The Cost of Implementing TMR in an FPGA-Based CPU

Ricardo Jasinski, Federal Center of Technological Education of Parana

Triple modular redundancy design techniques add reliability to a system at the expense of extra hardware resources. An intuitive conclusion is that the resulting design would require approximately three times as many resources as the original design; however, due to limitations specific to the logic fabric of traditional FPGAs and conventional design techniques, practical results tend to present a much greater ratio.

The implementation of a TMR version for an existent CPU (the Xilinx PicoBlaze [1]) is described in this paper. From its source code described in VHDL, a new functionally identical CPU is created, with each building block protected by means of TMR. The use of this approach instead of simply voting the CPU's outputs provides a more efficient and adequate degree of protection, due to the granularity of the error-correcting structures [2].

A novel method for the estimation of logic resources usage is proposed, which allows the prediction of the logic resources utilization for the resulting CPU. This method, besides providing a means for the choice of a specific device, can be used as a tool during the implementation of the TMR; by comparing the expected resource utilization with the results produced by the synthesis tool, one can detect design errors or the unintentional removal of redundant logic.

Special emphasis is given to the reuse of existing code. Whenever possible, the original source files are reused within the mitigated blocks. Changes, when necessary, are introduced in a precise and systematic manner.

The adaptation process begins necessarily with the identification of the CPU's building blocks, and their classification according to the type of logic implemented (combinational, sequential, or memory). Each block is protected by the adequate TMR technique, as described in [3]. For each module, the area overhead is presented, and compared against the expected value.

The TMR version of the CPU has been verified by simulation, followed by actual implementation and testing in commercial FPGAs. The timing penalty for the implementation of the TMR is traced to the delay introduced by the voting circuits, plus the increased routing complexity of the redundant structures within the FPGA. Register elements utilization is found to be 3.1 times that of the original CPU, and the logic elements usage is increased by a factor of 4.6.

References:

[1] Xilinx, Inc. PicoBlaze 8-bit Microcontroller for CPLD Devices. Xilinx Application Note 387, v1.1. Jan, 2003. Available at: www.xilinx.com.

[2] Lyons, R. E., and Vanderkulk, W. The Use of Triple-Modular Redundancy to Improve Computer Reliability. IBM Journal of Research and Development. New York, 6(3): 200-209, Apr. 1962.

[3] Xilinx, Inc. Triple Module Redundancy Design Techniques for Virtex Series FPGA. Xilinx Application Note 197, v1.0. Aut.: Carl Carmichael. Nov, 2001. Available at: www.xilinx.com.

Author Contact Information:

Ricardo Jasinski
Federal Center of Technological Education of Parana
Curitiba, PR, Brazil
jasinski@cefetpr.br
Tel.: +55 41 310 4756
Fax: +55 41 310 4753