Energy Consumption of Computation and Reliability Issues Concerning Triple Module Redundancy, and Different Circuit Designs

Jordi Niebla
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL 32816-2362

Abstract*— In this paper we will be discussing Triple Module Redundancy and how other methods can be used to implement a more reliable design. In this report we will also observe a program whose objective is to read a user input string and output the number of times the user selected word has appeared in the input. The index of where the word is should also be shown in the output. We will also analyze different circuit designs and compare their energy consumptions. From our analyzation, we can see that the "DNU Latch" consumed the least amount of energy, therefore, is the better design for this program.

***Keywords*— Fault-Tolerant Voter, Reliability, TMR, LCDMA, NMR system, Defects, Energy Consumption, Assembly Language**

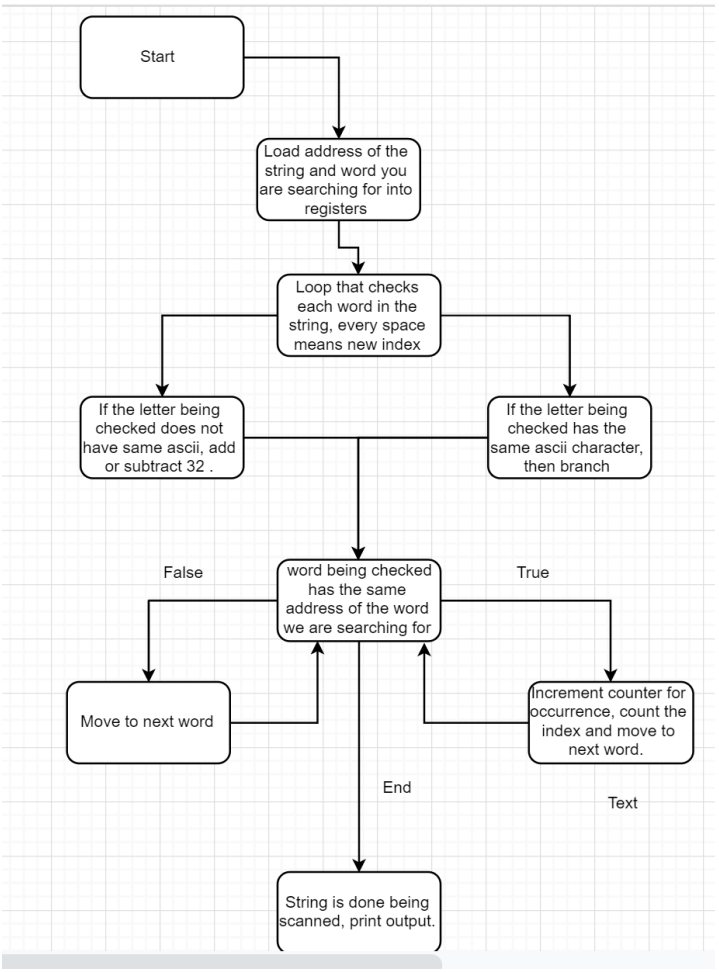
I. Introduction

A. Project Design

To begin the implementation of this program, we must load the prompts we want to ask the user to input. We can do this by a simple load immediate instruction, using the syscall 4, to print the string. After we load our inputs, we can start implementing the logic for the program. First, we want to load the address of the word we are searching for into a new register. Next, we implement a loop that begins to search each word in the string. We can do this by using the load byte instruction to check each word in the string. We can then use branch statements to check if the characters in the input are upper and lower case. We can compare the address of the input and the word search to see if they have the same value of ascii, if not then we can add or subtract 32. Now the program can go through each word in the string, and by implementing a loop with a counter, we can jump to it every time the address of the word searched appears.

To implement the indexes, the logic is similar, but we will be using the ascii value of 32 (ascii value for space) to indicate us every time the index has increased. As we are searching for the word along each word of the string, at the end of each loop we will jump to the index counter label, which has a set of loops that runs to check what the index is when the word is found.

Fig.1: Flowchart of the assembly program



B. *Test Cases*

To test the program, I will be using 3 different user input statements. The first one will focus on the use of upper- and lower-case letters. The second one will be a simple short sentence. And the last input will be a blank statement, so that we can see what happens to the program when nothing is inputted.

