Xilinx Virtex 4 LX25 Preliminary Test Results

Melanie Berg – Principle Investigator MEI
Hak Kim, Christian Poivey, Mark Friendlich, Anthony Phan: MEI
Ken LaBel: NASA/GSFC

Test Date: 3/2006(IUCF); 6/2006(TAMU); 8/2006(TAMU); 7/2006(IUCF); 8/2007(TAMU); 10/2007(IUCF)

Report Date: 1/2008

1. INTRODUCTION

This study was undertaken to determine the single event destructive and transient susceptibility of the Xilinx Virtex 4 – LX25. The tests reported in this document are preliminary steps within the full study of the Virtex 4- family. Key objective of the research was to perform a preliminary investigation of sensitivity levels for the Xilinx V4 LX25 SRAM-based FPGA device (with and without scrubbing). The devices were monitored for Single Event Transient (SET) and Single Event Upset (SEU) induced faults by exposing them to a heavy ion beam at the Texas A&M University Cyclotron Single Event Effects Test Facility. SEU testing was also performed with a proton beam at the Indiana University Cyclotron Facility (IUCF).

2. DEVICES TESTED

There were several shift register designs with 2 separate scrubbing schemes tested within separate LX25 parts (internal scrubbing and external scrubbing). The devices were manufactured on an advanced 0.9nm copper CMOS Process Technology. The manufacturer is Xilinx.

2.1 DUT Architecture

All DUTS consisted of windowed shift register strings see Figure 1. We started with a length of 10,000 and observed that the duration of time of inoperability was too large upon configuration hits (time to scrub the bit + 10,000 cycles of recovery). Later tests implemented multiple shift register strings within a DUT each containing 300 flip-flops with varying layers of combinatorial logic (see Figure 2 for an illustration of DFFs and combinatorial logic within a string).

The premise of using a windowed shift register is to reliably perform high speed testing. In order to calculate accurate cross sections, it is mandatory that the tester have observability at every shift clock cycle. Implementation is as follows: All bits are shifted at every clock cycle. The last 4 DFFs are copied into a window at every clock period. A windowed shift register divides the system clock by-4 clock to shift the last 4 bits of the Shift register string into a DFF window (SCAN_DATA). The window is output to the tester. A data clock (SHIFT_CLK) is also output to the tester for high speed synchronous data capture. See Figure 1 for more information.

DUT Top Level Architecture

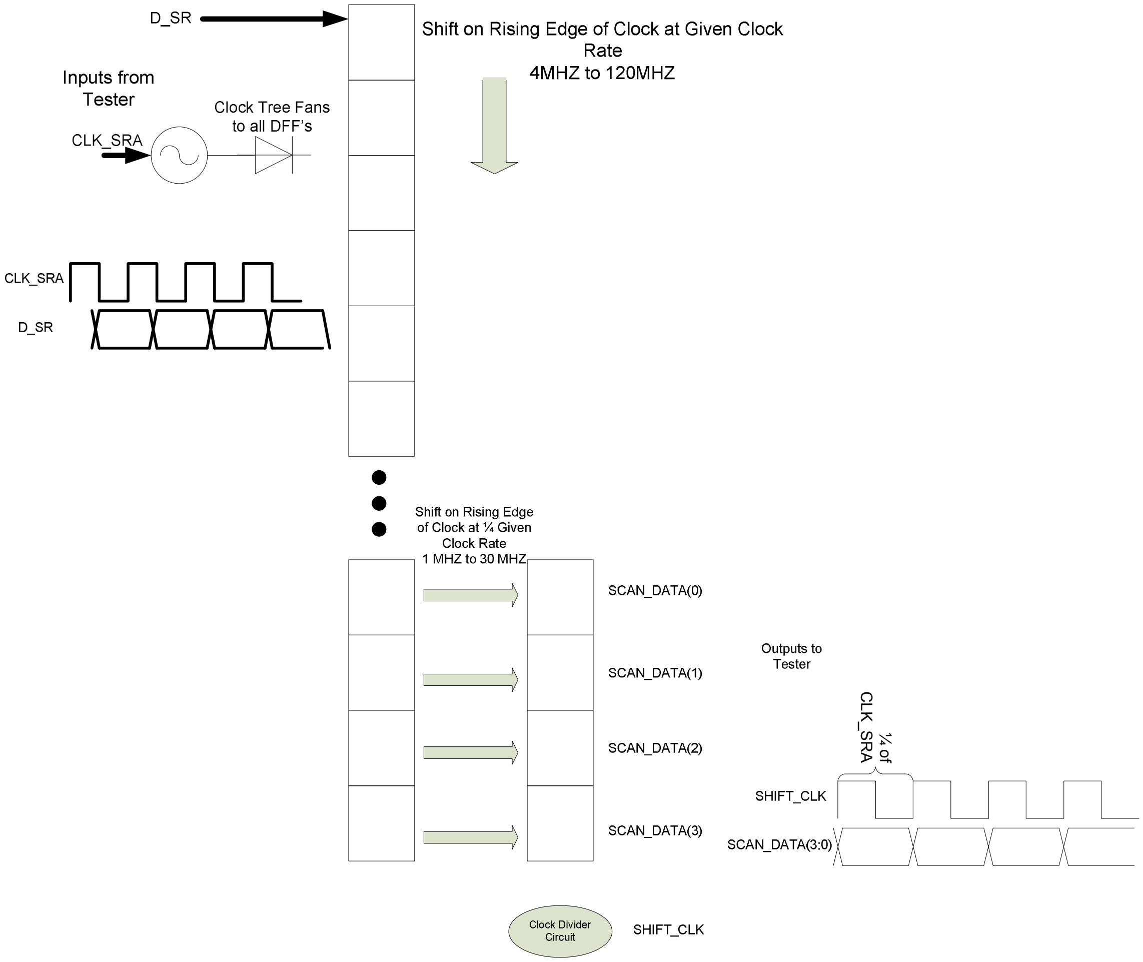


Figure 1: Device Under Test Top Level Architecture

2.1.1 DUTs that Contained 10,000 DFFs per DUT

DUT was configured to contain one windowed shift register string.

2.1.2 DUTs Containing 300 DFFs

There were 3 types of shift registers each containing 300 flip flops within its string (2 copies of each were implemented in the DUT – 6 total strings of shift registers). The types are differentiated by the number of inverters between the flip-flops (see Figure 2) as follows:

1. Only Flip-flops within the string (no combinatorial logic N= 0).
2. Flip-flops with 8 inverters between each flip-flop stage (N= 8).
3. Flip-flops with 20 inverters between each flip-flop stage (N= 20).

Because the inverters were mapped into Look-Up-Tables (LUTs: configuration memory plus combinatorial logic implementing a function generator), the shift register implementations that contain inverters utilized both flip-flops and a large number of LUT resources (see Table 1 and Table 2).

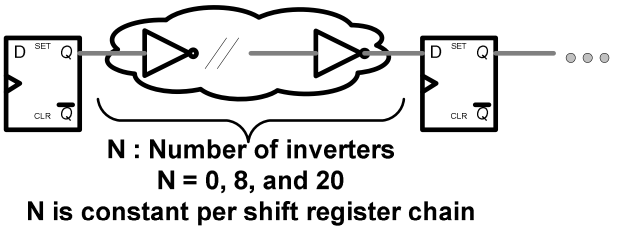


Figure 2: One Stage of the DUT Shift Register Chain

2.1.3 Shift Register String Lengths

Test dates 3/2006 to 8/2006 implemented shift register strings of 10,000. Test dates in 08/2007 used shift register strings of 300.

2.1.4 DUT Internal Reset Circuitry

Reset passes through an asynchronous assert – synchronous de-assert circuit and is supplied to every DFF. The following is the reset circuit used within the DUT.

