WP414 (v1.0) April 9, 2012

SEU Emulation Environment

By: Paul Schumacher

Testing for the effects of soft errors can be challenging, frustrating, and expensive. The options are few: accelerated beam testing, which is in the range of $100K per day; or real-world testing, which takes years to produce any useful data.

Xilinx performs both types of testing on their FPGAs and also offers a cost-effective and easy-to-use method for customers to test the effects of soft errors on their own designs.

Introduction

Soft errors can affect configuration memory cells and can, therefore, occasionally affect the operation of a design. Xilinx testing shows that as little as 2%–10% of all soft errors actually cause a system error. This means that it is very valuable for a customer to test and understand the effects of soft errors in different locations of their designs.

The SEU emulation environment enables test and verification for the impact of soft errors on a given design. Two instantiations of a design are included in the overall emulation environment, with each design being driven by the same linear feedback shift registers (LFSRs). Error injections are emulated using the soft-error mitigation (SEM) IP core; the outputs of the two designs are compared against each other on every clock cycle for a specified amount of time. Injected errors and observed function failures are tallied until a specified total has been reached.

When completed, results are relayed to the user with an appropriate derating factor and are calculated based upon the instantiated design and target part.

Hardware Details

Figure 1 shows a block diagram of the hardware design for the SEU emulation environment.

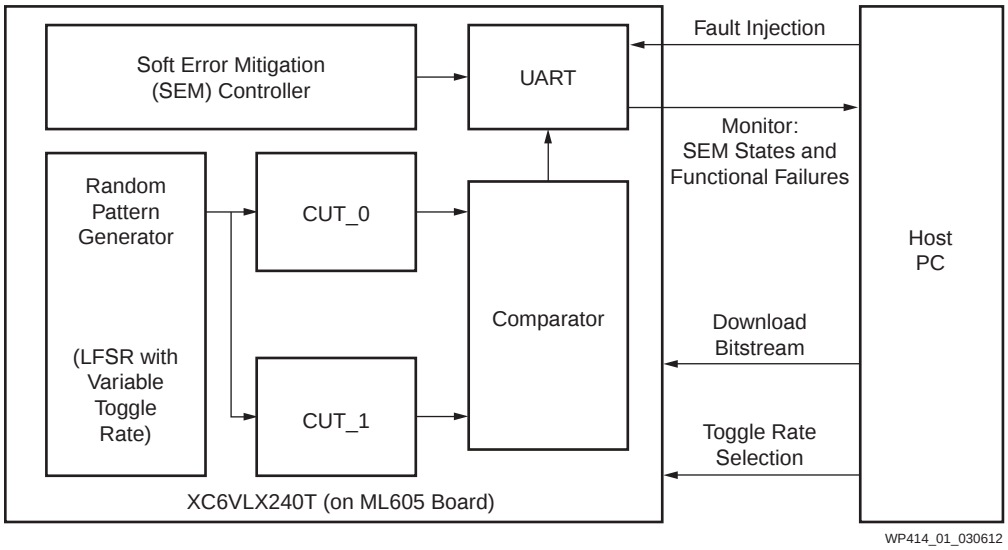

Figure 1: **Block Diagram of SEU Emulation Environment**

The circuit under test (CUT) is the user design under test and is instantiated twice for comparative purposes. The CUT instances are driven by a random pattern generator, which includes a LFSR with variable toggle rate selection. The SEM controller is included for purposes of error injection and correction.

A comparator is also included to compare the outputs of the CUT instances. Every output of the CUTs is compiled into equivalent buses and compared bit-for-bit on every clock cycle. This method provides a worst-case definition of functional failure; therefore, the SEU emulation environment also provides a worst-case value for the derating factor.

