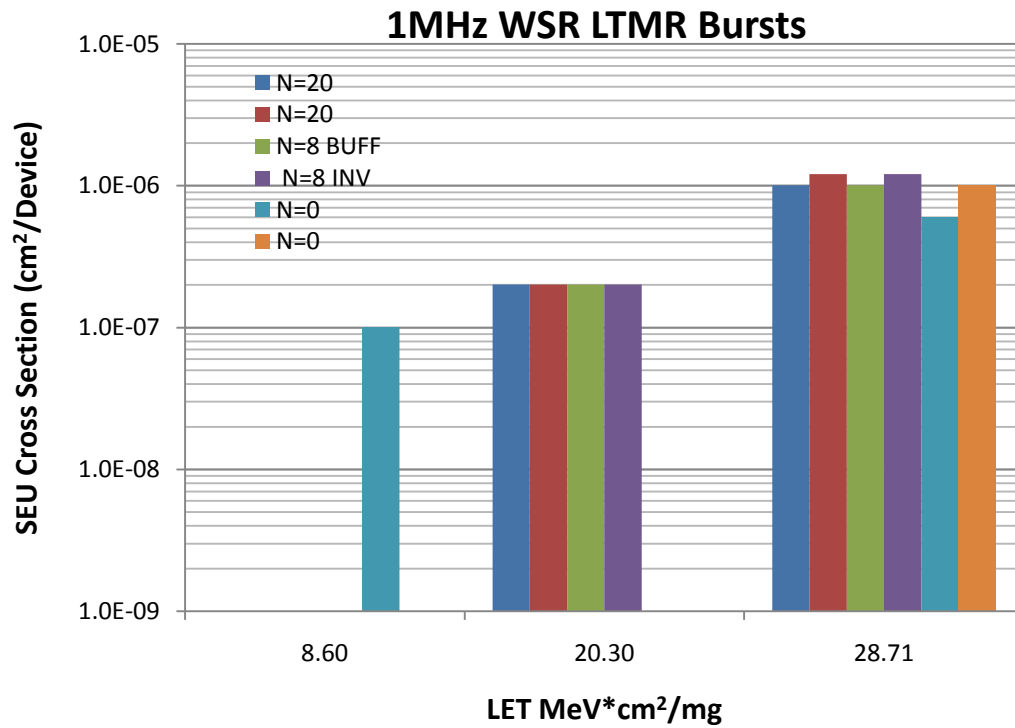


Figure 63: 1MHz WSR LTMR Bursts per Chain. Differences in Burst Cross Sections are insignificant