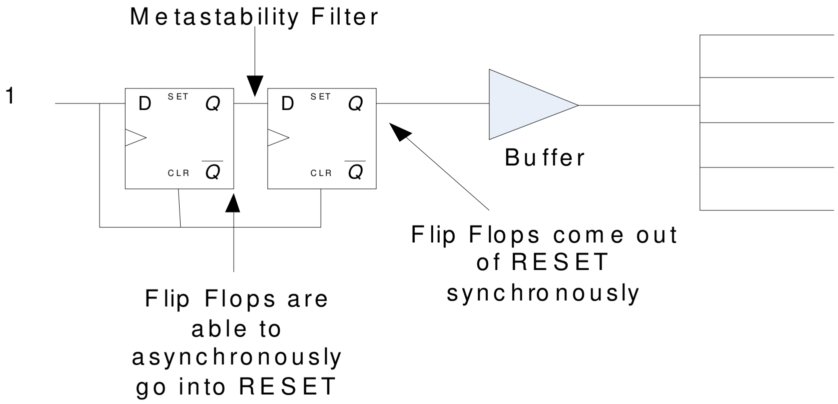

Metastability Filter
1
D
SET
Q
CLR
Q
D
SET
Q
CLR
Q
Buffer
Flip Flops come out of RESET synchronously
Flip Flops are able to asynchronously go into RESET

Figure 3: Asynchronous Assert - Synchronous De-assert

2.1.5 Device Utilization: 300 Count Shift Registers without Internal Scrubbing and with Internal Scrubbing

Table 1 and Table 2 each contain a brief summary of resource utilization for the shift register implementation sans the Xilinx internal scrubber and containing the internal scrubber respectively. Inserting the internal scrubber increases the resource utilization between 1% and 2%. This increase is negligible and thus eases the insertion process for the designer during the logic mapping and place and route phases of implementation.

All shift registers were run at 100 MHz. All 6 of the shift register strings were contained in one FPGA device. The 6 shift registers were implemented as one design without the Xilinx IP Core internal scrubber and as another design containing the Xilinx IP core internal scrubber.

Table 1: Device Utilization Summary - 300 length shift register chain

Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	1,798	21,504	8%
Number of 4 input LUTs	16,578	21,504	77%
Logic Distribution			
Number of occupied Slices	9,638	10,752	89%
Number of Slices containing only related logic	9,638	9,638	100%
Number of Slices containing unrelated logic	0	9,638	0%
Total Number of 4 input LUTs	16,578	21,504	77%
Number of bonded IOBs	33	448	7%
Number of BUFG/BUFGCTRLs	1	32	3%
Number used as BUFGs	1		
Number used as BUFGCTRLs	0		
Total equivalent gate count for design	114,100		
Additional JTAG gate count for IOBs	1,584		

Table 2: Device Utilization Summary - 300 length shift register chain w/ ICAP

Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	1,983	21,504	9%
Number of 4 input LUTs	16,784	21,504	78%
Logic Distribution			
Number of occupied Slices	9,947	10,752	92%
Number of Slices containing only related logic	9,947	9,947	100%
Number of Slices containing unrelated logic	0	9,947	0%
Total Number 4 input LUTs	16,854	21,504	78%
Number used as logic	16,784		
Number used as a route-thru	2		
Number used for Dual Port RAMs	16		
Number used for 32x1 RAMs	52		
Number of bonded IOBs	37	448	8%
Number of BUFG/BUFGCTRLs	1	32	3%
Number used as BUFGs	1		
Number used as BUFGCTRLs	0		
Number of FIFO16/RAMB16s	2	72	2%
Number used as FIFO16s	0		
Number used as RAMB16s	2		
Number of ICAP_VIRTEX4s	1	2	50%
Number of FRAME_ECC_VIRTEX4s	1	1	100%
Total equivalent gate count for design	255,719		
Additional JTAG gate count for IOBs	1,776		

3. TEST FACILITY

3.1 Heavy Ion.

Facility: Texas A&M University Cyclotron Single Event Effects Test Facility, 25 MeV/amu tune).
Flux: 1.0E^{02} particles/(cm^2*s)
Fluence: All tests were run to 1 x 10^5 p/cm^2 or until destructive or functional events occurred.

Table 3: LET Table

Ion	Energy (MEV/Nucleon)	LET (MEV/cm2/mg) 0 deg	LET (MEV/cm2/mg) 45 deg
Ar	25	5.7	8.5
Cu	25	20.7	
Kr	25	28.5	40.26
Xe	25	52.7	74.5

Data were taken at Ar and Kr. No data available as of yet for Cu and Xe.

3.2 Proton

Facility: Indiana University Cyclotron Facility (IUCF)
Flux: 3.0E^9 particles/cm^2*s
Fluence: All tests were run to 1.0E^{12} p/cm^2 or until unrecoverable errors were observed
Energy : 89 and 198 MeV

4. TEST CONDITIONS

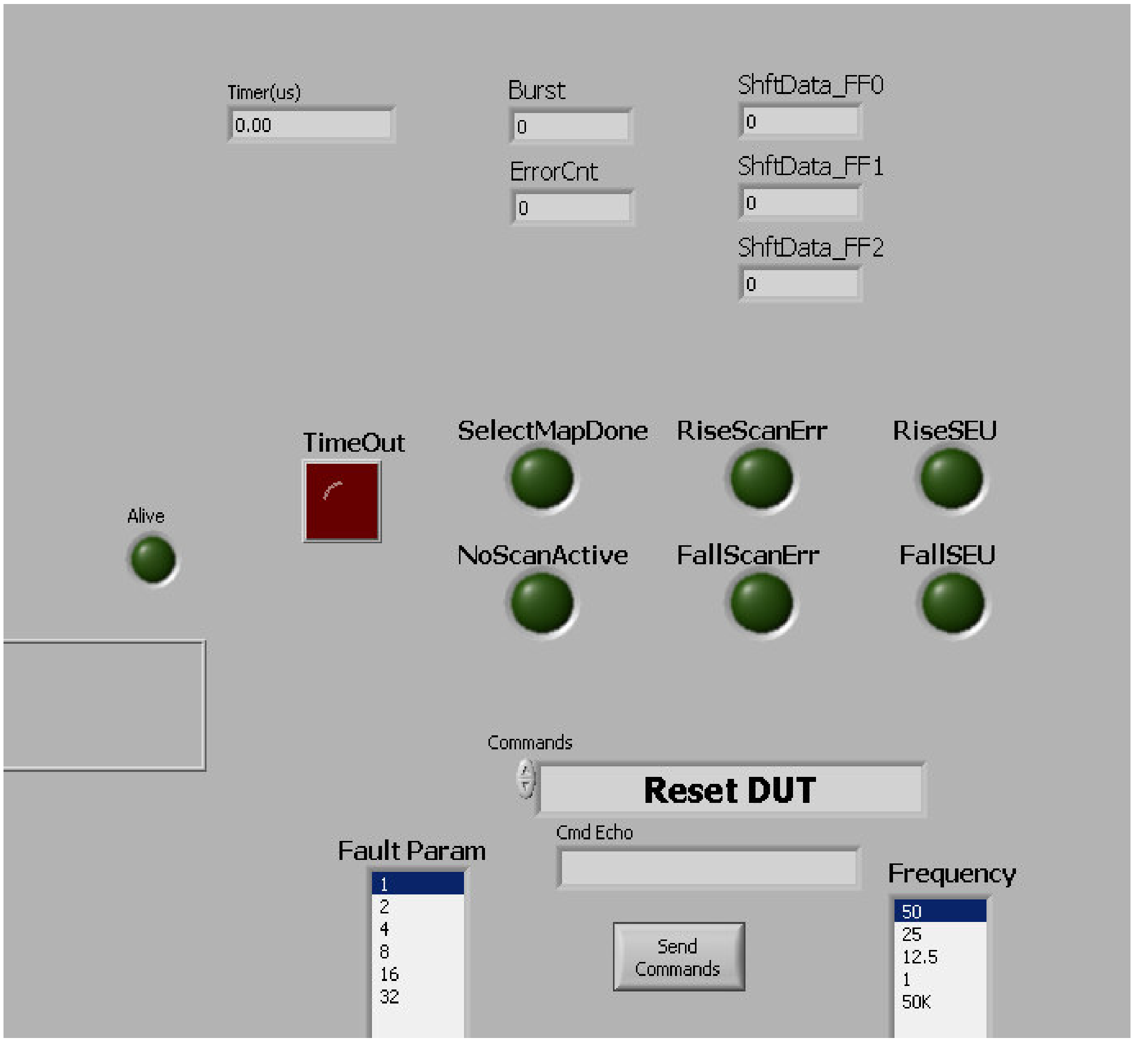
Test Temperature: Room Temperature
Operating Frequency: 100MHz
Power Supply Voltage: 3.3v I/O; 2.5V Auxiliary; 1.8V PROM; 1.2V Core

5. TEST METHODS

5.1 Architectural Overview

The Low Cost Digital Tester (LCDT) consists of a Mother Board (FPGA Based Controller/Processor) and a daughter board (containing DUT and its associated circuitry). The DUT is socketed onto the daughter card. The objective of the DUT Controller/processor (mother board) is to supply inputs to the Virtex-4 LX25 Device (DUT – daughter card) and perform data processing on the outputs of the Virtex-4 LX25. The LCDT communicates with a user controlled PC. The user interface is LAB-VIEW (see Figure 4). The user will send specified commands to the motherboard and receive information from the motherboard via the LabView GUI. Burst and ErrorCnt windows are available for the user to determine the state of the DUT (temporary error state vs. unrecoverable error state). ShftData_FFn is a window that contains

DUT output error information. One ShftData_FF0 represents all 6 shift_registers during non-TMR circuit testing. There are 3 ShftData_FFn windows for future TMR testing. The Alive button indicates that the tester (not the DUT) is still alive. The Alive button flashes as the test is running. Please see Documents: "LCDT" and "General Tester" for further information concerning the LCDT functionality.



