Improving the Reliability of Xilinx 7 Series FPGAs through Configuration Scrubbing*

Alex Harding and Michael Wirthlin
NSF Center for High-Performance Reconfigurable Computing (CHREC)
Department of Electrical and Computer Engineering
Brigham Young university, Provo, UT 84602
zaren171@gmail.com, nelson@ee.byu.edu, wirthlin@ee.byu.edu

April 16, 2014

Abstract

Scrubbing is a process in which a memory is systematically read, checked for errors, and corrected when an error is found. Scrubbing is used as a form of mitigation against data corruption that may be induced by internal or external stimuli.

Radiation is an example of an external stimulus which can corrupt data values in semiconductor devices. Some environments, such as space, have high levels of radiation. When using semiconductor devices in a high radiation environment scrubbing can be used to help mitigate the effects the radiation has on the circuitry.

The development of two FPGA configuration scrubbing architectures are discussed. The limitations of the first system are presented and a second system designed to resolve these issues is presented. A configuration scrubbing system suitable for space flight is also presented.

1 Introduction

Field Programmable Gate Arrays (FPGAs) provide a flexible platform for hardware design. An FPGA is an integrated circuit that is designed to be configured by the user. FPGAs store user configuration in a configuration memory, or CRAM. Programming different configurations to an FPGA allows the device to perform different logic operations. Xilinx's 7 series FPGAs provide large configurable regions along side many specialized circuitry. These circuits include block memories, DSPs, multi-gigabit transceivers, and processor cores [1].

The addition of these complex fixed function circuits allows FPGAs to perform advanced tasks which must be run at higher speeds than can be accomplished in the FPGA configurable logic. Operating electronics in radiation environments provides unique challenges to hardware development. Most hardware used in radiation environments must be custom made. FPGAs provide a flexible platform for the development of such custom hardware. The fixed function circuits make FPGA based designs more robust and practical. These fixed function units combined with the flexibility of the configurable logic have generated much interest in using this latest generation of FPGAs in radiation environments.

Radiation, unfortunately, does not interact well with semiconductor devices. Radiation causes upsets in semiconductor devices. An upset is caused when a radiation particle changes the state of a transistor, causing it's value to transition. For FPGAs specifically these upsets can cause changes in the CRAM where the configuration of the FPGA is stored. These upsets can break or change the functionality of the programmed circuitry. As an FPGA continues to operate in a radiation environment the upsets in the configuration memory will accumulate, causing increasingly severe issues for the configured design.

FPGA configuration is often scrubbed in order to mitigate against these configuration upsets. Scrubbing is a process in which a memory is read in small chunks, checked for errors, and overwritten with correct data when errors are found. Scrubbing occurs while the FPGA is on and operational.

This work will develop scrubbers which are capable of detecting and correcting configuration errors found in 7 series FPGAs. These scrubbers provide the benefit of correcting errors that occur in the configuration memory of an FPGA while operating in a radiation environment. Correcting errors ensures that the design

*This work was supported by the I/UCRC Program of the National Science Foundation under Grant No. 0801876

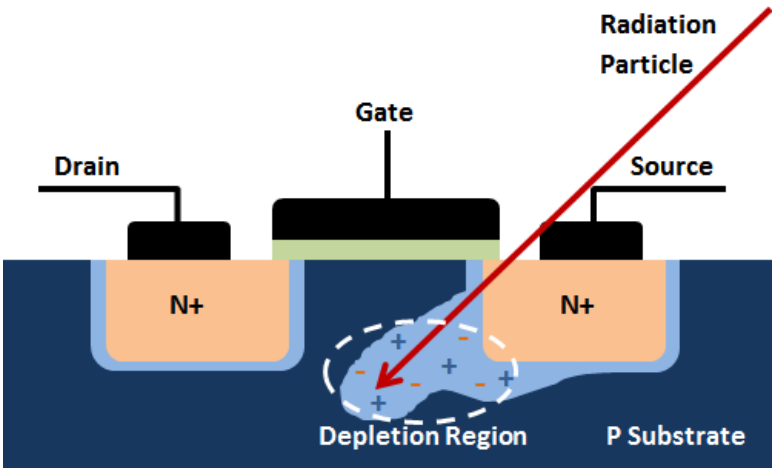

Figure 1: A radiation particle strikes a silicon device creating a depletion region within a transistor.

will resume correct operation shortly after an error occurs.