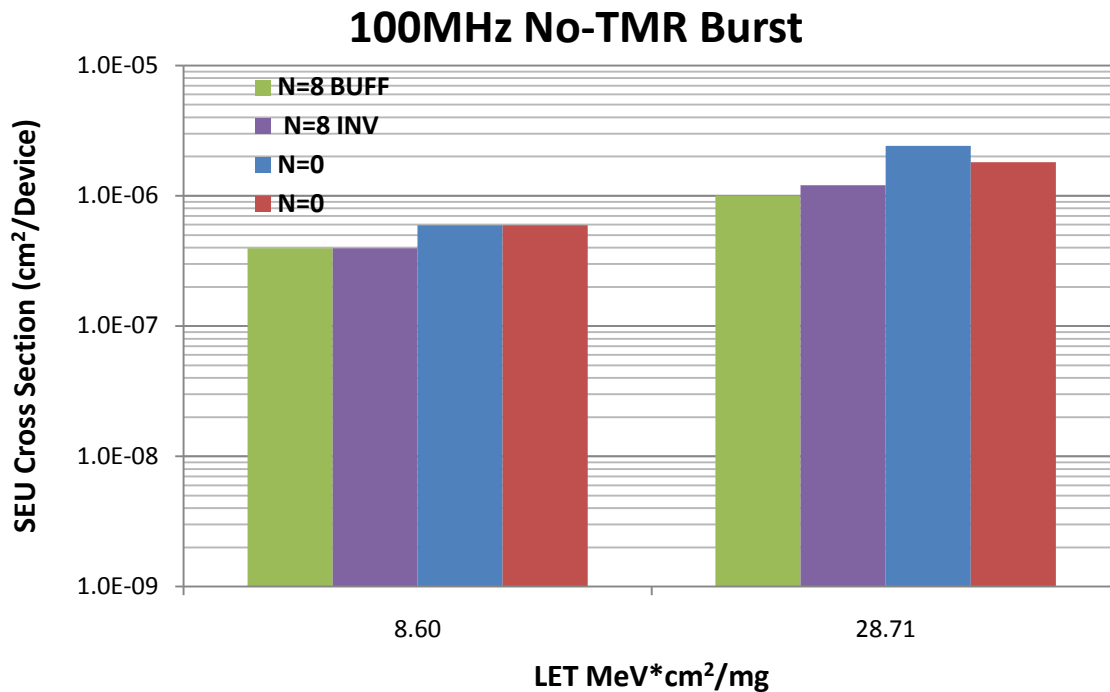


Figure 64: 100MHz WSR No-TMR Bursts per Chain. Differences in Burst Cross Sections are insignificant per LET

What is important to note regarding Figure 61 through Figure 64, is that the LET_{TH} of bursts was observed at 8.6 MeV*cm²/mg. Hence, circuits can reach a reset state or incur clock glitches at fairly low LET values.

9.6 Hamming Code 3 Finite State Machine (H3FSM)

H3FSM array is architected similar to the counter array. The difference is that the H3FSM has its unique mitigation within the state machine's encoding scheme. States are selected with a Hamming code-3 distance. EDAC circuitry is then inserted within the next state logic such that if an upset occurs, the next state logic can fix the upset.