Timer(us): 0.00

Burst: 0

ErrorCnt: 0

ShftData_FF0: 0

ShftData_FF1: 0

ShftData_FF2: 0

Commands: Reset DUT

Cmd Echo:

Fault Param: 1

Frequency: 50

Figure 4: LABView User Interface (Resides on Host PC)

The LCDT is connected to the Virtex-4 LX25 DUT as shown in the following Block Diagram.

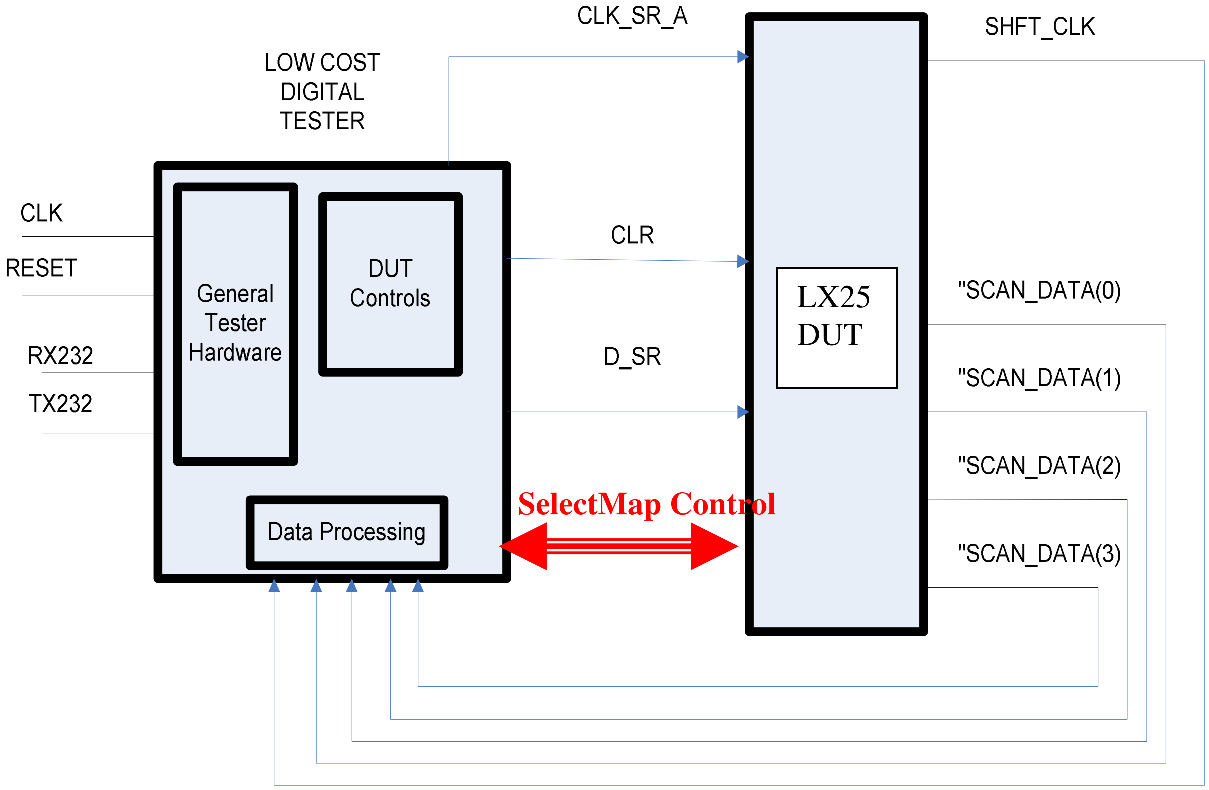


Figure 5: System Level Tester Architecture

5.1.1 Tester I/O List and Definitions

Table 4: I/O Table

Input Name	Description	Direction	Synchronous	Slew	Pullup
CLK	System clock of the LCDT	Input	Clock		N
RESET	LCDT system reset	Input	A		N
RX232	Serial receive input	Input	A		N
SCAN_DATA(4:0)	Data window of Virtex-4 LX25. Data is processed by LCDT and compared	Input	A		N

	against expected value				
SHIFT_CLK	Output clock of VIRTEX-4 LX25. Used to control SCAN_DATA capture. SHIFT_CLK is always ¼ the speed of CLK_SR_A. However it is not synchronous with CLK_SR_A	Input	A		N
CLK_SR_A	Input clock to VIRTEX-4 LX25. Max speed is 150mhz	Output		FAST	N
CLR	Reset to the Virtex-4 LX25	Output		FAST	N
D_SR	Data Input to the Virtex-4 LX25	Output			N
TX232	Serial transmission line	Output			N
Selectmap_CCLK	SelectMap 30MHz clock	Output			
Selectmap_CSN	SelectMap Data valid	Output			
Selectmap_DONE	SelectMap Done (configuration) signal	Output			
Selectmap_BUSY	SelectMap Busy	Output			

Selectmap_INIT	SelectMap INIT	Output			
Selectmap_PROG_B	SelectMap Reset Signal	Output			
Selectmap_RW_B	SelectMap Read/Write	Output			
Selectmap_DATA	SelectMap 8bit Data	Output			
SCRUB_RX232	Serial input to receive configuration data via RS232 port	Input	A		
SCRUB_Tx232	Not Used				
SRAM_D	SRAM Data 16bit	INOUT	A		
SRAM_A	SRAM Address = 20 bits	Output			
SRAM_WR	SRAM read/write	Output			
SRAM_OE	SRAM output enable	Output			
SRAM_CE	SRAM chip select	Output			
SRAM_BLEN	SRAM byte enable	Output			
SRAM_BLHN	SRAM byte enable	Output			
RUN_SCAN	Turn self scrubber on	Output			
SCAN_ACTIVE	Self scrubber	Input	A		
SCAN_ERROR	Self scrubber	Input	A		
SCAN_MODE	Self scrubber	Output			
ERROR_INJECT	Self scrubber	Output			

SEU_DETECT	Self scrubber	Output			
QUICKUSB_FCLK	50MHz clock from quickusb device	Input	CLK		
QUICKUSB_REN		Output			
QUICKUSB_WEN		Output			
QUICKUSB_DATA	8 bit data	INOUT	A		
QUICKUSB_CMD_DATA	Command differentiation	Output			
TP*n*	Test points	Output			

5.2 Requirements

5.2.1 Requirement Summary

There are 3 main investigations:

1. Test shift register logic structures
2. Test external scrubber
3. Test Xilinx internal scrubber

The requirements for the Virtex-4 LX25 LCDT tester are listed in Table 3.

Table 5 – Requirements Table

Item	Requirement
1	Supply System Clock to the Virtex-4 LX25 DUT
2	Supply Reset to Virtex-4 LX25
3	Supply Data Input to the Virtex-4 LX25
4	Clock Frequency of Virtex-4 LX25 shall be variable
5	Maximum Virtex-4 LX25 input clock frequency shall be 100Mhz
6	0,1, and checker board data patterns shall be generated and placed on the VIRTEX-4 LX25 data lines
7	VIRTEX-4 LX25 reset shall be active low

8	VIRTEX-4 LX25 reset shall be active for at least 3 VIRTEX-4 LX25 system clocks
9	VIRTEX-4 LX25 Data Inputs shall be stable at the Rising Edge of the VIRTEX-4 LX25 system clock with a set-up time of 3ns and a hold time of 3ns
10	VIRTEX-4 LX25 data inputs shall be captured by the LCDT data processing module once detecting the rising edge of the data clock (SHIFT_CLK)
11	SHIFT_CLK rising edge detection must include a metastability filter because the SHIFT_CLK input is asynchronous.
12	Input Data must be registered before the data processing block implements the compares – protects against radiation induced I/O transients.
13	Data processing block shall report every error to the FIFO block
14	Tester must supply external LX25 scrubber
15	External scrubber clock shall be separate than the system (shift register) clock
16	External scrubber clock shall be able to functional at 25MHz
17	Tester must be able to control an internal scrubber
18	Internal scrubber clock shall be operational at 100MHz
19	Internal scrubber clock shall be separate than the system (shift register) clock
20	External scrubber and Internal scrubber shall not operate simultaneously (during the same test)

The tester supplies inputs as follows: Data (D_SR) changes at the falling edge of the input clock (CLK_SR) so that it is stable and can be captured at the rising edge. CLK_SR and D_SR will be at the user specified frequency. The user will also supply (by command) the preferred data pattern. Data patterns range from all 0's , all 1's, and alternating 1's and 0's.