This work seeks to improve on past work by developing a scrubber that operates at speeds similar to hardware based scrubbers, but without the addition of specialized circuitry. To accomplish this basic scrubbing is performed internal to the FPGA. This allows the scrubbing to happen very fast. Unfortunately this internal scrubbing is limited in its functionality. Using the external interfacing capabilities already available to the FPGA a scrubber with full scrubbing functionality is used to correct errors that the scrubbing internal to the FPGA cannot. This external scrubber runs much slower than the scrubber within the FPGA, however, it is only needed occasionally to fix an error, allowing the complete system to operate at an accelerated speed.

2 Background

Radiation is comprised of particles with a great deal of energy. When these particles collide with other matter they transfer energy to the matter they interact with. This added energy can disrupt the structure of the materials it is added to.

When high energy particles strike the silicon lattice in semiconductor devices electrons are broken free. The area in the silicon where the electrons are displaced is known as a depletion region. This electron displacement produces free electrons and holes. Within the depletion region of the particle strike the flow of electrons may be either enhanced or inhibited, depending on the orientation of free electrons and holes generated. An example of a radiation creating a depletion region is shown in Figure 1.

If the depletion region occurs close to a transistor the effects of the depletion region can disrupt the operation of the transistor. These disruptions may cause the transistor to either turn on or off. If the functionality of the transistor is interrupted a Single Event Upset (SEU) has occurred. SEUs alter the functionality of digital circuitry by switching signals. If an SEU occurs within a memory element then the upset will be held in the system until the memory location is overwritten with a new value.

The configuration of a reprogrammable FPGA is stored in a memory structure. As an FPGA is bombarded with radiation the configuration will change. These changes to the configuration will break the circuitry programmed to the FPGA. If these upsets are left unmitigated even using multiple copies of the hardware will eventually succumb to the erroneous configuration.

Configuration scrubbing is used to combat this accumulation of errors. As the FPGA operates the configuration can be read and overwritten. There are several ports available on the FPGA to access the configuration. The main ports are the ICAP, SelectMAP, and JTAG.

The ICAP is available internal to FPGA designs. Using the ICAP the FPGA is capable of reading and modifying the configuration internally. Unfortunately, FPGA designs using the ICAP are susceptible to the same radiation upsets they are developed to fix. For such designs multiple copies of the scrubbing state machine will be needed. There will still be a single point of failure where the logic connects with the ICAP interface at which a single upset could hang the system.

Both SelectMAP and JTAG are external ports for accessing the configuration. The SelectMAP is a par-

allel interface, allowing for 8, 16 or 32 bit wide connections. SelectMAP interfaces typically require additional external hardware to be attached to the FPGA. Such implementations would require a custom PCB board to be manufactured that establishes direct connections with the SelectMAP pins.

JTAG is a serial interface to the configuration. JTAG is the most common method of accessing the FPGA configuration. Most FPGA programming happens over a JTAG port. The JTAG port provides the easiest method to access the configuration. Since the JTAG port is the primary method used for FPGA configuration there are an array of commercially available methods to access an FPGA through JTAG. The JTAG port will be used for scrubber development in this work.

The process of scrubbing a memory includes reading the memory, checking for errors, and correcting errors as needed. It is easier to scrub a memory in small segments than all at once. The FPGA configuration can be read in small increments called frames. A configuration frame is 101 32-bit words long and the number of frames in an FPGA varies part to part. Each frame has a unique address with which it can be read.

The scrubbing process for FPGA configuration scrubbing through a JTAG port is presented in Figure 2. First, a JTAG connection is established with the FPGA. Then the FPGA configuration is read from the device providing a golden copy to compare subsequently read data with. Once the golden configuration has been read and stored the device is read frame by frame.

When a frame is read it is checked for an error. If no error is present the scrubber proceeds to the next memory location. If an error is found it is recorded and the correct data is written to the frame from the golden configuration previously stored.

It has been found that some bits on an FPGA cannot be overwritten without re-powering the device. These bits are known as stuck bits. To check for stuck bits after the correct configuration data has been written to a frame the frame is reread and checked for errors. If the same errors are found then the golden configuration is updated with the new values of the stuck bits. This is done to prevent the same error from being reported every time the scrubber cycles through the device. Once an error has been corrected, and the golden data updated as needed the scrubber then continues to scrub the next frame.