A single UART is included to provide a communication channel to the host PC. The receive channel can involve a fault injection command issued from the host PC and

delivered to the SEM controller. This act, performed during normal operation of the SEU emulator, injects errors into the FPGA configuration bits. The transmit UART channel is multiplexed between two sets of data:

1. **SEM controller state bits:** These are transmitted every time one of the following state bits has changed:
   - `injection`
   - `classification`
   - `correction`
   - `observation`
   - `initialization`
2. **Functional failure status:** This is transmitted on the UART every ~0.5s, or whenever a comparator error is detected. The possible status values are all ASCII characters and include:
   - **`0x50` (P)** – healthy signal (no functional failures found yet)
   - **`0x45` (E)** – functional failure has been found
   - **`0x55` (U)** – unrecoverable error found

Software Details

Controlling the SEU Emulation Environment and Running the Software

The software for controlling the SEU emulation environment is run on a host PC. UART-based communication to/from an ML605 board is used to inject errors, monitor the impact on the design, and keep track of the results.

Command Line Options Example

There are a number of user-controlled options for this software. This command line option example lists the command line options for running the program. There are no mandatory options; however, the filename of the FPGA bitstream for the ML605 board must be provided.

```
Usage: seuemu [OPTIONS] file.bit

OPTIONS:
      -c    Specify the COM port for Comparator (default: COM1)
      -f    Specify output file name (default: output.txt)
      -e    Specify the EBD file (default: not used)
      -m    Specify monitoring waiting time (default: 1000 ms)
      -n    Specify maximum number of failures to emulate (default: 100)
      -w    Specify number of words in a frame (default: 81)
      -t    Specify total number of frames (default: 5000)
      -h    Print this help message

Example usage: seuemu -c COM1 -f output.txt -e circuit.ebd circuit.bit
```

If an EBD file is specified using the '**`-e`**' option, then the random locations of the error injections are chosen only from the subset of bits deemed essential by the EBD file. Otherwise, all possible configuration bits are used.

The monitoring wait time using the '**`-m`**' option can be specified to decide how long to monitor for functional failures after each error injection. Limited testing has shown

that anywhere in the range of 1 to 10 seconds is adequate, depending on the functionality of the CUT and the desired emulation run-time.

The maximum number of functional failures to emulate can be specified using the '**-n**' option (default is 100). The proper value for this depends upon the desired accuracy of the calculated derating factor. Before deciding how to set this functional failure count, users are encouraged to review Li, et al., *Online Estimation of Architectural Vulnerability Factor for Soft Errors* [Ref 1].

There are two other command line options, which are family- and part-specific.

The '**-w**' command-line option is used to specify the number of words in a frame on the FPGA. Currently, only the default specification of 81 words is supported, which is compatible with Virtex®-6 devices. Future support might include other FPGA families, in which case the '**-w**' command-line option would become relevant.

The '**-t**' command-line option is used to specify the maximum frame count on the FPGA. This value is used only if an EBD file is *not* specified. The random configuration bit location generation uses this frame threshold to calculate the bit location of the injected error.

Running the Software

Figure 2 shows a flow diagram of the test operations for the SEU emulation environment.

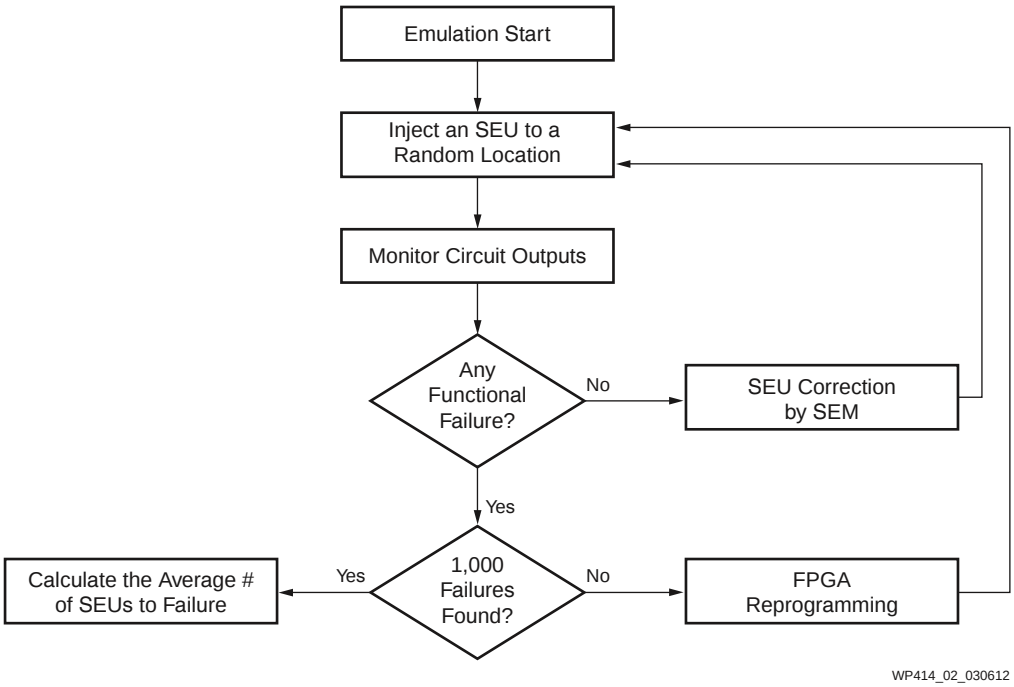

Figure 2: **Flow Diagram of Test Operations for SEU Emulation Environment**

Using the UART connection to the SEM controller shown in Figure 1, errors are injected into the FPGA configuration bits.

The random locations of the injections are chosen from one of two user-specified subsets of bits:

- all configuration bits
- only those bits deemed essential by an optionally specified EBD file (the '**-e**' command-line option)

After every injection, the UART communication channel is monitored for the notification of functional failures (see Figure 2). The amount of time spent waiting and monitoring is determined either by the default (1,000 ms) or by the user-specified number in the '**-m**' command-line option.

A tally of total functional failures and unrecoverable errors is kept. The specified threshold of functional failures is determined either by the default (100) or by the user-specified number in the '**-n**' command-line option.

With the emulation thus completed, the total errors and failures are used to calculate the derating factor of the design.

> ***Note:*** When a functional failure is detected, the recovery mechanism currently employed is to reconfigure the FPGA with the specified bitstream. While this is more time-consuming than requesting the SEM core to attempt to make the repair, it ensures that every new injection begins with a new bitstream.

Evaluating the SEU Emulation Output

This command line example shows an exemplary transcript output after an emulation run has been completed.

```
Total SEUs:       3000
Total failures:    761
  unrecoverable:     3
  func. failure:   753
  comparator:        0
  waiting for I:     0 (before injection)
  waiting for I:     0 (after injection)
  waiting for O:     5 (observation)
  waiting for C:     0 (comparator)

Essential Bits:  14.13%
Derating factor from emulation: 3.58431%
Maximum number of failures found.
Program terminated.
```

An EBD file was provided for this run, therefore, the essential bits in the bitstream was used as the subset of bits used for random injections.

The percentage of essential bits reported in this command line example was read directly from the bitstream report file (BGN) assumed to be in the same directory as the EBD file. The calculated derating factor is calculated using Equation 1:

$$\text{Derating Factor} = \text{Essential Bits} \bullet \frac{\text{Total Failures}}{\text{Total SEUs}} \qquad \textit{Equation 1}$$

Equation 1 provides an approximation of the derating factor because the cycle-by-cycle comparison for functional failures can be too strict. Also, only a subset of bits is used for emulating soft errors. This subset can have an impact on accuracy [Ref 1]. However, the total number of errors emulated only needs to be in the range of 2,000–3,000 to provide reasonable accuracy.

Conclusion

The effects of soft errors on an actual system can vary greatly. With the emulation tools provided by Xilinx, customers are in a much better position to maximize their availability and reliability of their system.

References

1. X. Li, et al., *Online Estimation of Architectural Vulnerability Factor for Soft Errors*, Proceedings of the International Symposium on Computer Architecture (ISCA), June 2008.

Related Information

Xilinx, Inc., UG764, *LogiCORE™ IP Soft Error Mitigation Controller User Guide*, v2.1, June 22, 2011.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
04/09/12	1.0	Initial Xilinx release.

Notice of Disclaimer