Balancing Microprocessor Reliability and Power Usage using Spintronics, FPGAs and Redundancy

Michael Burton
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL 32816-2362

***Abstract*— This paper seeks to find a way to balance microprocessor reliability and power consumption. Rogue radiation can affect logical and memory operations in a microprocessor and create soft errors. Some methods for mitigation can increase reliability by 100 to 100,000 fold, but at the cost of doubling the energy use. Some of these methods involve duplicating hardware or reusing hardware to verify results of an operation. These come at the cost of increasing energy consumption for each operation. After creating and analyzing a program using four different methods, the design that yields the least amount of energy use is [3].**

Keywords—MIPS, Spintronic, Fault resilience, Non-volatility, Soft error, Registers, Redundancy, Logic design, Microprocessor chip

I. Project Design

The test program by design will read an input from the users of type string or character array. The final output will be the number of times that the entered string was found in the comparison string already in the program, disregarding the case of each character. To achieve this, the program first stores the input string in a MIPS register. It will iterate over each of the characters one by one until the null terminator. After reading each character, it will convert it to the same case as the already stored comparison string, then store it in its own register. Next the program will read the comparison string one character at a time until the null terminator. Each character will be compared to the registers populated by the input in the order they were populated. If the next register is not equal it will move to the next comparison character, and if the next register is null, it has reached the end of the input string and all characters have matched. When a null register is hit, the program will add one to the match counter. After the comparison string is finished, the program will output the input string and the amount of times it appears in the comparison string. These events are outlines in the flowchart in Fig.1. This program was written in this way to minimize the number of branching statements and reading commands to reduce the energy use. The three test cases showing in Fig.2 show the tests used to verify the program's logic. The three variations are the word "knight" with all upper, all lower and mix case.

This will test the programs ability to deal with both cases while comparing the two strings. The outputs have been verified

Fig.1: Flowchart of the assembly program.

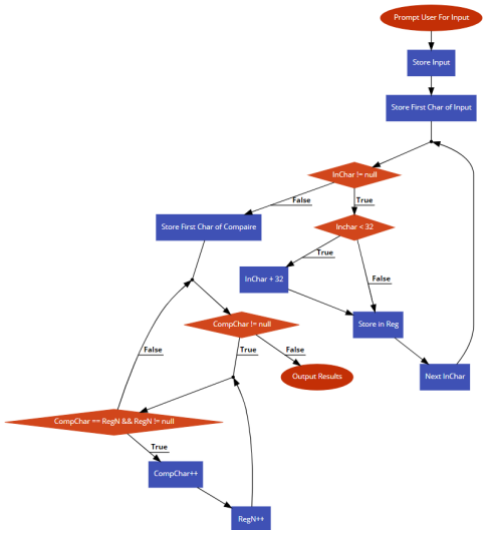

Fig.2: Sample outputs of the program.

1.
```
Please input first word:knight
knight
6
-- program is finished running --
```

2.
```
Please input first word:KNIGHT
KNIGHT
6
-- program is finished running --
```

3.
```
Please input first word:KnIgHt
KnIgHt
6
-- program is finished running --
```

by hand analysis to be correct. The output is meant to show that the string was stored correctly and the total number of matches.

II. Memory Bit-Cells

As CMOS technology approaches smaller half-pitches, they become more error prone due to radiation effects on their components. Rogue radiation particles can affect memory and logic operations. These effects can be mitigated by using spatial (redundant hardware) or temporal (reuse of hardware multiple times) methods. Triple Modular Redundancy (TMR) is a spatial redundancy. TMR is the method of creating copies of the logic operation to preform the same tasks simultaneously then comparing them in a majority gate [5]. If one of the redundant operations is affected by a radiation particle, the other two instances will overrule it in the majority gate. This can be done with N instances, usually an odd N to avoid ties [1].

The TMR drastically increases reliability by a factor of 100 to 100,000 fold [1], but at the cost of doubling the chip space and energy usage. While the increase in reliability of the logical operation covers the majority of the area, there is still a strong reliance on the majority gate (or voter) to be accurate. There are other methods to increase reliability including the use of field programmable gate arrays (FPGAs) [6][7] and spintronics [2-4]

III. Results and Discussion

Table 1 displays the energy used for a read operation in the designs discussed in [2-4]. Each design has tradeoffs between reliability and energy cost. [1] is less energy efficient than [2], but costs less to fabricate. Each design should be chosen for its application. Most fault tolerant systems are used in mission critical tasks such as aerospace, deep space, and other applications that need fast, accurate calculations at the cost of more energy usage.

Table I: Energy consumption for a single bit-cell read operation in the designs provided in [2-4].

Design	Energy Consumption For Each Bit-cell's Read Operation
FTNV-L [2]	1.17 pJ
3MR [2]	0.366 pJ
[3]	0.15 pJ
[4]	0.8 pJ

Table II: Total Energy consumption for the assembly program using designs provided in [2-4].

Design	Total Energy Consumption
FTNV-L [2]	748,821 pJ
3MR [2]	238,281 pJ
[3]	101,121 pJ
[4]	513,871 pJ

Table 2 is the total energy used to complete the program described in the project description. It includes ALU, branch, jump, and other instructions at a constant rate, varying for only the memory read instructions based on table 1.

IV. Conclusion

In mission critical computing tasks that are susceptible to rogue radiation waves, that can effect logical and memory operations of a processor, reliability is key. As components get smaller and use less voltage these problems get worse. These can be mitigated by adding spatial or temporal redundancy. These methods duplicate or reuse hardware to then send all of the outputs to a majority gate (or voter). This voter will ensure that a single operation cannot be affected by radiation because the likelihood of multiple logical components to be affected in the same way is extremely small. Memory and the voter still remain vulnerable to radiation. The voter can become more robust by using either FPGAs or Spintronics. After surveying these articles, design [3] produces the most energy efficient design.

References

[1] F. S. Alghareb, R. A. Ashraf and R. F. DeMara, "Designing and Evaluating Redundancy-based Soft Error Masking on a Continuum of Energy versus Robustness," in IEEE Transactions on Sustainable Computing, vol. PP, no. 99, pp. 1-1.

[2] R. Bishnoi, F. Oboril and M. B. Tahoori, "Fault Tolerant Non-Volatile spintronic flip-flop," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2016, pp. 261-264.

[3] J. Lopes, G. D. Pendina, E. Zianbetov, E. Beigne and L. Torres, "Radiative Effects on MRAM-Based Non-Volatile Elementary Structures," 2015 IEEE Computer Society Annual Symposium on VLSI, Montpellier, 2015, pp. 321-326.

[4] Y. Lakys, W. S. Zhao, J. O. Klein and C. Chappert, "Hardening Techniques for MRAM-Based Nonvolatile Latches and Logic," in *IEEE Transactions on Nuclear Science*, vol. 59, no. 4, pp. 1136-1141, Aug. 2012.

[5] J. F. Wakerly, "Microcomputer reliability improvement using triple-modular redundancy," in *Proceedings of the IEEE*, vol. 64, no. 6, pp. 889-895, June 1976.

[6] DeMara, R. F. (2017). Adaptive Resilience Approaches for FPGA Fabrics.

[7] A. Ramos, A. Ullah, P. Reviriego and J. A. Maestro, "Efficient Protection of the Register File in Soft-Processors Implemented on Xilinx FPGAs," in *IEEE Transactions on Computers*, vol. 67, no. 2, pp. 299-304, Feb. 1 2018.