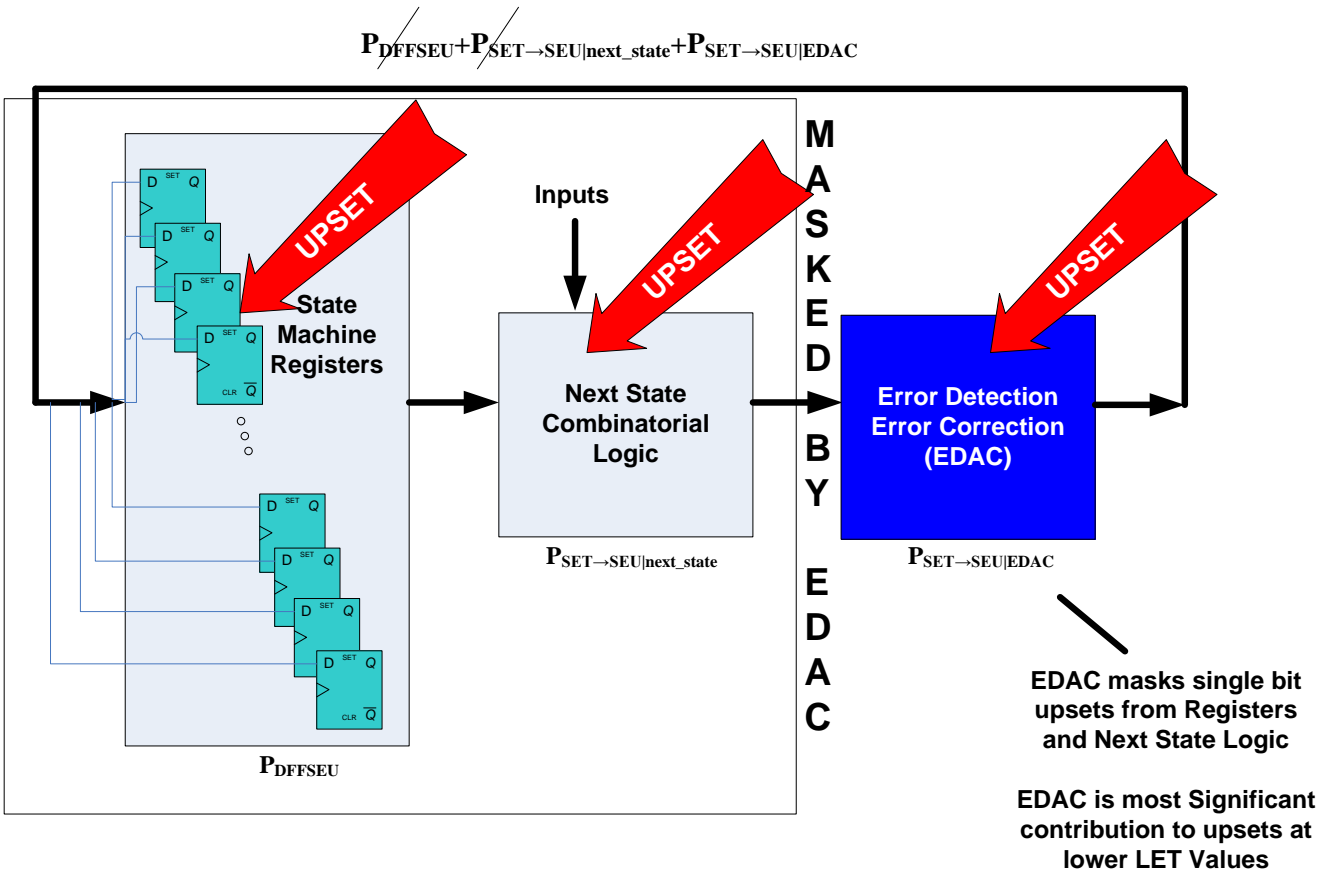


Figure 65: H3FSM potential upsets and EDAC masking of upsets. EDAC Combinatorial logic becomes the most significant contributor to SEU cross sections at lower LET values

Problems that can arise from such a method are:

If an upset occurs in a DFF mid-clock cycle, then the delay through next state logic and EDAC logic can cause the unsettled logic to be captured as incorrect next state

If an upset occurs in the EDAC or next state circuitry and fans-out to multiple gates, the incorrect next state can be calculated.

It is important to note that the H3FSMs contain 9-bits each. 5-bits represent the current state, while the additional 4-bits represent redundant logic. Upsets in the 4-bit redundant logic are not reported as upsets. However, upsets in the 5-bit state are considered as an upset because these are the bits that the design uses as control.

Figure 66 is a SEU cross section comparison between single bit upsets and multiple bit upsets for the H3FSM. As expected, single bit upsets have a significantly larger SEU cross section than multiple bit. In addition, a decade difference in frequency produces close to a decade difference in SEU cross section as illustrated in Figure 66.

H3FSM SEU Cross Sections: Single bit upsets and Multiple bit upsets at 80MHz and 8MHz

