Fault tolerance techniques for FPGA-based implementations

**Master Student Projects, start date: As soon as Possible**

**Initial goal:** Applying Triple Modular Redundancy (TMR) for an FPGA-based design at netlist-level. The goal of this assignment is to automatically apply TMR for a given design using the net-list generated by synthesis tool and compared the vulnerability of the TMRed design with plain design.

Students are engaged with HDL programming, Learning FPGA characterization, tools for FPGA-based design synthesis, implementation and techniques for protecting a design implemented on FPGA against faults.

**Project Description:**

FPGAs are used widely in reliability-critical systems such as avionic and space systems. SRAM-based FPGAs are very sensitive to Single event upsets (SEUs). A single event upset is a change in memory cell state of a circuit, caused by energetic particles such as galactic cosmic rays, cosmic solar particles and trapped protons in radiation belt. When ionized particle hits the silicon devices, electric charge is deposited which may invert the transistors state or produce temporary voltage spike. Thus, the correct operation of the circuit is affected adversely. Thus, protecting the circuit implemented on FPGA against transient faults is essential for using them in avionic and space systems. This project is actually inspired from a space-based project run by a team of researchers in the Netherlands developing a digital receiver system for future space-based mission aiming for radio detections in deep space. You will find more information on NCLE in the following address: ([NCLE Moon Mission](#))

![TMR](#)  
*Figure 1 TMR a) for combinational logic and b) sequential logic*

The most popular fault tolerance technique widely used in space-based FPGA design is called Triple Modular Redundancy (TMR). TMR architecture is used for both detection and correction of faults occurring in the design. TMR uses three redundant, either combinational or sequential logics, whose outputs are voted by a majority comparator in order to determine the correct result. In a full TMR or XTMR, the whole logic is replaced three times, including combinational and sequential logics. Obviously TMR introduces large overhead in the design, e.g. area and power increase by a factor of 3, which is significant overhead that not all the designs can cope with it. In this work, we therefore are interested to
develop a framework for applying TMR for only a subset of the design components vulnerable to single even upsets. Toward this, for Xilinx FPGA design applications “RapidSmith2” tool is supposed to be used.

RapidSmith2, the Vivado successor to RapidSmith is a research-based, open source FPGA CAD tool written in Java for modern Xilinx FPGAs. Its objective is to serve as a rapid prototyping platform for research ideas and algorithms relating to low level FPGA CAD tools. Specifically, it allows users to manipulate Xilinx designs outside the standard Xilinx-provided CAD tool flow.

For a given design, selective TMR design which is generated by this M.Sc. project is required compare with XTMR version of the design in the terms of accuracy and area-overhead. This will end with an efficient and reliable design for the use of a given design in future space missions.

Project Organization

This project will be performed in close cooperation with the Radboud-Radio Lab in Radboud University Nijmegen and Electronic Systems group at Eindhoven University of Technology.

Requirements

we are looking for someone who has the following profile:

- Programming of FPGAs
- C and/or Java programming
- Experience with Xilinx Vivado Design Suite

Please send your application including Curriculum vitae and your availability (starting date) to Mahsa Mousavi (m.mousavi@tue.nl).