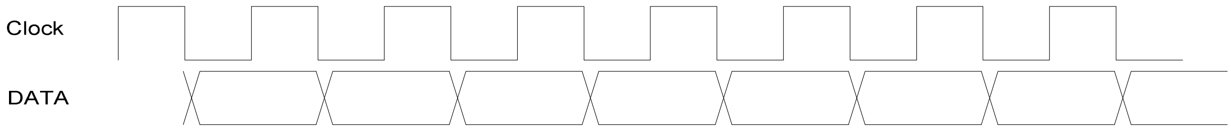


Figure 6: Tester Output to DUT - Clock and Data Relation

5.2.2 External Scrubbing

5.2.2.1 Brief Overview

External scrubber is contained in the tester. It will communicate with the DUT via SelectMap I/O (8 bit). External scrubber is implemented very similar to the DUT configuration manager. Difference between the external scrubber and configuration manager is the command stream and the total number of data bytes to be written. It is considered a blind scrubber… i.e. periodically writes over configuration memory with correct bit-stream (regardless of error).

5.2.2.2 Operational Flow

The configuration bit-stream is comprised of a series of commands followed by the actual configuration mapping data. The configuration controller is expanded to contain a scrubbing mode. While scrubbing, the controller discards the command portion of the original configuration bit-stream and creates its own in order to avoid bringing down the FPGA (and to over-write the interface registers to place them in the appropriate modes for scrubbing). Following the command segment, the original bit-stream that contains the configuration data is ported to the FPGA via the SelectMAP interface. The extension to the configuration controller is minute in order to implement a hardened scrubber.

The REAG external scrubber does not use configuration read-back and ECC circuitry in order to correct (contrary to the Xilinx internal scrubber). Instead, the external scrubber (while in correction mode) only performs writes to the configuration memory. The golden (modified) configuration bit-stream is periodically written to the configuration memory regardless of potential faults as long as the faults do not disrupt configuration interface control. The periodicity of configuration writes is user-programmable.

5.2.3 Internal Scrubbing

5.2.3.1 Brief Overview

The internal scrubber is contained in the DUT. It is an implementation of the Xilinx SEU_cntrl IP core block. The internal scrubber is not a blind scrubber, it requires reading a configuration memory frame, performing Error Correction and Detection (EDAC), and then writing the corrected frame into configuration memory if a fault was detected in that frame. Because there are so many steps to be performed, average correction time is over 100ms. Detect time can vary (depending on where the fault has occurred, time to perform EDAC, and if there frequency of generated configuration memory faults).

5.2.3.2 Operational Flow

Its operation is as follows: The configuration memory space is divided into frames containing a corresponding parity syndrome (also stored in configuration memory) **Error! Reference source not found.**. Once the SEU controller commences operation, frames are read one at a time and a SECDED ECC is performed **Error! Reference source not found.**. If a single error within a frame is detected, the SEU DETECT (Figure 1) flag will go high, after the correction is calculated and the corrected frame is written into configuration memory, the SEU DETECT will go low. If there is a double bit error the ECC circuitry will raise the SCAN ERROR Flag (see Figure 1).

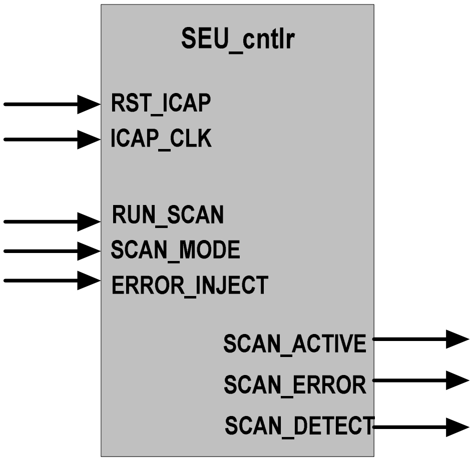


Figure 7: Top-Level Block Diagram of Internal Scrubber

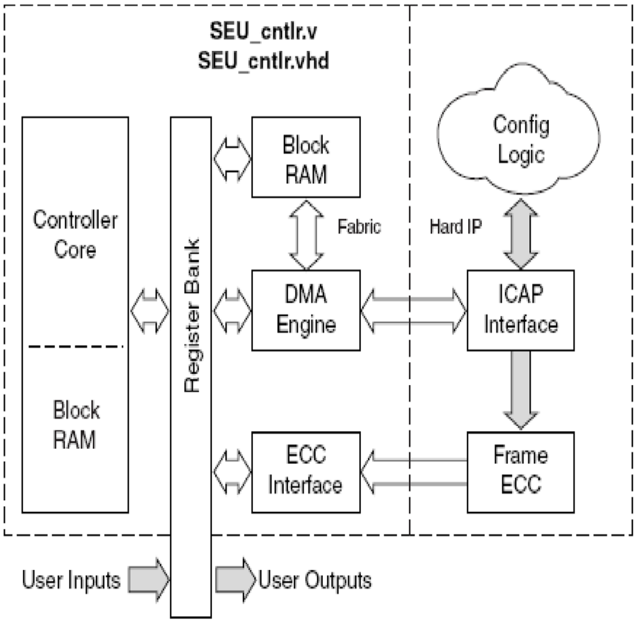


Figure 8: Sub Modules of Internal Scrubber

5.2.4 External Scrubbing vs. Internal Scrubbing

The performance of the internal scrubber verses external scrubbing was of interest in order to supply NASA/GSFC Space Cube project with the corresponding SEE data. Tests were run in both modes including no scrubbing.

5.3 User Commands and Control

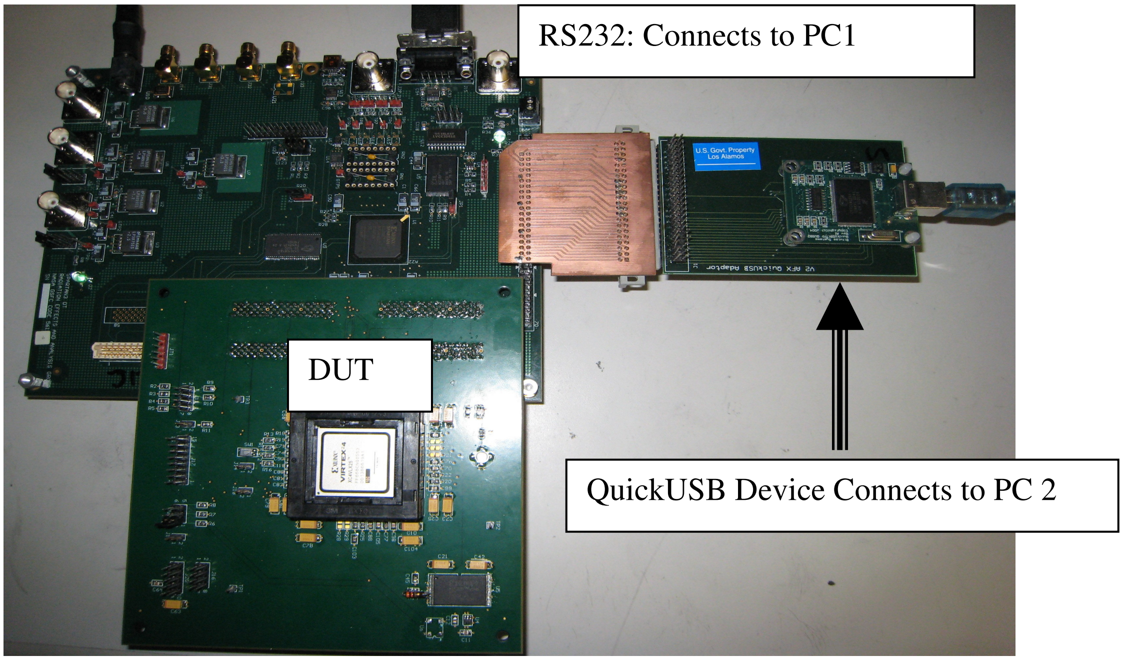


Figure 9: Tester and Xilinx DUT Board

The primary method of which the User controls the tests is via a LABVIEW interface(see Figure 4 for picture of LabView GUI) running on a host PC (noted as PC1 in Figure 9. PC 1 communicates with the LCDT with a RS232 serial link. The format of communication is a command/Data 4 byte word (see Table 6 : Summary of Commands Used in Virtex-4 LX25 Tester).