To perform basic scrubbing the addresses are systematically read, from first to last. As each frame

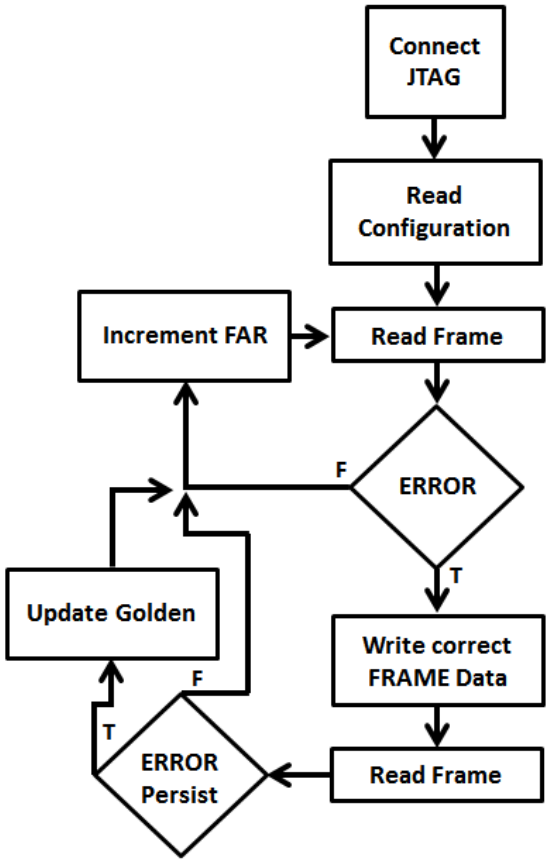

Figure 2: The flow of FPGA configuration scrubbing.

is read the system checks the output data for errors. When the last frame of the device has been read the process is repeated again from the beginning. Eventually, in a radiation environment, a particle strike will cause an upset in the memory. As the scrubber continues to read small blocks of the memory it will eventually encounter the upset memory location. When the upset frame is read the error will be detected. Once the error is detected the error is fixed by overwriting the data. The scrubbing process is illustrated in Figure 3.

Reading the memory in small pieces allows the scrubbing process to pause as needed. This is useful in a space environment so that power can be conserved and scrubbing can be performed as needed. In addition, when an error is found the smaller the memory region being scrubbed the less work needs to be performed to correct the error.

When scrubbing an FPGA there are a few additional consideration to take into account. Not all frames are used in every design, and some informa-