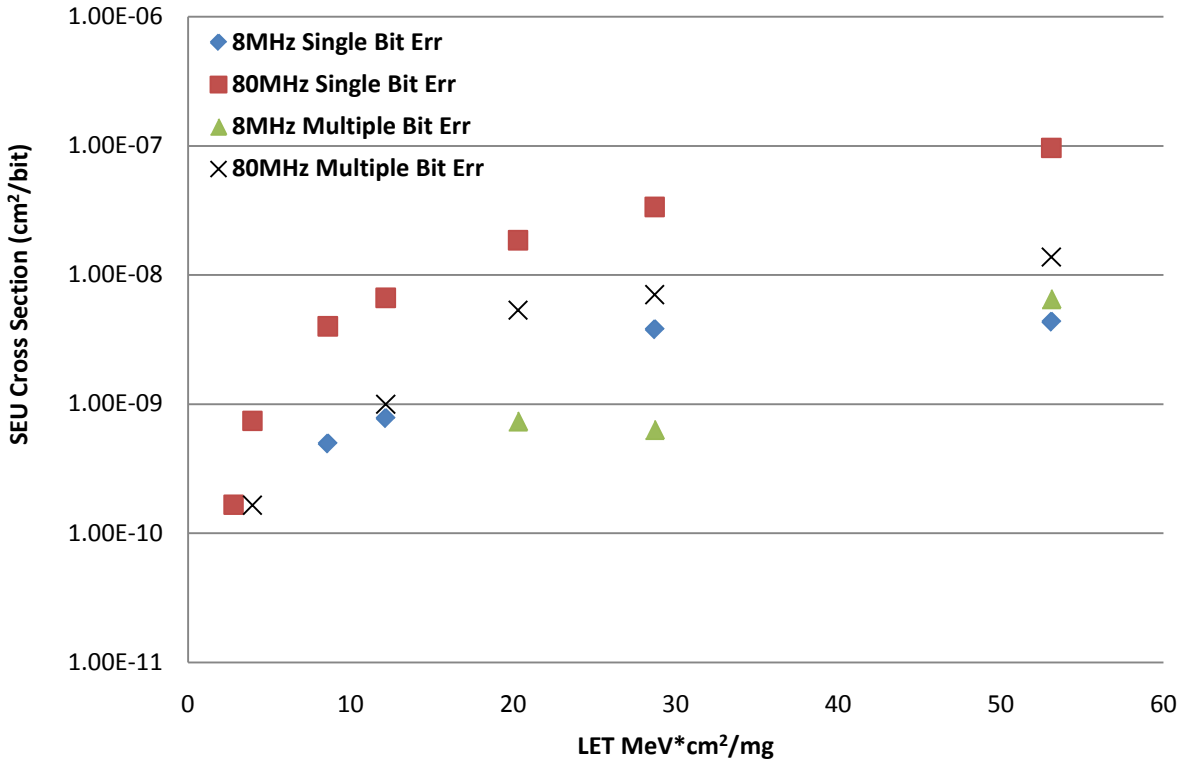


Figure 66: H3FSM Cross Sections for Single bit upsets and multiple bit upsets

9.6.1.1 LET_{TH} of H3FSM versus other Architectures and Mitigation Strategies

It is shown in Figure 67 that at 80MHz H3FSM does not reduce the LET_{TH} compared to No-TMR (or no mitigation). LTMR proves to be a more effective mitigation strategy than H3FSM. As previously stated, H3FSM relies on the premise that $P_{DFFSEU} \gg P_{SET \rightarrow SEU}$. Subsequently, the SEU data disproves (at high frequencies) the premise that $P_{DFFSEU} \gg P_{SET \rightarrow SEU}$ for the H3FSM architecture. This is as expected at high frequency because with H3FSM, the only mitigation is the EDAC block of circuitry. However, the EDAC circuitry is unprotected. Unprotected combinatorial logic causes can incur SETs that can potentially be captured by destination DFFs.

Figure 68 illustrates the SEU cross sections due to single bit upsets versus multiple bit upsets within the current state of the H3FSM.

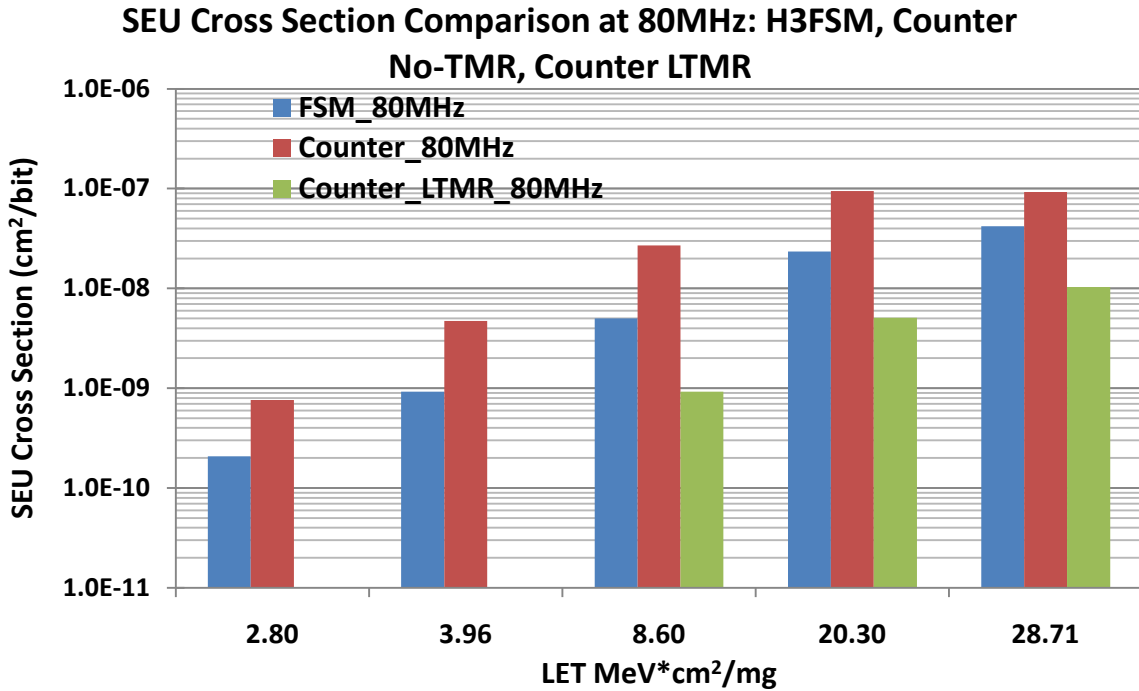


Figure 67: 80MHz Comparison between H3FSM, Counter No-TMR, and Counter LTMR. It is shown that LTMR provides a lower overall SEU cross section and reduces LET_{TH}. H3FSM slightly reduces the SEU cross sections versus No-TMR Counters. However, LET_{TH} does not change.