1) “TeStiNg TESTing testing TESTING.” Word search: “Testing”

2) “Living in Florida is amazing. The beautiful weather makes it easy to get up and get started with my day.” Word search: “day”

3) (Blank statement)

Fig.2: Sample outputs of the program.

```
# Number of times the word "testing" was found in the input stri

# Indexes of the matches found: 1, 2, 3, 4.
```

```
# Number of times the word "day" was found in the input strir

# Indexes of the matches found: 20.
```

```
# Number of times the word "" was found in the input str

# Indexes of the matches found:
```

II. Memory Bit-Cells

Triple Module Redundancy, also known as TMR, is a fault tolerance technique which has three identical latches and a voter circuit. It is set up so that the output returns the correct result even if failure occurs to one of the latches [1]. Most forms of defect tolerance techniques like TMR, are based on strategies that rely on a majority voting. Since the voter plays a big role in TMR systems, it is important that the voter is reliable. If the voter fails, regardless whether the other modules fail or not, the TMR system will fail [5].

Although redundancy is a useful in some case, it is not always the solution. Sometimes too much or too little redundancy can cause less reliability. The challenge comes from attempting to find the right amount of redundancy needed to achieve optimal reliability [4]. Some tradeoffs for the TMR can include, that the design implemented must include 3n devices, instead of 3 devices, while also needing a majority gate [4]. TMR also includes significant area and power consumption [1].

As future technology nodes develop, it is expected that there will be an increase in the number of defects. Therefore; TMR based on fault-tolerant voters can be less reliable if the voter circuit is not redesigned [6]. Other methods to make sure that data will transfer with low error rates are being implemented. One method can include a voter less fault-tolerant strategy to implement a robust NMR system design. Using a LCDMA (Logic Code Division Multiple Access), the author was able to eliminate the need for a voter unit [2].

III. Results and Discussion

Using the second test case string, the calculation for the assembly program will be determined. Below we have how many fJ will be used per instruction. The total fJ is 4742 + 442*(memory fJ). This was calculated by adding the sum of the instructions found in the MIPS Instructions Statistics. From the results of Table II, it is seen that the DNU latch consumed the least amount of energy.

1) *ALU = 1 fJ*
2) *Branch = 3 fJ*
3) *Jump = 2 fJ*
4) *Memory = Refer to Table I*
5) *Other = 5 fJ*

Table I: Energy consumption for a single bit-cell memory in the designs provided in [1-3].

Design	Energy consumption of a Single Bit-Cell Memory
SEU-Latch [1]	0.88 fJ
DNU-Latch [1]	0.28 fJ
[2]	6.96 fJ
[3]	1.51 fJ

Table II: Total Energy consumption for the assembly program using designs provided in [1-3].

Design	Total Energy Consumption
SEU-Latch [1]	5130.96 fJ
DNU-Latch [1]	4865.76 fJ
NRAM [2]	7818.32 fJ
Double Node SEU Tolerant-Latch [3]	5409.42 fJ

IV. Conclusion

From our observations of the energy consumption calculations, we can see that the DNU latch consumed the least amount of energy. We can conclude that the DNU latch was the best design for this program. The SEU Latch was not far behind, and neither was the DN SEU Tolerant Latch. The NRAM's energy consumption seemed shockingly high, but this is due to its high energy consumption from the memory instruction. Overall, a deeper understanding in TMR was developed. Also, different designs being implemented to increase reliability such as: SEU, DNU, NRAM, and Double Node SEU Tolerant-Latch. Skills in assembly language programming were also gained in the making of this report.

References

[1] F. S. Alghareb, R. Zand and R. F. Demara, "Non-Volatile Spintronic Flip-Flop Design for Energy-Efficient SEU and DNU Resilience," in IEEE Transactions on Magnetics, vol. 55, no. 3, pp. 1-11, March 2019, Art no. 3400611.

[2] H. Pourmeidani and M. Habibi, "Hierarchical defect tolerance technique for NRAM repairing with range matching CAM," 2013 21st Iranian Conference on Electrical Engineering (ICEE), Mashhad, 2013, pp. 1-6.

[3] K. Katsarou and Y. Tsiatouhas, "Double node charge sharing SEU tolerant latch design," 2014 IEEE 20th International On-Line Testing Symposium (IOLTS), Platja d'Aro, Girona, 2014, pp. 122-127.

[4] D. Bhaduri and S. K. Shukla, "Nanoprism," Proceedins of the 14th ACM Great Lakes symposium on VLSI - GLSVLSI 04, 2004.

[5] T. Ban and L. A. D. B. Naviner, "A simple fault-tolerant digital voter circuit in TMR nanoarchitectures," Proceedings of the 8th IEEE International NEWCAS Conference 2010, pp. 269–272, 2010.

[6] R. Kshirsagar and R. Patrikar, "Design of a novel fault-tolerant voter circuit for TMR implementation to improve reliability in digital circuits," Microelectronics Reliability, vol. 49, no. 12, pp. 1573–1577, 2009.