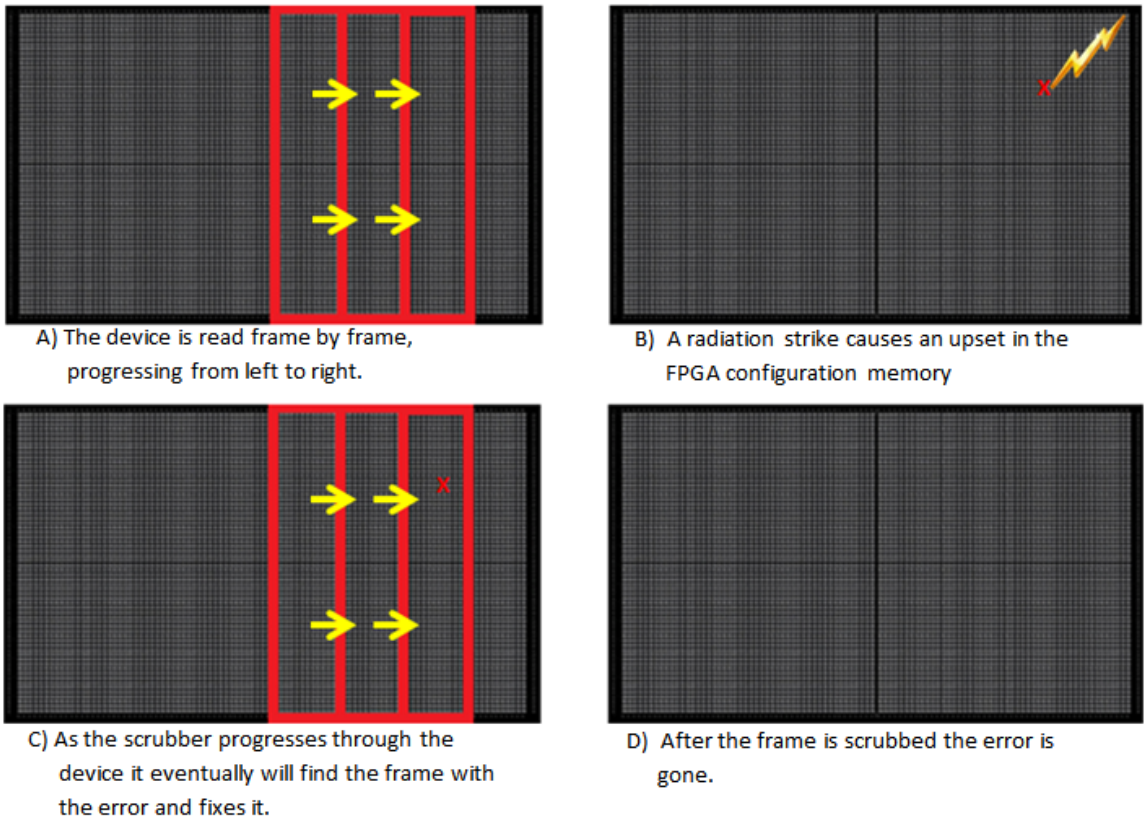

Figure 3: An example of scrubbing.

tion in the configuration changes as FPGA runs. For example the data stored in a register in a state machine will change throughout execution of a design. These changes would look like errors when compared to the golden copy of the design. These bits, however, are a part of the proper operation of the design, and if they are corrected by the scrubber the flow of the design will continuously be interrupted. To handle this a mask file may be generated along side the FPGA programming file. This mask file contains a one or zero for every bit in the configuration indicating whether or not the configuration bit is static or changing/unused.

As the feature size used in fabrication of FPGAs shrinks radiation upsets become increasingly likely in FPGA configuration cells. Smaller feature sizes also result in higher rates of multi-bit upsets. A multi-bit error occurs when one radiation particle strike causes multiple elements to change value. Radiation particles of similar energy will create similar size depletion regions within silicon. As the size of transistors shrinks the depletion regions can span multiple transistors, upsetting multiple bits at once. This can become a serious issue as many of the corrections schemes typically used are designed to only correct single-bit errors.

Xilinx 7 series FPGAs have introduces measures to combat these issues. 7 series devices contain an internal configuration scrubber which can be configured to read and optionally correct errors found in the configuration. This scrubber is able to correct single bit errors and detect multi-bit errors. Our preliminary radiation testing shows that multi-bit errors occur with a frequency of about one in six errors. These multi-bit errors will halt the functionality of the internal scrubber. The scrubber will not resume until an external means has corrected the error in the configuration and reset the internal scrubber.

In the 7 series the configuration frames have been interleaved. This interleaving is meant to spread multi-bit errors into two frames. The hope is that multi-bit errors will only cause one upset in each frame, allowing single error corrections schemes, such as the internal scrubber, to fix these errors. Interleaving frames does help to mitigate against multi-bit errors. Unfortunately, in radiation testing it is seen that multi-bit errors still do effect the system. The

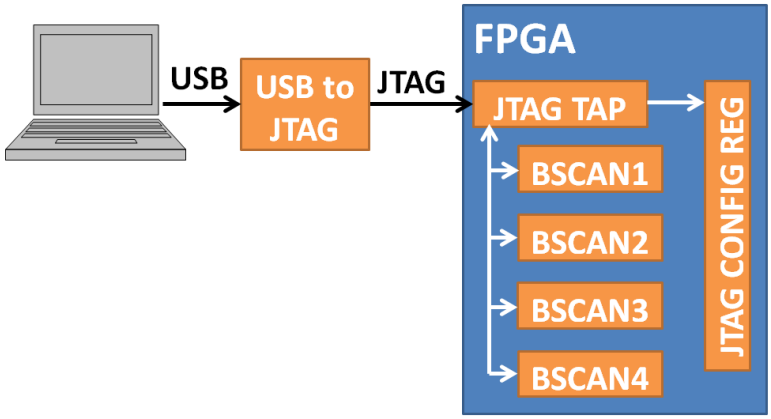

Figure 4: Setup of the JTAG scrubbing architecture.

frequency of multi-bit errors was reduced from one in six to one in ten events through frame interleaving. Frame interleaving is represented in Figure 5.