Table 6 : Summary of Commands Used in Virtex-4 LX25 Tester

Command #Hex	Command	D0	D1	D2	Description
01	Reset DUT	N	N	N	Resets VIRTEX-4 LX25
02	Start Test	N	N	N	Starts VIRTEX-4 LX25 clock and data generation
90	Pattern Number	Y	N	N	0,1,or checker board
A0	Clock Frequency	Y	N	N	Clock frequency divider of 150mhz
81	Write Configuration Data	N	N	N	PC sends configuration data to Tester via RS232 port. Tester stores data in onboard SRAM
04	Start configuration	N	N	N	Tester configures the LX25 by sending the downloaded configuration data to the LX25 via the SelectMap parallel port.
06	Start Scrub	N	N	N	Scrubbing is turned on
0E	Inject error On	N	N	N	Scrubber will turn on error injection
7A	Scrub Error High	Y	Y	Y	Upper address bound for error inject
79	Scrub Error Low	Y	Y	Y	Lower address bound for error inject
05	Start Readback	N	N	N	Turns readback on
7C	Time Between Readbacks	Y	Y	Y	Time period in ms between each readback. From end of readback to start of next readback
7B	End of Configuration	Y	Y	Y	Memory Location (relative to configuration file) to stop scrubbing – avoids BRAM
89	Set SelectMap Control Register	Y	Y	Y	24 bit value to be placed in control register
8A	Set SelectMap Mask Register	Y	Y	Y	24 bit value to be placed in Mask register
AA	Do Scan	N	N	N	Start the internal scrubber

The following sections are detailed descriptions of commands and their associated functionality.

5.3.1 RESET DUT:

The RESET DUT command is decoded as x01. The following represents the command as noted in Table 6:

:

x01	xx	xx	xx

Figure 10: Reset Command Format – Command Number, D0, D1, and D2

Once decoded, all DUT inputs will go into reset mode (Reset, CLK_SR and D_SR are low; SelectMap Interface and QuickUSB interface are in reset)

5.3.2 START TEST:

START TEST is decoded as x02. The following represents the command as noted in Table 6:

x02	xx	xx	xx

Figure 11: Start Command Format

All other commands should be supplied before start test. I.e. the user should define the pattern and clock frequency before administering a start. This command activates the CLK_SR and D_SR DUT inputs. DUT must be configured before this command is sent.

5.3.3 PATTERN NUMBER:

There are three data patterns that can be generated by the tester. Data can be a static 0, a static 1, or alternate every VIRTEX-4 LX25 clock cycle (checker board). The command number is x90. The first byte of data (D0) is also decoded (all other bytes are ignored but must be supplied – i.e. all commands must be 4 bytes of data).

x90	x00	xx	xx

x90	x01	xx	xx

x90	x02	xx	xx

Figure 12: Pattern Command Format

D0 decode is as follows:

X00: Static 0
X01: Static 1

X02: Checkerboard Pattern

5.3.4 CLOCK FREQUECY:

The clock frequency command is decoded as xA0. The following represents the command as noted in Figure 13:

xA0	xnn	xx	xx

Figure 13: Clock Frequency Command Format

Upon the receipt of this command, D0 is used as a clock frequency divider. This command must be sent after a RESET DUT and before a START TEST. D0 must be an even number and must be greater than or equal to 2. The associated output is CLOCK_FREQ. See the LCDT General Tester for more information concerning the processing of CLOCK_FREQ.

5.3.5 Write Configuration Data

5.3.5.1 (Old method used in 2006 tests):

The command is decoded as x81. The PC sends the configuration bit file via the RS232 link to the tester. The bit file is generated from Xilinx IMPACT and is actually the ".bin" file. Bin file must be 977488 bytes long.

x81	xx	xx	xx

Figure 14: RS232 Write Configuration Command Format

5.3.5.2 New Method – QuickUSB 2007 Tests

New method of sending data was used in the 2007 test setup. RS232 link takes roughly 85 seconds. We developed a USB controller using the QUICKUSB parallel to serial device. The full data set is now able to be sent under 1 ms.

5.3.6 Start Configuration

The command is decoded as x04. Once the configuration data has been dumped to SRAM (either by RS232 link or QuickUSB link), the tester is able to configure the DUT. Data does not need to be sent to SRAM every time the user wants to configure. Once bin file is stored in SRAM (and SRAM has not been corrupted by a data run or loss of power), then the user can reconfigure repeatedly.

x04	xx	xx	xx

Figure 15: Start Configuration Command Format

5.3.7 Start Scrub

The command is decoded as x06. Starts externally scrubbing the DUT configuration-memory. DUT must first be configured (along with the corresponding bin file dumped into onboard Tester SRAM).

x06	xx	xx	xx

Figure 16: Scrub Command Format

5.3.8 Inject Error

The command is decoded as x0E. DUT must first be configured (along with the corresponding bin file dumped into onboard Tester SRAM). Start Scrub must be turned on. This command should be used in conjunction with Scrub Error High and Scrub Error low commands. Command will inject error within a range of configuration address spaces. If the Scrub Error High and Scrub Error low commands are not used, Error injection will occur from address x800 to x70000 (relating to actual bin file byte addressing)

.

x0E	xx	xx	xx

Figure 17: Start Injecting Errors Command Format

5.3.9 Scrub Error High

The command is decoded as x7A. Designates the upper bound address of configuration memory error injection. See section 5.3.8.

x7A	MSB	NN	LSB

Figure 18: Error Injection Upper Bound Command Format

Address is 20 bits (19:0). MSB 3:0 is used; MSB 7:4 is unused. NN and LSB 7:0 is used. Address(19:0) = MSB(3:0)NN(7:0)LSB(7:0)

5.3.10 Scrub Error Low

The command is decoded as x79. Designates the lower bound address of configuration memory error injection. See section 5.3.8.

x79	MSB	NN	LSB

Figure 19: Error Injection Lower Bound Command Format

Address is 20 bits (19:0). MSB 3:0 is used; MSB 7:4 is unused. NN and LSB 7:0 is used. Address(19:0) = MSB(3:0)NN(7:0)LSB(7:0)

5.3.11 Start Readback

The command is decoded as x0E. DUT must first be configured (along with the corresponding bin file dumped into onboard Tester SRAM). Start Scrub must be turned on. This command should be used in conjunction with Scrub Error High and Scrub Error low commands. Command will inject error within a range of configuration address spaces. If the Scrub Error High and Scrub Error low commands are not used, Error injection will occur from address x800 to x70000 (relating to actual bin file byte addressing)

.

x05	xx	xx	xx

Figure 20: Start Readback of Configuration Memory Command Format

5.3.12 Time Between Readbacks

The command is decoded as x7C. Designates wait time after a readback has been performed until the next readback will be performed. It is measured in milliseconds and is 24 bits. See section 5.3.11 for start readback command. Time(23:0) = MSB(7:0)NN(7:0)LSB(7:0)

x7A	MSB	NN	LSB

Figure 21: Time between Readbacks Command Format

5.3.13 End of Configuration

The command is decoded as x7B. Designates what byte address (relative to the configuration bin file) to stop scrubbing. Address(19:0) = MSB(3:0)NN(7:0)LSB(7:0)

x7B	MSB	NN	LSB

Figure 22: Last Address Scrub Command Format

5.3.14 Set Control Register and Set Mask Register

The command is decoded as x89 (control register setting) and x8A(Mask Register setting). When scrubbing the Xilinx V4 series, the GLUT MASK bit must be set if SRL or Dynamic RAM is used. Byte 0 can not be changed and is hard-coded in the tester (byte 0 corresponds to the LSByte of the register). Care must be taken because although the bytes are LSByte first, the ordering in the byte is MSB first (7:0).

x89	Byte1	Byte2	Byte3

Figure 23: Control and Mask Register Command Format

5.3.15 Start Scan

The command is decoded as xAA. This command starts the internal scrubber. The DUT should be configured via JTAG (no configuration dump via RS232 and QuickUSB) ;do not use the start configuration ; and do not use the start scrub command

