Energy consumption enhancements using Full Adder based circuits

Gean Morales
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL 32816-2362

Abstract—**With power being a major improvement in realizing adders for SRAM- Field Programmable Gate Arrays (FPGA). In this paper, a summarization of the full adder structure circuit and comparisons to two-input Boolean gates is discussed briefly as well as energy consumption enhancements from Full adder-based circuits. With energy being a topic in matter, two energy consumption tables included in the second page display energy consumptions for four designs for single ALU instructions and the total energy consumed for the assembly program. With heat dissipation and energy consumption being the main factors, where even up to a hypothetical ten percent cut on both can have positive economic difference. As for this analysis, a Full-Adders uses Boolean logic-based circuit provides a reliable solution to this issue and overall beneficial results. MUX (multiplexer) for ADCs (Analog to Digital Converter) within FPGAs (Field Programmable Gate Arrays) provide considerable power improvements.**

Keywords—ALU, FPGA, Full adder, SRAM, ADC, MUX, NVFA, MTJ

I. Introduction

The core objective of the MIPS code provided was to search for a string that varied in length and find keywords provided and chosen by the user. After that the program would display the number of occurrences as well as the index of where the word was found. Each letter on the string is treated as an element int the array. The program compares both arrays, the word array and the sentence array for similarities.

The sections below will explain how the code is working once it has all been developed and executed. The first will describe the code project design using a flowchart the next will describe the test case.

A. Project Design

The code is fairly straightforward a string stored in .data is used to count how many times a keyword, chosen by the user (e.g. Knight or Ucf, etc), occurs. One loop loads the bytes from keyword and string, and then compare current index from keyword and string. The string index is modified to compare for lowercase and uppercase scenarios in the keyword. If there is no match in indexes, the keyword is reset. If every index in the keyword matches the string index, the counter is updated, and the keyword is reset again until the loop reaches the NULL terminator in string. Fig.1 shows the basic idea of how the assembly code works.

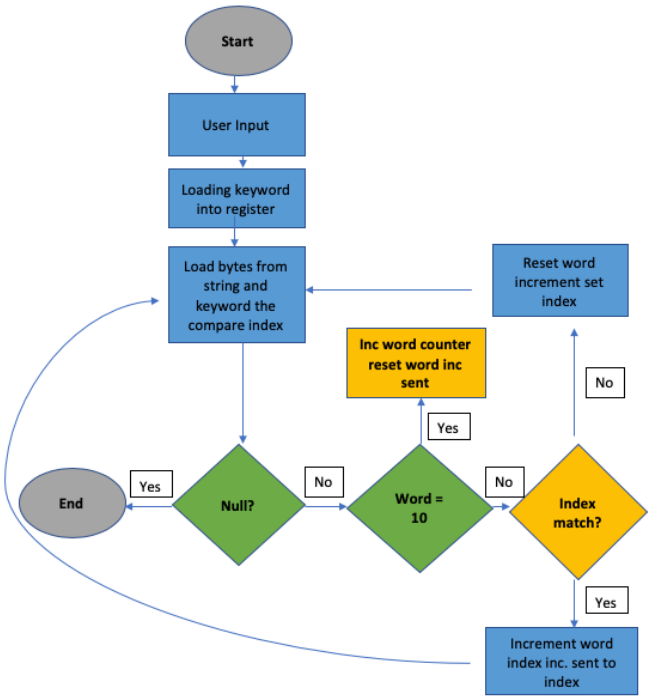

Fig.1: Flowchart of the assembly program.

```
# Please type in a word (up to 10 characters) that you are looking for (e

KnIgHt

# Number of times the word "KnIgHt" was found in the input statement: 6

# Indexes of the matches found: 2, 32, 42, 53, 85, 97.
-- program is finished running --

# Please type in a word (up to 10 characters) that you are looking for

UCF

# Number of times the word "UCF" was found in the input statement: 2

# Indexes of the matches found: 9, 120.
-- program is finished running --

# Please type in a word (up to 10 characters) that you are looking for (e.

tuition

# Number of times the word "tuition" was found in the input statement: 1

# Indexes of the matches found: 19.
-- program is finished running --
```

Fig.2: Sample outputs of the assembly program.

B. Test Cases

In Fig.2 the three sample inputs chosen were: knigHt, UCF, and tuition. A combination of uppercase and lowercase letters were included in the keywords to show that the program compares for both cases. The results are shown in Fig.2, along with the keyword chosen by the user.

II. Full-Adder Circuit

The contents of the full adder consist of logic gates implemented together to create a what is called a full adder. A full adder is designed to receive three one-bit binary numbers. Which are the two numbers that operations are to be carried out on, with the following being its carry bit. These are regularly referred to as A, B, and Carry in. Once the full adder has received the information it performs a calculation and outputs only two different pieces of information being the sum of the operands and the carry out bit.

A full adder can be implemented in many different ways such as with a custom transistor level circuit or composed of other gates, transistors may prove better energy enhancements according to J. Lin, Y. Hwang. A last OR gate before the complete yield might be supplanted by a XOR door without adjusting the subsequent logic. Using two types of gates can prove to be more convenient if the circuit is being implemented using simple integrated chips which typically contain only one gate type per chip according to M. Aguirre-Hernandez and M. Linares-Aranda. A full adder can also be constructed from two other half adders by two ports to the input of one half adder, then taking its sum-output *S* as one of the inputs to the second half adder and its other input, and finally the carry outputs from the two half-adders are connected to an OR gate. The final sum output is the sum-output from the second half adder of the full adder and the OR gate output is the final carry output.

In the articles data the utilization of the full adder is shown to have numerous advantages when put against other forms of computing approaches. Ronald F. DeMara showed that the use of the full adder had significant speed increases and substantially less energy consumption. His study proved ro achieve a 2-3.8 fold decrease in power dissipation and 2.3 and 1.13 fold decreased delay on 1GHz and 500MGhz operating speeds, which show great improvements. Similar results were described in the article “Domain wall motion based magnetic adder” where area efficiency is shown to be significantly increased, roughly 13 fold which is a drastical increase in energy consumption enhancements.

III. Results and Discussion

Calculation obtained through:

Energy = (ALU*Table1[])
+(Jump*3pJ)+(Branch*4pJ)+(Memory*100pJ)+(Other*5pJ)

Table I: Energy consumption for a single ALU Instruction in the designs provided in [1-3].

Design	Energy Consumption For Each ALU Instruction
[1]	0.6 pJ
[2]	6.3 pJ
RTM-based [3]	1.67 pJ
STT-based [3]	1.61 pJ

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

Design	Total Energy Consumption
[1]	4.1E-7 pJ
[2]	5.1E-7 pJ
RTM-based [3]	4.4E-7 pJ
STT-based [3]	4.3E-7 pJ

IV. Conclusion

Design 1 clearly provides the least energy consumption upon being used with the assembly program. Furthermore, a trend pertaining to the steady energy consumption of the assembly program with the ALU consumption per design, thus proving to be a linear relationship. This reveals that different implementation of a Full Adder can dramatically impact the net efficiency of a circuit. Drastical alterations shown in MFAs by mainly focusing in area efficiency, or the RTM based NVFA making overall improvements on both area and speed rate, and finally energy saving SLIM-ADC. This comes to show that implementing different ways of performing a specific task can greatly affect the surrounding variables while producing the same output.

References

[1] S. Salehi and R. F. DeMara, "SLIM-ADC: Spin-based Logic-In-Memory Analog to Digital Converter Leveraging SHE-enabled Domain Wall Motion Devices," Microelectronics Journal, Vol. 81, pp. 137–143, November 2018.

[2] H. P. Trinh, W. S. Zhao, J. O. Klein, Y. Zhang, D. Ravelsona, and C. Chappert, “Magnetic adder based on racetrack memory,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 60, no. 6, pp. 1469–1477, Jun. 2013.

[3] K. Huang, R. Zhao and Y. Lian, "A Low Power and High Sensing Margin Non-Volatile Full Adder Using Racetrack Memory," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 4, April 2015.

[4] A. M. Shams, T. K. Darwish and M. A. Bayoumi, "Performance analysis of low-power 1-bit CMOS full adder cells," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 1, pp. 20-29, Feb. 2002.

[5] J. Lin, Y. Hwang, M. Sheu and C. Ho, "A Novel High-Speed and Energy Efficient 10-Transistor Full Adder Design," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 5, May 2007.

[6] M. Aguirre-Hernandez and M. Linares-Aranda, "CMOS Full-Adders for Energy-Efficient Arithmetic Applications," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 4, pp. 718-721, April 2011.