AUTOMATIC GENERATION OF SEU-IMMUNITY FOR FPGA BASED ELECTRONICS FOR ACCELERATORS *

M. Grecki, DESY, Hamburg, G. Jablonski, W. Jalmuzna, D. Makowski, DMCS-TUL, LODZ

Abstract

The modern accelerator control systems nowadays are build using digital technology based on FPGA circuits. However, digital circuits working in radioactive environment are exposed to disturbing effects, in particular neutron radiation causing SEU (Single Event Upset). The countermeasure is a redundancy in circuit that allow to detect and correct errors caused by radiation. Unfortunately CAD software provides no support to automatically include required redundancy in the FPGA project. Moreover, optimization procedures remove redundant parts and special effort must be made to prevent that. The paper presents a software environment to process VHDL description of the circuit and automatically generate the redundant blocks together with voting circuits. The generated redundancy uses Triple Module Redundancy (TMR) scheme. It also supports the VHDL simulation with SEUs in order to enable identification of the most sensitive components. Since the TMR is costly, the designer can indicate which parts of the circuit should be replicated basing on the results of simulation.

INTRODUCTION

The XFEL (X-Ray Free Electron Laser) machine [1] will be built build in the next few years at DESY Hamburg. It will be built with single tunnel and the control electronics will be placed in the close proximity to the accelerator pipe. Therefore electronic circuits will be exposed to gamma and neutron radiation. While gammas reduce the lifetime of the circuits neutrons can cause the temporary malfunction during circuit operation [2].

The temporary effects (Single Event Effects - SEE) are caused by charged particles energetic enough to ionize the semiconductor by generating excessive electron/hole pairs [2]. However, also neutrons can generate SEE indirectly through neutron capture initiated nuclear reaction resulting in emission of charged particle [3]. The typical nuclear reaction of neutron and nucleus of boron ^{10}B (about 20% of boron commonly used as silicon dopant) creates α particle energetic enough to ionize the semiconductor on the way through (the typical range of α particles in silicon is several microns). Those ionized areas of semiconductor can conduct electric current until the excessive electric carriers recombine and thus random parts of the electronic circuits can be exposed to current pulses disturbing normal operation. Most of the SEEs are temporary but in some cases SEE can cause permanent (but reversible after power turn-off/turn-on cycle) Single Event Latchup (SEL) or even non-reversible Single Event Burn-out.

The Single Event Transients (SET) [4] caused by temporary ionization of the semiconductor resulting in current pulses can propagate through the circuit logic and generate faulty signals. These disturbances are only temporary and after a short period (usually below a nanosecond) the SET vanishes. Different situation occurs when the SET is locked in a data storage component (flip-flop, register or memory). In such a case the SET is no longer temporary, but permanently changes information stored in the affected component (Single Event Upset) [5], however restoring information to the previous state recovers normal operation of the circuit.

It should be mentioned that in the case of SRAM-based FPGAs SEUs can occur not only in the working registers and memory but also in the configuration memory [6]. The SEU in the configuration memory can change both the functionality of the FPGA cell and the programmable connections between them.

AUTOMATIC TMR GENERATION FOR FPGA BASED CIRCUITS

The control for XFEL will use sophisticated digital systems based on FPGAs and DSPs [7]. They are sensitive to SEUs and therefore require additional countermeasures to assure their uninterruptible and reliable operation in in radioactive environment. One of the method is to provide redundancy in hardware and software [8]. The TMR is commonly used for hardware. This method assumes that even in case of single error in one of the three identical modules it is possible to calculate the proper output signal. However the multiplication of the whole circuit means that resource utilization increases also 3 times and more. Therefore the most critical parts of the circuit should be identified and only these parts should be multiplicated in order to save the precious resources of FPGA (Fig. 1).

In order to identify them it is necessary to perform simulation of the circuit operation together with SEUs. The interactions between elementary particles and semiconductor crystal, that is a source of SEUs, have a random nature described by density of probability. The simulation of SEUs is usually performed on microscopic level with application of physical based, multi dimensional models of semiconductor device [5]. Such tools are very valuable when the device is optimized for radiation hardness in the factory but are useless for the designers of the systems based on FPGAs and microprocessors.