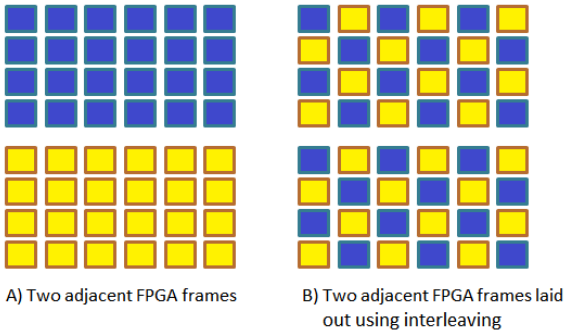

Figure 5: An example of frame interleaving.

The occurrence of multi-bit errors in Xilinx 7 series FPGAs require a more robust scrubbing mechanism to handle the harsh environment of space. This work seeks to develop FPGA configuration scrubbers that are capable of correcting multi-bit errors.

3 JTAG Scrubbing

In this work the JTAG port was used to facilitate scrubbing. A Digilent USB to JTAG conversion chip was used to allow a computer to communicate with the FPGA. Digilent supplies an SDK that can be used to manipulate the JTAG connection through a USB port on the computer [2]. A block diagram of the scrubbing architecture can be seen in Figure 4. In this figure the USB to JTAG block represents the Digilent chip. The boxes within the FPGA represent the various JTAG registers available. The BSCAN registers can be used to feed information from the internal design to the scrubber. The JTAG configuration register is the means by which the JTAG port accesses the FPGA configuration.

The JTAG scrubber developed in this work operates on a frame by frame basis. The scrubber reads the configuration, one frame at a time, starting from the beginning of the devices. When the last frame was reached the scrubber would start again at the first frame and scrub the device again. When each frame was read it was compared against the mask file to see which bits were static. The static bits were then checked against a golden version of the bit file to determine if upsets had occurred. Any errors found were logged for later processing and the corrupted content was overwritten.

This JTAG scrubber was tested at The Svedburg Laboratory in Uppsala, Sweden from May 14-19th of 2013. The FPGA was exposed to both protons and neutrons. A triplicated circuit was implemented on the FPGA and places in the beam. The triplicated system used voters to determine if any one of the three identical copies of the circuit experienced a radiation upset. These scheme of design is called triple modular redundancy (TMR). A count was kept for each circuit to determine the number of times it experienced and error. In addition there was a count kept to determine if TMR failed, or, in other words, if two of the three systems experienced an error simultaneously. The scrubber was found to be successful because while the circuit experienced errors in the individual systems the scrubber was able to prevent any TMR failures.

The device used for testing was a XC7K325T. For this device the JTAG scrubber to approximately two minutes to make a complete pass through the device configuration. This is slow when compared to the internal scrubber that can check the entire device within

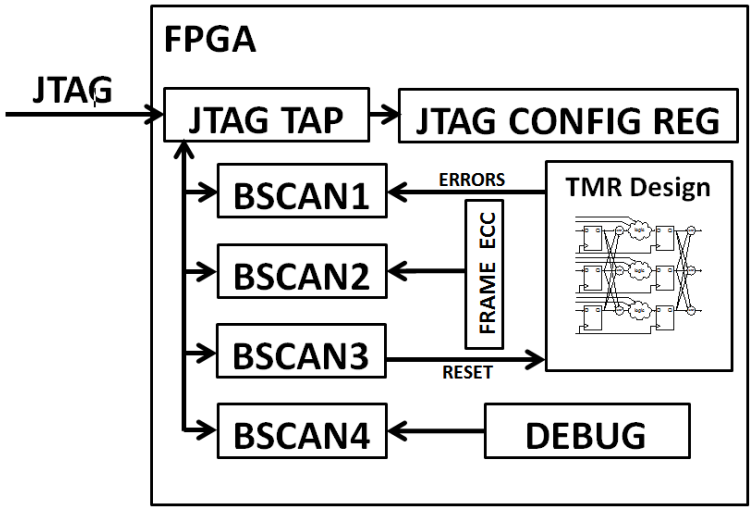

Figure 6: The hybrid scrubing system internal to the FPGA.

30 ms. These times can be used as worst case durations from an error occurring to correction. While the bottleneck can partially be attributed to the serial nature of the JTAG connection there exist two other issues.

Firstly, the JTAG has a maximum clock rate of 33MHz. This rate is much slower than hardware alternatives can operate at. Second, when using the Digilent SDK arrays of data must be built to pass to through the JTAG connection. Data is returned in a similar array. Unfortunately this returned array is not byte aligned, and is returned in reverse order. All data read through the JTAG port must therefore be shifted and flipped before it can be used. Large portions of time are spend in array manipulations trying to check if the read data is correct.

While this basic JTAG scrubber performed well in the beam it is unlikely a system could take a two minute break in operation to facilitate configuration scrubbing. A faster method of scrubbing was needed.

4 Hybrid Scrubber

As mentioned in the previous section, the internal scrubber of 7 series FPGAs is capable of checking the entire device configuration every 30 ms. This hardware block provides an excellent means to speed up device scrubbing speed, though multi-bit errors will still need to be handled. The JTAG scrubber, though slow, is more than capable of fixing all errors, no matter how severe, as it scrubs the configuration. By combining these two scrubbing techniques a fast yet robust scrubbing system was developed.

For this combined scrubber the internal scrubbing circuit is configured to correct all single bit errors it encounters. When in this mode the internal scrubber will also broadcast any multi-bit errors it encounters. The internal scrubber will freeze on any error it does not correct until the configuration is accessed through another port. Once the configuration has been accessed externally the internal scrubber begins scrubbing again starting at the first frame of the device. Whenever the internal scrubber finds an error it pushes all error information into a FIFO that feeds data into a JTAG register. A block digram of the scrubber internal to the FPGA is shown in Figure 6. For this design many hardware components have been connected to the BSCAN ports of the JTAG. The scrubber then reads in information from these JTAG ports to get the status of the system.

The software on the computer continuously reads the JTAG register looking for error information. If error information for a single bit error is seen it is logged to a file. When multi-bit error information is encountered the computer performs a scrub on the indicated frame. Once the errors have been found and corrected the fixed bits are logged.

The hybrid scrubber was tested at LANSCE in Los Alamos, NM on September 17-20th 2013 in a neutron beam. This test showed that the hybrid scrubber worked correctly, and fixed errors much faster than using JTAG alone.

During the testing the scrubber was able to correct all events encountered. Table 1 shows the results of the radiation testing. From this table it can be seen

that the internal scrubbing mechanism fixed 80% of all upsets that occurred within the FPGA configuration. Only one fifth of the events required the JTAG connection for recovery.

Table 1: Duration of Recovery Mechanisms.

Event Type	Count	% of Errors
Single-bit	758	80.55%
Multi-bit	183	19.45%

From previous radiation testing efforts it would be expected that the internal scrubbing mechanism would be able to fix 90% of configuration errors encountered in the beam testing. The results presented in this table are based on a relatively small number of errors, approximately 1000. The previous testing data that was used to determine the multi-bit upset frequency consisted of 50,000+ events. The relatively small data set used in this work leaves room for error in the frequency of multi-bit upsets. However, in this testing all single bit error were recovered by the internal scrubbing mechanism, and the scrubber was able to correct all events that occurred validating the scrubbing architecture.

5 Future Work and Conclusion

In this work a quick yet robust scrubbing architecture was developed for 7 series Xilinx FPGAs. This scrubber combined the high speed scrubbing abilities of the FPGA internal scrubber circuit with occasional slow multi-bit correction. This scrubber is very useful for high radiation environments as it ensures user designs remain correct.

Part of Xilinx 7 series of devices is an FPGA known as ZYNQ. What sets ZYNQ apart from the rest of the 7 series FPGAs is that the ZYNQ FPGAs contain dual ARM processor cores. These ARM cores have made the ZYNQ a highly attractive design platform. Moving forward it would be useful to implement the hybrid scrubber presented in this work on a ZYNQ FPGA. The ZYNQ processor has an alternative method of accessing the configuration called the PCAP. The PCAP provides a parallel port from the processor core into the configuration logic. Since most of the code for JTAG scrubbing centers around arrays the code could easily be ported to use the PCAP. Implementing such a system would create a robust scrubbing infrastructure that is completely contained on the FPGA. The ZYNQs internal ARM cores would remove the need for a remote computer to probe the configuration to fix multi-bit errors. Implementing the hybrid scrubber on a ZYNQ chip would result in an FPGA capable of handling the rigours of space.

References

[1] 7 Series FPGAs Overview. Technical report, Xilinx, Inc., February 2014. DS180 (v1.15).

[2] Digilent Port Communications Programmers Reference Manual. Technical report, Digilent, Inc., June 2005. Doc: 576-000.