.

xAA	xx	xx	xx

Figure 24: Start Readback of Configuration Memory Command Format

5.4 Configuring the DUT via the Tester

Because the Xilinx device has SRAM based configuration, the user must write the configuration memory. We have decided to control the configuration from the tester. See Figure 25: Tester Controlled Configuration Flow Diagram for a flow of configuration:

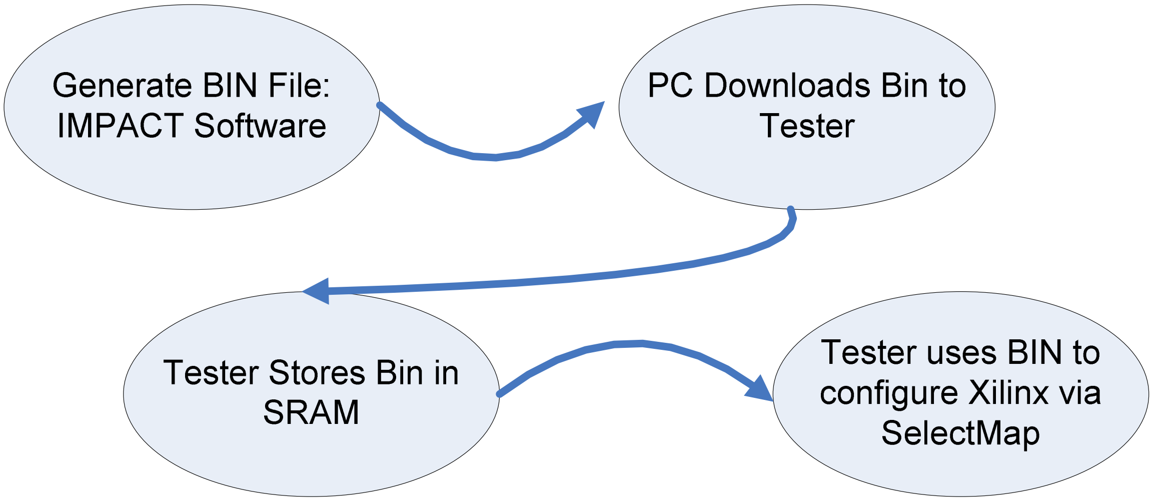

Generate BIN File: IMPACT Software
PC Downloads Bin to Tester
Tester Stores Bin in SRAM
Tester uses BIN to configure Xilinx via SelectMap

Figure 25: Tester Controlled Configuration Flow Diagram

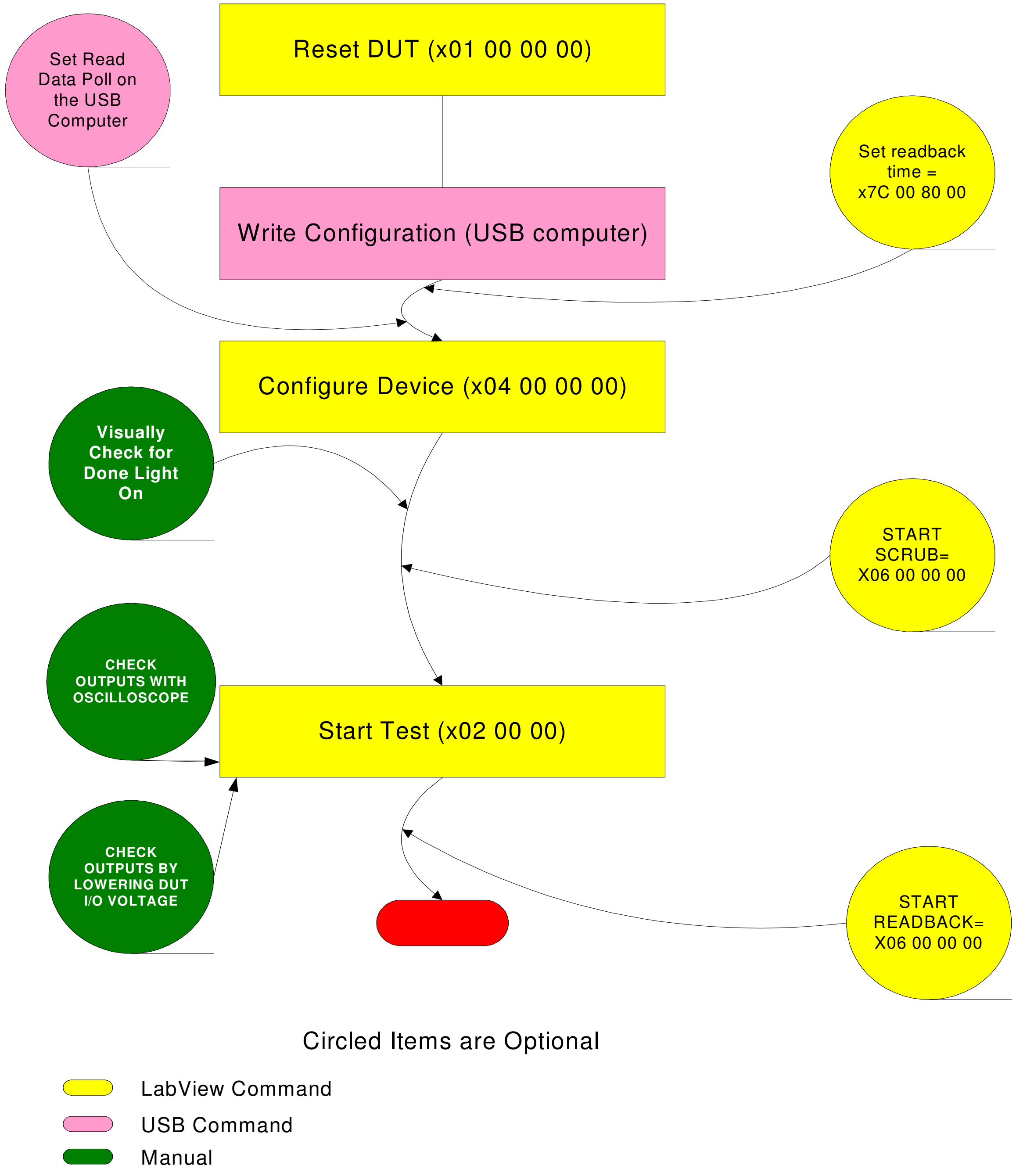
The tester is connected to SelectMap I/O of the DUT and the mode is set so that the Tester is Master and the DUT is slave. 8 bit SelectMap is used at 30 MHz. Before configuration can be performed, the user must dump the ".bin" file to the tester SRAM. This can be accomplished in 1 of 2 ways:

1. via RS232 cable using the write configuration command: x81 (labview environment). This method uses the computer that is running labview. It is part of the regular command stream.
2. via the QuickUSB set-up including the user interface. This method requires a separate computer that is running the Launch USBmain.exe (NASA/GSFC created QuickUsb

GUI). Write data command must be set pointing to the bin file. The software will determine the size of the file and send the correct number of words (16 bit transfers @ 50 MHz).

If using the QuickUSB device, the Tester I/O containing QuickUSB are utilized (connected to the parallel portion of the quickusb device). The serial end of the QuickUSB device is connected to a USB cable that is plugged into the extra computer (PC2). The user must run the Launch USBmain.exe program to invoke the user GUI.

CONFIGURING AND STARTING THE DUT VIA USB


Set Read Data Poll on the USB Computer
Reset DUT (x01 00 00 00)
Write Configuration (USB computer)
Set readback time = x7C 00 80 00
Configure Device (x04 00 00 00)
Visually Check for Done Light On
START SCRUB= X06 00 00 00
CHECK OUTPUTS WITH OSCILLOSCOPE
Start Test (x02 00 00)
CHECK OUTPUTS BY LOWERING DUT I/O VOLTAGE
START READBACK= X06 00 00 00
LabView Command
USB Command
Manual

Circled Items are Optional

Figure 26: Example of Running a Test using the QuickUSB Device

5.5 After Configuration: Running a test

After sending the test parameters and successfully configuring the DUT, the start test command is sent. The beam is turned on after the start test command. Tests are run until a fluence of 1e5 to 1e6 is reached or unrecoverable error is detected (from LabView user interface).