The method for simulation of the digital system together

* We acknowledge the support of Polish National Science Council Grant ”N515 07231/3756”)

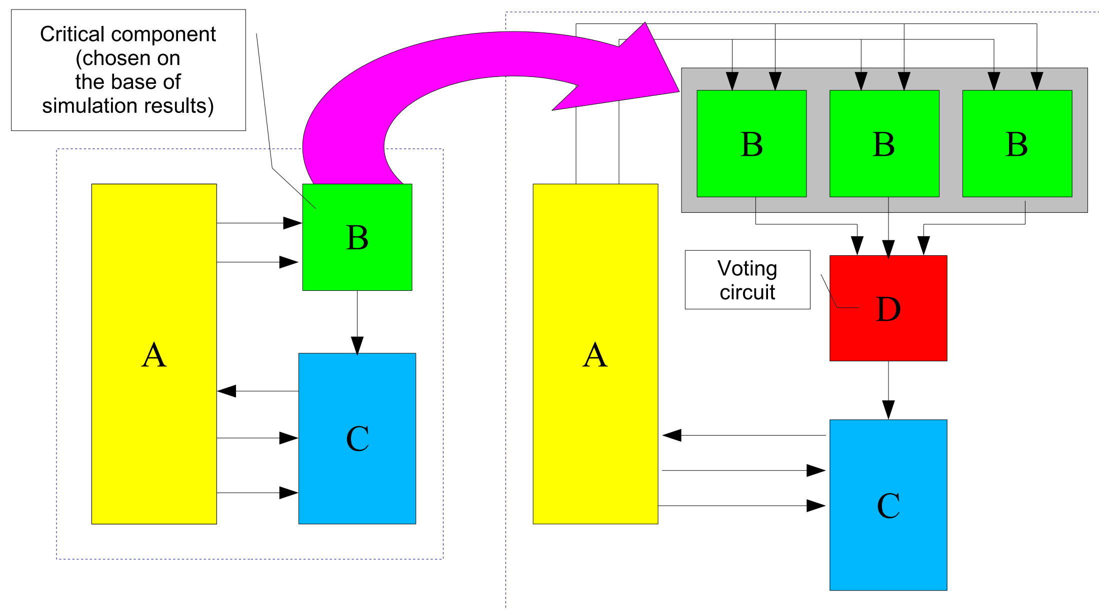

Figure 1: TMR method of FPGA based circuit improvement.

with SEU generation using common languages and simulation tools is described in [9]. It requires a tiny modification of the VHDL description and application of SEUSIM library including the necessary functions. The modification can be made automatically basing on the analysis of the source code. The special CAD environment was developed to facilitate the process. For syntactical analysis of VHDL source code the VHDL parser and analyzer *SAVANT* [10] is used. It parses the source file and creates the objects tree of VHDL primitives: statements, expressions, etc. All these information are translated into the AIRE (Advanced Intermediate Representation with Extensibility) format and exported as an *xml* file. Finally, the HaOS (Hardware Operating System) computer program analyses this data identifying the flip-flops signals and make modifications to their assignment statements causing SEUs to be randomly generated and effecting the information stored in flip-flops. The modified source code can be compiled and simulated by commonly used VHDL simulators (ModelSim, Active-HDL, gHDL).

The designer decides which parts of the whole circuit are the most critical basing on simulation results. The HaOS program supports the designer with special GUI with the tree view of the circuit that allows to chose and mark the parts for TMR multiplication (Fig. 2). These parts (components, processes, signals) are triplicated automatically and all necessary additional logic (voting circuits) are created during the final code generation. That means that designer can develop the circuit without thinking about TMR and finally when the circuit has all functionality implemented and debugged the TMR can be automatically generated and added. Current version of the HaOS program creates only the spatial redundancy. Future plans are to include also the temporal redundancy. It will allow to choose between slower but occupying less surface time redundancy and faster but more expensive in the term o resources spatial TMR.

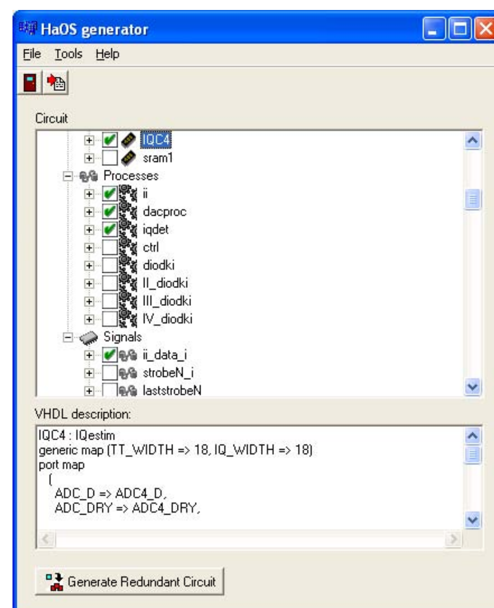

Figure 2: Program window allowing to chose and mark the components of the circuit for TMR multiplication.

IMPLEMENTATION OF LLRF CONTROLLER

In order to check the efficiency of the proposed method and tools the LLRF (Low Level Radio Frequency) controller [11] was implemented without and with TMR and the results of both implementations were compared. The

controller is a complex digital design. It consist of several modules: global control, clock generation, timing, data acquisition system and computational pipe. Additional modules provide interfaces through optical transmission and VME bus.