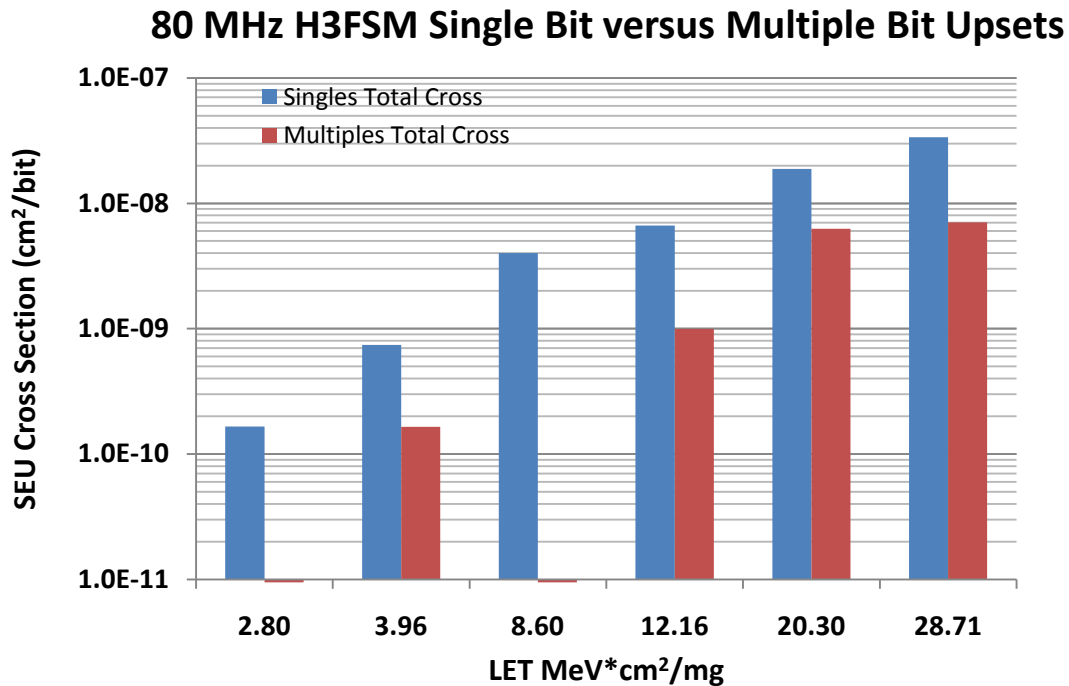


Figure 68: Comparison of 80MHz H3FSM Single bit upset SEU Cross sections vs. Multiple Bit SEU Cross sections. Multiple bit upsets are mostly non-global route related. All cross sections are single cycle. They are generated due to incorrect next-state calculation.

9.6.1.2 H3FSM Frequency Effects

Figure 69 shows that SEU frequency effects exist with H3FSM. There is a significant difference between 80MHz versus 8 MHz SEU Cross Sections. This suggests that $P_{SET \rightarrow SEU}$ significantly contributes to cross sections. However, unlike LTMR, $P_{DFE \rightarrow SEU}$ is not significantly decreased due to the lower LET_{TH} at both 80MHz.

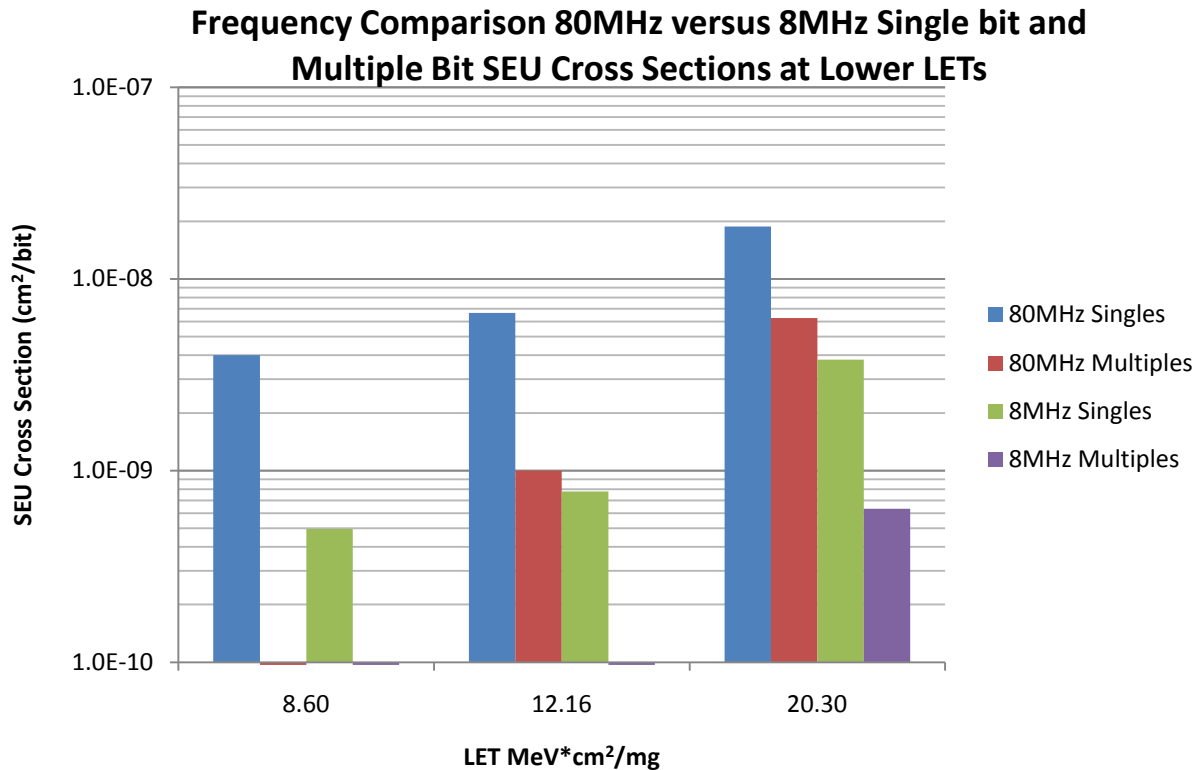


Figure 69: Comparison of 80MHz and 8MHz H3FSM Single bit upset SEU Cross sections vs. Multiple Bit SEU Cross sections. Frequency effects exist. No test runs were performed for 8MHz with $LET < 8.6 \text{ MeV*cm}^2/\text{mg}$

9.6.1.3 H3FSM SEU Cross Sections for each of the 5-Bits

As illustrated in Figure 70 and Figure 71, SEU cross sections do not differ significantly from bit to bit across the state machine. This is because the EDAC logic is shared across bits and hence equally distributes upset probability.

80MHz H3FSM SEU Cross Sections per Bit at Lower LETs

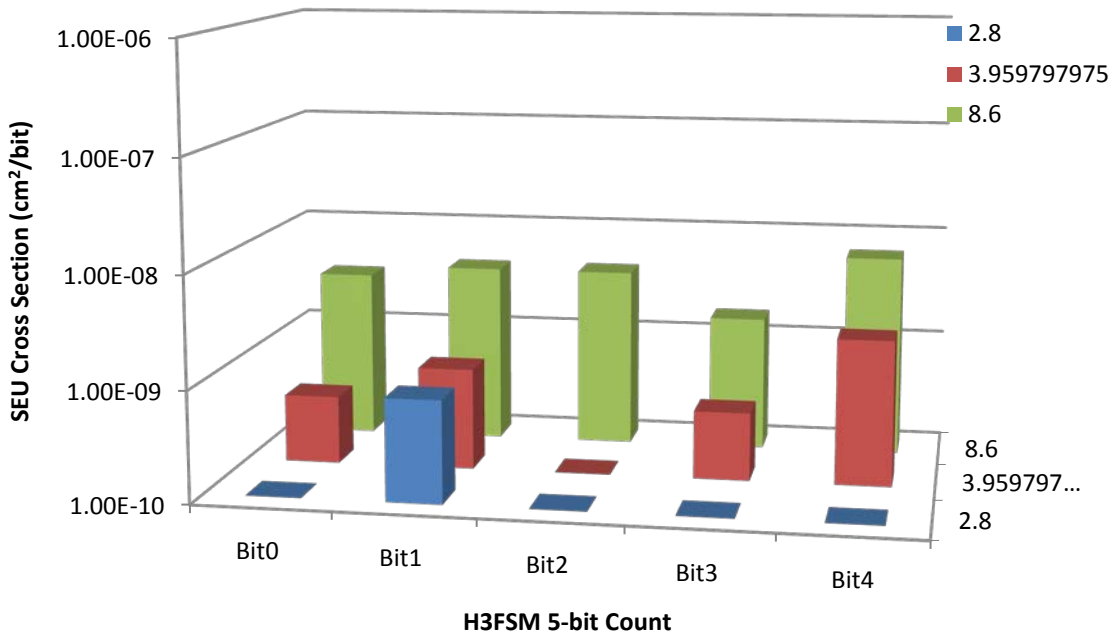


Figure 70: 80MHz H3FSM SEU Cross Sections at Lower LET values across the 5 state bits.

80MHz H3FSM SEU Cross Sections per Bit at Higher LETs

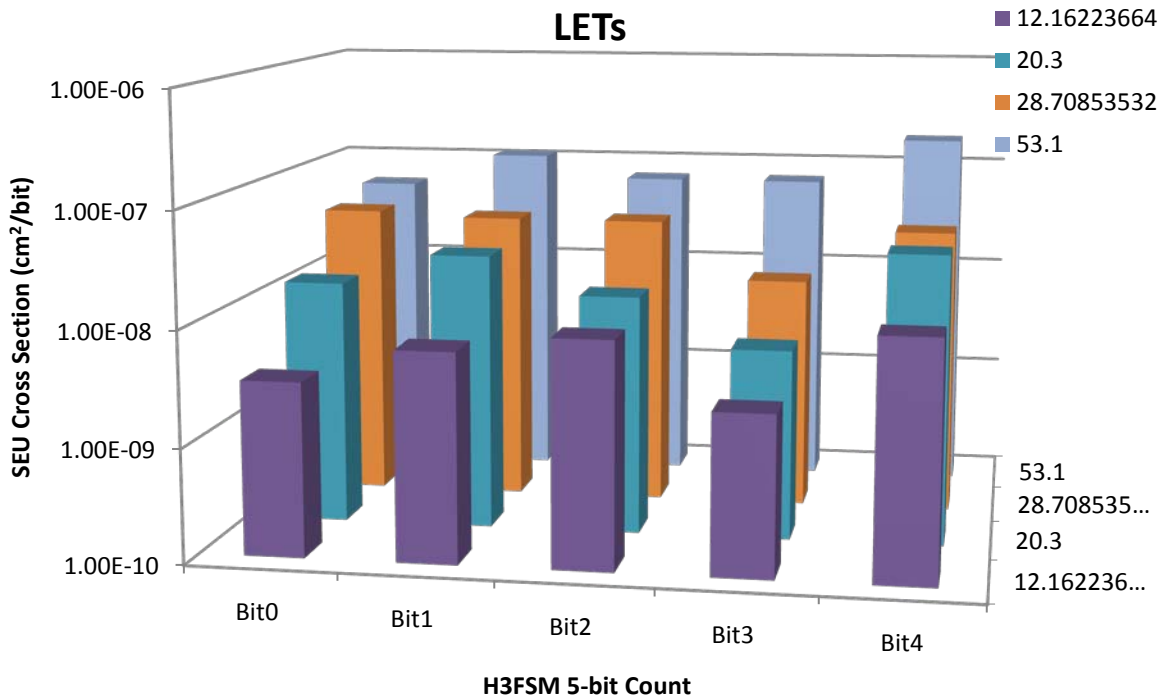


Figure 71: 80MHz H3FSM SEU Cross Sections at Higher LET values across the 5 state bits.

10. CONCLUSIONS

The primary goal of calculating SEU cross sections is to predict design/device upset rates for critical flight projects. Subsequently, a comparison between SEU cross sections across design state space assists in interpolating or extrapolating calculated SEU data to actual designs. This study incorporated placing various designs and mitigation strategies within the ProASIC device for heavy-ion radiation tests. WSRs are compared to Counters to determine how complexity affects SEU Cross sections. The mitigation techniques were used in order to investigate how effective they reduce the SEU Cross Sections. All mitigation was inserting using [Mentor Graphics Precision Rad-Tolerant](#) [4].

10.1 Reprogramming Issues

Some devices have not been able to be reprogrammed after being exposed to heavy ions with LET $>53\text{MeV}\cdot\text{cm}^2/\text{mg}$. If the device is expected to be re-programmed within a mission, then this can be considered as a destructive event. The LET_{th} of this event has not been determined at this time. More testing is required to find LET_{th} . It is important to note that if the device does not require reprogramming, no destructive behavior was observed at high LET values prior to expected dose limitations (10Mrad). No reprogramming issues were observed for LET values $<28\text{MeV}\cdot\text{cm}^2/\text{mg}$.

10.2 WSR and Counters

With No-TMR, there was not a significant difference between WSR and counter architectures. This is because of the dominance of:

P_{SEUDFF} at low LET values. Upsets are not frequency dependent. Upsets are due to transients within the DFF feedback loop (not captured transients from the data path). $P_{\text{SET}\rightarrow\text{SEU}}$ exists but is minimal compared to P_{SEUDFF} .

P_{SEFI} at high LET values. Upsets are not frequency dependent. Upsets are from global routes (e.g. Clock, Reset, and global Enables). Hence, the usage of global routes will drive the upset rate of the design.

LTMR circuits pertaining to WSR and Counters produced SEU cross sections with $P_{\text{SET}\rightarrow\text{SEU}}$ dominance at lower LET values. This is because LTMR masks P_{SEUDFF} . This suggests that data pattern, combinatorial blocks, and clock frequency will affect upset rates. It has been shown that as the number of combinatorial logic blocks increase between DFFs, so does the SEU cross section. Cross section also increases with frequency and data pattern rate. P_{SEFI} begins to take affect and have more of a contribution at higher LET values.

Single cycle (or localized) DTMR circuits pertaining to WSR and Counters are dominated by P_{SEFI} . Data pattern can have an effect, depending on reset conditions, and enables.

	No-TMR	LTMR	DTMR
LET_{TH}	$2.8\text{MeV}\cdot\text{cm}^2/\text{mg}$	$8.6\text{MeV}\cdot\text{cm}^2/\text{mg}$	$20\text{MeV}\cdot\text{cm}^2/\text{mg}$
Frequency Dependency	No	Yes	No
Data pattern Dependency	No	Yes	(1) Depends on reset and enable conditions. Static patterns opposite reset condition has a significantly higher cross section (2) Static patterns are clock friendly
$P(\text{fs})$ @ Low LET	P_{SEUDFF}	$P_{\text{SET}\rightarrow\text{SEU}}$	P_{SEFI}
$P(\text{fs})$ @ High LET	$P_{\text{SEUDFF}} + P_{\text{SEFI}}$	P_{SEFI}	P_{SEFI}

10.3 H3FSM

Although the H3FSM slightly reduced the overall SEU cross section versus No-TMR, it did not affect LET_{TH} . $LET_{TH} = 2.8\text{MeVcm}^2/\text{mg}$ for No-TMR and H3FSM. The reason is because the circuitry used to mask and correct $P_{SEU\rightarrow DFF}$ and $P_{SET\rightarrow SEU}$ was large enough to have its own significant $P_{SET\rightarrow SEU}$. Due to the complexity of the H3FSM and lack of impact to SEU reduction, the H3FSM as currently implemented [4] is not a recommended means of mitigation for the ProASIC device.

10.4 Bursts

If the DUT outputs are erroneous for consecutive clock cycles, then the event is reported as a burst. With the DUT designs under investigation in this study, bursts are due to global route circuitry. The DTMR SEU Cross sections reported in Section 9.5.1 pertain to localized (to a DFF) single cycle upsets and are normalized per bit. Bursts are not localized errors – i.e. many DFFs become upset at once. As an example, if a reset glitches due to a transient, the portion of the circuitry affected by the reset will go to a 0-state. This will cause the outputs to be in erroneous states for several clock cycles – i.e. the outputs will be in error until the circuit can return to the expected state.

The LET_{TH} of bursts in ProASIC devices was detected at $8.6\text{MeV}\cdot\text{cm}^2/\text{mg}$.

11. APPENDIX 1:

- [1] Actel Datasheet: "PROASIC/SL RadTolerant FPGAs" http://www.actel.com/documents/PROASIC_DS.pdf, V5.2, October 2007.
- [2] M. Berg "An Analysis of Single Event Upset Dependencies on High Frequency and Architectural Implementations within Actel PROASIC Family Field Programmable Gate Arrays," IEEE Trans. Nucl. Sci., vol. 53, n° 6, Dec. 2006.
- [3] M. Berg "Trading Application Specific Integrated Circuit (ASIC) and Field Programmable Gate Array (FPGA) Considerations for System Insertion", NSREC Short Course, Quebec City, CN, July 2009
- [4] Mentor Graphics Precision Documentation:
https://supportnet.mentor.com/docs/201009057/docs/pdfdocs/precisionRTL_users.pdf
https://supportnet.mentor.com/docs/201009057/docs/pdfdocs/precisionRTL_users.pdf (see chapter 7)