Error cross-section calculation is such that a burst of errors is considered as one event. However, since a burst can have a long duration, the fluence during this period must be subtracted from the total fluence to get an accurate error cross-section calculation. If the DUT becomes unconfigured, the test is terminated. In order to get an accurate fluence count, the tester must stop simultaneously with the beam (as close as possible – flux is low enough to accommodate imperfection). By following this procedure during testing, fluence adjustment can easily be calculated based off of average flux (determined by radiation facility hardware) and duration of configuration corruption (determined by time stamped error information from tester). Example of an invalid test: beam is stopped but tester is still running and returning error information (DUT is unconfigured at this point – errors continue to flow). In order to calculate the approximate fluence to be subtracted off due to the burst, the average flux and the time in error is necessary. In this case, however, the reported time in burst-error (from tester) will be much greater than the duration that corresponds to the actual error's fluence period (time in error while beam is active). The calculated fluence (to be subtracted from the total) will be too large and will result in a much lower (and thus inaccurate) error cross-section.

****** Point of this section: Stop tester at the same time the beam is turned off.**

5.6 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 synchronously capture data using SHFT_CLK (as a data enable) and SCAN_DATA (as a 4 bit window of DUT Data). SHIFT_CLK has a maximum frequency of 37.5MHZ (150 MHZ divided by 4). It is a control signal indicating new data. It is considered asynchronous to the tester and is sampled using the tester's system clock (max 150 MHZ). 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). Once the edge is detected, data is then sampled and registered. The data is then registered again. The comparison is made against the second registered data and if there is a mismatch, the error is reported.

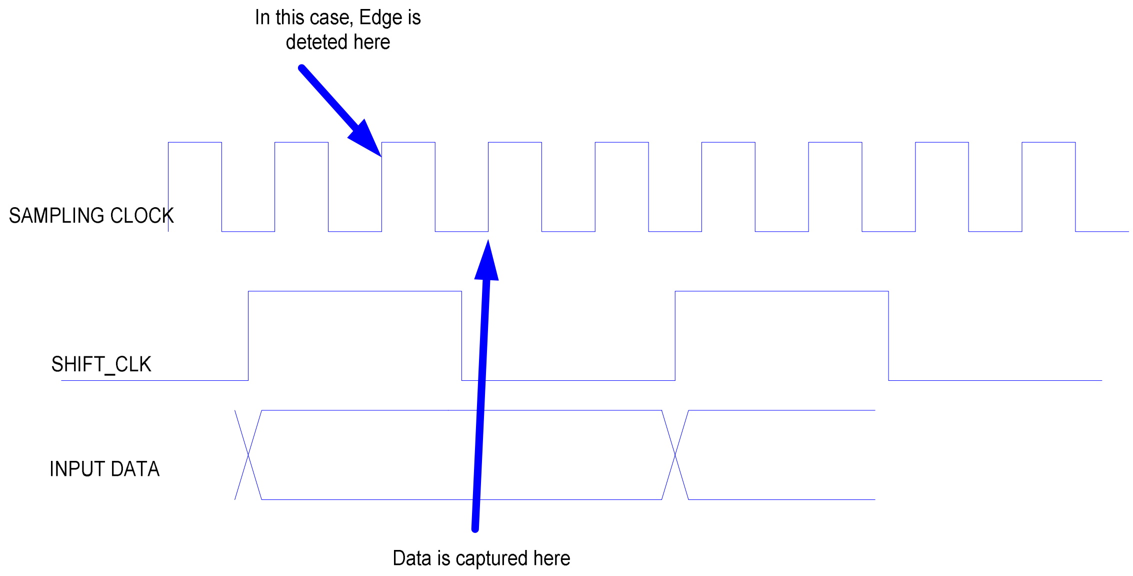


Figure 27: Timing Diagram of Expected DUT Outputs

6. TEST RESULTS:

6.1 Burst Effects and Analysis

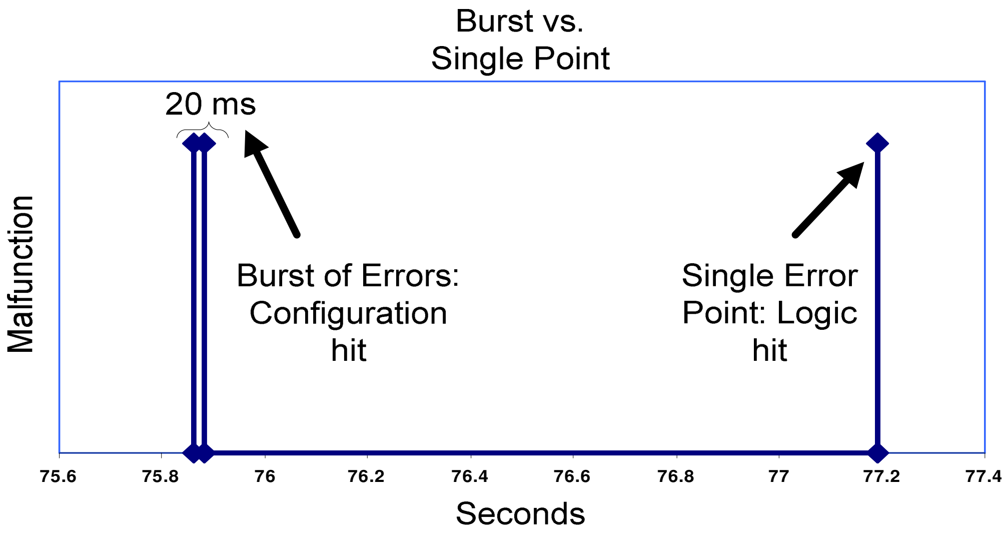


Figure 28: Illustration of Burst vs. Single Point Failures

Figure 28: Illustration of Burst vs. Single Point Failures demonstrates two categories of errors:

(1) Burst: Errors occurring for a long period of time
(2) Single Point: error occurring for one clock cycle of the DUT

A category 1 malfunction occurs when a configuration bit gets hit. It can only be corrected upon reconfiguration or scrubbing. In this case, the external scrubber corrected the bit(s) and normal operation resumed after 20ms. Operational recovery time depended on scrubbing clock rate, scrubbing periodicity, size of configuration memory, and length of shift register string.

A category 2 single point failure occurs when the internal logic portion of the FPGA reverses its state due to a radiation strike and the effect is stored within a flip-flop. The implemented function within the DUT allows all single point failures to be overwritten by the next clock cycle (no enables are utilized).

6.2 Error Cross-Section Calculations

When the output of the DUT is stuck in a burst state, other potential errors will be masked. Due to the long duration of functional error, a true cross-section can not be accurately calculated by the traditional method of number of events divided by fluence (see equation (1)).

$$\sigma = \frac{NE}{TFL} \qquad (1)$$

Because possible error events can be masked during a burst period, an adjustment to the reported fluence must be made for accurate cross section calculations. The proposed modification to equation (1) handles a burst error as if it were one event. The adjustment to the formula is the difference in total reported fluence (TFL) and the fluence accumulated while in burst (TB*FLUX).

$$\sigma = \frac{NE}{TFL - (TB * Flux)} \qquad (2)$$

NE: Number of Events
TFL: Total Effective Fluence: particles/cm^2
TB: Time in Burst
Flux: Approximate reported particle flux (particles/(cm^2*second)).

6.3 Cross Section Analysis

6.3.1 Heavy Ion

Four types of data analysis:

1. Error Cross Section: Calculated using equation (2). Premise is number of error events per particle strike.

2. Meantime to failure: average time between failures. The accelerated environment (flux) must be taken into account before utilization within availability prediction formulas
3. Burst duration: Average time in error (only bursts).
4. Error Rate: failures (failures are not by bit but by device/design) per day

Ar data can be used to calculate analysis information (1 thru 4). Data retrieved during the Kr runs can be used to determine meantime to failure and burst duration but not cross-section. This is due to the fact that the tester remained operational after the beam was stopped… error data were still be computed by the tester (burst time will be too high) thus does not correlate to reported fluence see section 6.2 and equation 2 for further information on fluence, flux, and burst time adjustments. Future tests will be rectified so that higher LET data can be retrieved.

