Fault Tolerant Computation and VLSI Testing

Individual Project: Self-Purging Redundancy on a MIPS ALU Unit

1 Introduction

Fault tolerant circuit design has become increasingly important over the past several decades, due to the demand for faster and smaller systems. A common method for making circuits tolerate one of the modules developing a fault is through redundancy: having multiple instances of the same module and taking the vote as the majority verdict. These systems would work until the majority of modules developed faults. A system with N modules is an N Module Redundancy (NMR) system.

Following on from NMR, research then investigate hybrid redundancy techniques. These techniques often involve more complex logic to ensure the system remains running even after more than the majority of active modules develop faults. Examples of this include NMR-with-Spare schemes, and Self-Purging Redundancy [1]. The latter scheme will be the focus of this report.

We investigate Self-Purging Redundancy in practice by implementing it on the ALU component of a MIPS processor. We show how the reundancy logic was implemented for our n-bit ALU and in Verilog. Mathematical models for reliability are used to measure how the reliability changes depending on the number of modules and the voter threashold. The circuit was synthesised by the Synopsys Design Compiler and metrics for area, timing and power were taken for different numbers of modules and voter threasholds. From these results, we reach a compromise between reliability and overhead and compare with the original circuit and NMR circuits with the same number of modules. The source code for the final self-purging redundancy scheme is included as part of this submission.

Also included in this report is the scripts used to generate the metrics for our circuits.

References

[1] J. Losq. "A Highly Efficient Redundancy Scheme: Self-Purging Redundancy". In: *Computers, IEEE Transactions on* C-25.6 (June 1976), pp. 569–578. ISSN: 0018-9340. DOI: 10.1109/TC.1976.1674656.