The implementation of the controller in Virtex 2 Pro FPGA chip without any TMR gives the results collected in Table 1. It is clearly visible that simple triplication of the whole controller will occupy more resources than it is available in single FPGA chip (utilization over 100%). The most critical (and most complex too) module in the controller is computational pipe.

The computational pipe block was marked for TMR multiplication by designer. After pressing the button “Generate Redundant Circuit” the program instantiated the 3 identical copies of computational pipe block. Then all the outputs from this blocks were connected automatically to the voting circuits. The outputs of the voting circuit were attached to the places where outputs from computational pipe block were connected before. The voting circuits were also added automatically. In order to prevent optimizer from combining together the triplicated modules one of the input signals was also triplicated (“start processing” signal was choosen for that) and gated with three independent external logical signals (called TMR1, TMR2, TMR3). During circuit operation all this three TMRx signals should be set to ‘1’ making the gates transparent. However during implementation the hardware compiler must treat this signal separately and that prevents from optimizing out the redundant modules.

The resource consumption for implementation of the controller with TMR is collected in Table 2. The computation block is really triplicated (the number of multipliers increased 3 times) but other resources increase is not in scale. Thanks to that it was possible to still fit the design in single FPGA chip.

Table 1: Device utilization summary (2vp30ff1152-6) for controller without TMR

Resource	**used**	**out of**	**percent**
Slices	5288	13696	38
Slice Flip Flops	5366	27392	19
4 input LUTs	8307	27392	30
BRAMs	84	136	61
MULT18X18s	45	136	33

CONCLUSION

The paper presents the CAD environment for automatic implementation of TMR in FPGA circuits. The TMR implementation of the LLRF controller allows to control the RF field in the cavities even if one copy of computational pipe does not work or work erroneously and even when the external interfaces do not work. The HaOS environment is an easy to use tool even for not experienced designers.

Table 2: Device utilization summary (2vp30ff1152-6) for controller with TMR in computation pipe

Resource	**used**	**out of**	**percent**
Slices	10410	13696	76
Slice Flip Flops	10148	27392	37
4 input LUTs	18520	27392	67
BRAMs	92	136	67
MULT18X18s	135	136	99

It allows to add TMR to the ready to use and debugged project in a simple way while still giving the designer the full control what parts should be multiplicated.

REFERENCES

[1] DESY XFEL Project Group, European XFEL Project Team 2006 *The European X-Ray Free-Electron Laser Technical Design Report* (Deutsches Elektronen-Synchrotron DESY, pp.71-74)

[2] Nguyen D.N., Scheick L.Z. 2002 *SEE and TID of emerging non-volatile memories* (Radiation Effects Data Workshop, 2002 IEEE, pages 6266)

[3] Baumann R.C., Smith, E.B. 2000 *Neutron-induced boron fission as a major source of soft errors indeep submicron SRAM devices*(Proc. 38th Int. Reliability Phys. Symp. 2000, pp.152-157)

[4] Alexandrescu D., Anghel L., Nicolaidis M. 2002 *New Methods for Evaluating the Impact of Single Event Transients in VDSM ICs* (Proc. 17th IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems DFT02, pp.99-107)

[5] Dodd P. E., Massengill W. L. 2003 *Basic Mechanisms and Modelling of Single-Event Upset in Digital Microelectronics* (IEEE Trans. on Nuclear Science, Vol. 50, No.3, June 2003, pp. 583–602)

[6] Crain S.H., Mazur J.E., Katz R.B., Koga R., Looper M.D., Lorentzen K.R. 2001 *Analog and Digital Single-Event Effects Experiments in Space*(Trans. Nulc. Sci. Vol. 48, No. 6, pp. 1941-1848)

[7] Giergusiewicz W. et all, 2005 *Low latency control board for LLRF system: SIMCON 3.1* (Proc. of SPIE Volume 5948, Photonics Applications in Industry and Research IV, pp.710-715)

[8] Hentschke R., Marques F., Lima F., Carro L., Susin A., Reis R. 2002 *Analyzing Area and Performance Penalty of Protecting Different Digital Modules with Hamming Code and Triple Modular Redundancy* (Proc. 15th Symp. on ICs and Systems Design, SBCCI02, pp.95-100)

[9] Grecki M. 2006 *VHDL Simulation considering Single Event Upsets (SEUs)* (Proc. of NSTI Nanotech 2006, Vol. 1, pp. 717-720)

[10] http://www.cliftonlabs.com/vhdl/savant.html

[11] Simrock S.N., Cichalewski W., Grecki M.K., Jablonski G.W., Jalmuzna W.J. 2006 *Universal Controller for Digital RF Control*(Proc. of EPAC 2006, pp.1459-1461)