6.3.1.1 Comparison of Scrubbing Techniques at 5.7 LET

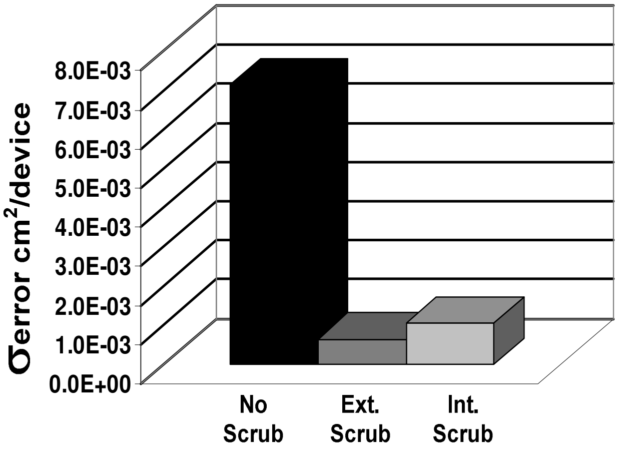


Figure 29: Error Cross-Section per Device. External Scrubbing vs. Internal Scrubbing @ 0 degress Incidence 5.7 LET: Irradiation Test Run #'s:1 (no scrub), 3 (external scrubber), and 17 (internal scrubber)

Figure 29 illustrates Test Results from TAMU with 24.8 MeV/U heavy ion beam (Argon) @ 0 degrees of incidence and LET=5.7MeV*cm^2/mg. Tests were run until a fluence between 1.0E5 and 1.0E6 was reached or the functionality was observed to be unrecoverable from error.

The external scrubber operated at 25MHz. The scrubber cycled though configuration memory approximately every 40ms (time is specific to the Virtex-4 LX25). Correction was accomplished when the scrubber reached a bit that was in error. The average time in functional burst error was calculated to be approximately 16ms. Tests were run with low flux (approximately 100 particles/(cm^2*seconds)) in order to not overwhelm the scrubbing mitigation. The Device Mal-functional cross-section was calculated using equation (2).

The internal scrubber operated at 100MHz. The flux was approximately 100 particles/(cm^2*seconds). The algorithm for correction is complex: a frame is read, SECDED is calculated, if the frame is in error, the processor portion of the scrubber (pico-blaze) writes to the configuration memory. The correction write via the pico-blaze took approximately 100 ms.

Therefore, time in functional burst was relatively longer than the external scrubber because time to correct was in the order of 100's of milliseconds (time in burst includes read-back). The Device Mal-functional cross-section was calculated using equation (2)

The difference in performance (error cross-section) between the external vs. internal scrubbers was relatively similar. This cross-section does not reflect time to unrecoverable failure, it only reflects recoverable malfunction during operational time. Accordingly, although the cross-sections did not have a large difference in value, the external scrubbing was always recoverable without the need for a reset or power cycle (specifically for the shift register DUT design), whereas the internal scrubbing was never recoverable – i.e. faults occurred that were uncorrectable and a reconfiguration was always necessary. Such results suggest that the internal scrubber consistently reached a state where it could no longer correct (either the scrubber circuitry getting hit or MBU's occurring).

6.3.1.2 Error Cross Section Analysis for External Scrubber at 5.7 and 8.1 LET

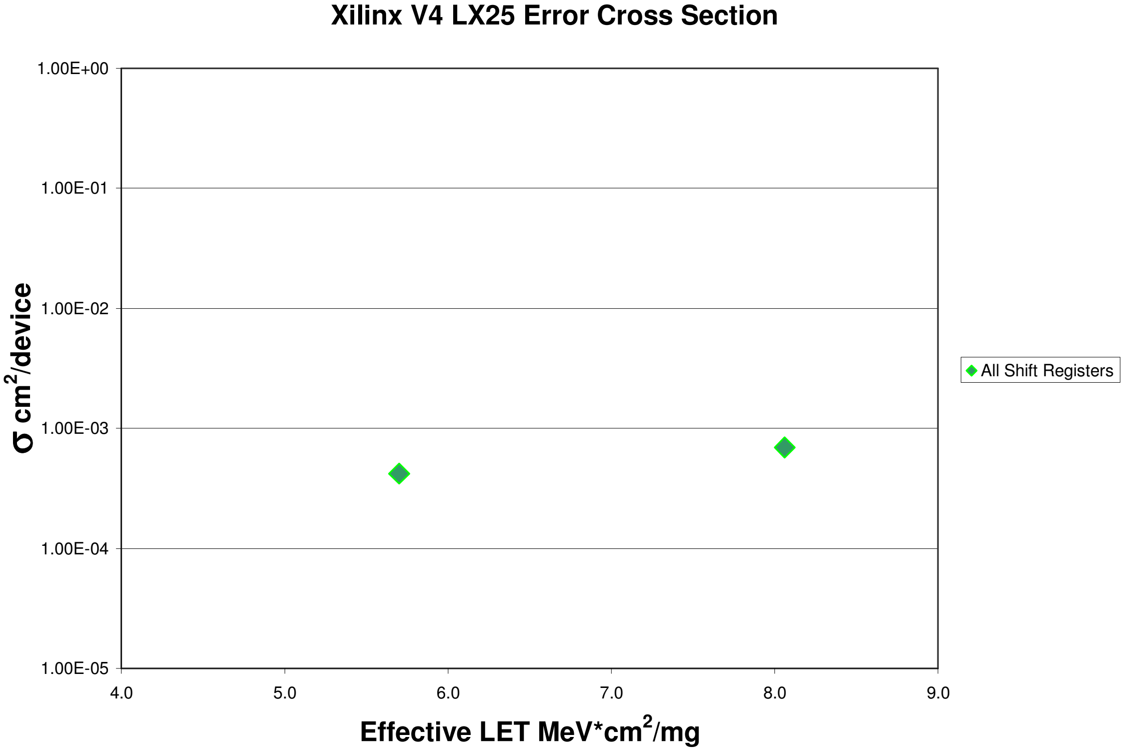


Figure 30: External Scrubber Error Cross-Section vs. LET

Future test will add additional data points to the graph in Figure 30 so that lower (for on-set calculations] and higher (for saturation determination) LET responses can be retreived. Figure 30 demonstrates that no on-set LET has been observed.

6.3.2 Proton

6.4 Resource Analysis

6.4.1 Heavy Ion

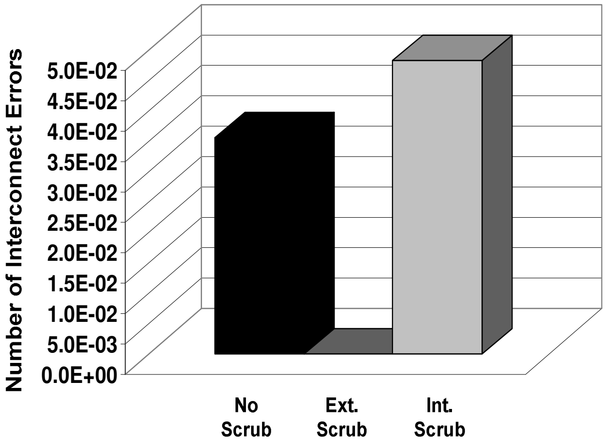


Figure 31: Resource Post Irradiation Test Run #'s:1 (no scrub), 3 (external scrubber), and 17 (internal scrubber)

Resource analysis was performed by reading back the configuration memory after every irradiation test. The NASA/GSFC external scrubber had the best performance as expected (because it is not dependent on SECDED circuitry and it is hardened). The design that used external scrubbing incurred zero interconnect errors (resources not including BRAM) at the end of each test run as illustrated in Figure 31. The design containing the internal scrubber would generally finish with approximately 100 to 500 interconnect errors. This was due to the fact that unrecoverable functionality with the internal scrubber tests was always reached before the target fluence of 1E^5 was obtained. If the circuit is in an unrecoverable state, this generally means that the FPGA has become un-configured. These results suggest that internal scrubber incurred faults and thus corrupted configuration memory (or ceased to function).

Data is currently being analyzed from tests that contained intermediate readbacks during irradiation. This will enable the analysis to have a finer granularity of resource observation.

The most significant information that can be understood from the presented resource analysis is that the external scrubber is the most reliable means of scrubbing (less potential of configuration memory corruption).

6.4.2 Proton

6.5 Bit Error Rate Calculations

Bit error rate calculations will be performed using the **CREME96** at worst case GEO, the device errors/day.

Currently only the Ar cross-section exists. Future tests will sweep the ion cocktail range to Xenon. Associated cross-sections will be calculated using equation 2. The appropriate test methodology of stopping the tester simultaneously with the beam will be implemented. The Bit error rate calculations will be given at that time.

7. FURTHER TEST REQUIREMENTS

.

Additional Data points (data pattern and frequency, with respect to LET) are needed to fill out the data sets for some of the architectures

7.1 Appendix 1:

See www.xilinx.com and contact Melanie.D.Berg@gsfc.nasa.gov for further details: