

Analysis of Full Adder Circuit In a Program that Counts Occurrences of a User-Entered Word.

Zachary Chiapperini

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

Abstract— In this paper, the fundamental metrics of a Full Adder are examined in the context of a program that counts the occurrences of a user-entered word in a pre-determined paragraph. The program first prompts the user to enter a word. The program converts the word and the paragraph to all capital letters, so the program will search for occurrences of a word regardless of letter case. The user-entered word is then searched for in the paragraph. Upon completion of the search, the program outputs the word entered, followed by the number of occurrences of the word in the paragraph. For this project, 4 inputs: “knight”, “citronaut”, “ “, and “pineapple” are utilized to demonstrate how this program operates under different conditions. Next the Full Adder ALU component is examined and some of the approaches to reduce overall power consumption are discussed including approximate computation and spintronic computation. Finally, the total energy consumption is calculated for different ALU designs. For this program the most efficient ALU is design [3].

Keywords—Base address, register, ASCII, loop, jump, branch, Full Adder, Digital Signal Processing (DSP), Conventional Mirror Adder (CMA), Approximate Mirror Adder (AMA), Nonvolatile Memory (NVM), Spin Transfer Torque (STT), Magnetic Random-Access Memory (MRAM), Perpendicular Magnetic Anisotropy (PMA).

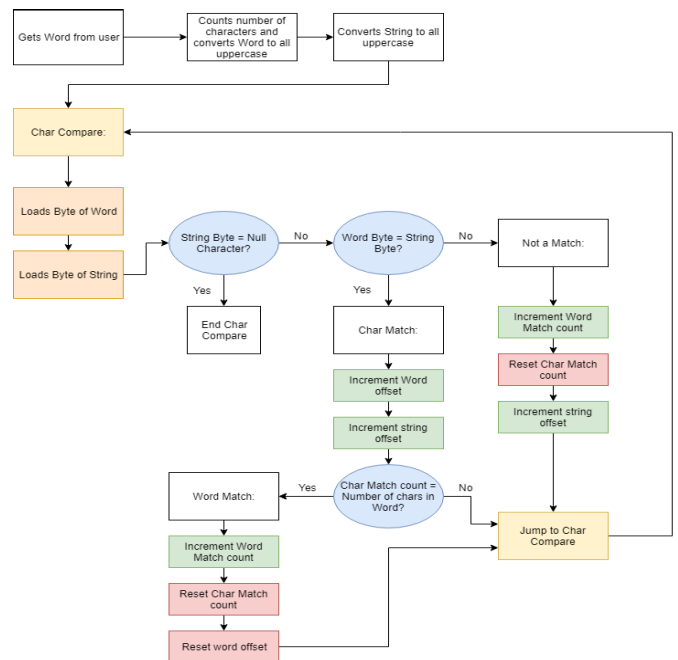
I. PROJECT DESIGN

The number of occurrences of the entered word in a paragraph is calculated using a loop. The CharCompare loop begins by loading the characters located at the base addresses of the word and paragraph into respective registers. First, if the null character is loaded from the paragraph then the CharCompare loop ends. Next, if the ASCII values of the two characters are equal then the loop branches to the CharMatch function. If the ASCII values are not equal, then the loop jumps to the NotMach function.

The CharMatch function will add one to the base addresses of the word and paragraph and increment a character match counter by one. When the character match counter equals the word length then the function branches to a WordMatch function. If they are not equal, the function jumps to the beginning of the CharCompare loop. The WordMatch function resets the word offset and the character match counter, increments a word counter by one, then jumps to the beginning of the CharCompare loop. The NotMatch function resets the

word offset and character match counter, then increments the base address of the paragraph by one and jumps to the beginning of the CharCompare loop.

I have chosen to include 4 unique inputs to illustrate the programs ability to effectively count the occurrences of an entered word in a paragraph. The first input “knight” demonstrates that the letter case of the entered word does not matter. The second input “citronaut” reaches the 9-character limit. The third input “ ” simply counts the spaces, which can be used to determine the number of words. The fourth input, “pineapple” reaches the character limit and yields 0 matches.



```

Mars Messages Run I/O
Please input first word: knight
KNIGHT: 6
-- program is finished running (dropped off bottom) --

Mars Messages Run I/O
Please input first word: citronaut
CITRONAUT: 1
-- program is finished running (dropped off bottom) --

Mars Messages Run I/O
Please input first word:
: 102
-- program is finished running (dropped off bottom) --

Mars Messages Run I/O
Please input first word: pineapple
PINEAPPLE: 0
-- program is finished running (dropped off bottom) --

```

II. FULL-ADDER CIRCUIT

The function of the Full Adder circuit, at its core, is to add two binary numbers. Addition is one of the most fundamental building blocks of arithmetic operations. All Full Adders have the same basic inputs and outputs. Single bit inputs A, B, and Cin are added to form the single bit outputs Sum and Cout (adding three single bits together yields maximum value of 3 which is expressible as a 2-bit binary integer). To add two n-bit binary numbers, use n Full Adders, where the Cout from the first bit is the Cin for the next, and so on. Arithmetic operations such as subtraction, multiplication, division, etc. can be achieved using adders [4]. Adders are also utilized in many applications that use Digital Signal Processing to output an image, sound, or video [2]. With misprocessing technology advancing faster than battery technology [4], it is crucial for engineers to develop and implement adders that consume less power to keep up with the increased usage of microelectronics. For this reason, adders remain to be an important domain of research [6].

Approximate computing can be used to reduce power consumption in applications which can tolerate some imprecision such as applications that use DSP blocks to output an image or video [2]. A Conventional Mirror Adder (CMA) consists of a total of 24 transistors. An Approximate Mirror Adder (AMA) is obtained by removing transistors one by one until the error constraints are breached, then reverting to the last modification [1]. Approximate computing can also be achieved through imprecise or approximate Full Adder cells with reduced logic complexity at the transistor level [2]. Another method of reducing power consumption is by using spintronic computing based on nonvolatile memory (NVM). Spintronics devices are based on the up or down spin of the electrons, and as a result are faster require less energy [5]. One of the most promising implementations of NVMs is the spin transfer torque (STT) magnetic random-access memory (MRAM) [3]. This design is based on perpendicular magnetic anisotropy (PMA) and offers power reduction on a smaller chip.

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

Design	Energy Consumption For Each ALU Instruction
[1]	5 fJ
CMA [2]	39 fJ
AMA [2]	12 fJ
[3]	1 fJ

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

Design	Total Energy Consumption
[1]	251.257 nJ
CMA [2]	251.420 nJ
AMA [2]	251.290 nJ
[3]	251.237 nJ

III. RESULTS AND DISCUSSION

The energy consumption for the program with the input “knight” is shown below using different ALU designs. Energy consumption for the program is calculated based on the following energy consumption values:

- 1) *ALU = Refer to Table I*
- 2) *Branch = 3 pJ*
- 3) *Jump = 2 pJ*
- 4) *Memory = 100 pJ*
- 5) *Other = 5 pJ*

IV. CONCLUSION

In conclusion, this paper illustrates the design approach of a word counter and examines the Full Adder component. One of the trends noticed during the programming portion of the project was the tendency for carriage return to be treated similarly to an ordinary character. When a word that is less than 9 characters is entered, the enter key is pressed to input the word. In these cases the carriage return was included as the last byte of the word. I found that both values (inclusive and exclusive of the carriage return) had their respective uses for offsetting the base address. Some of the topics covered in this paper include:

- 1) Algorithm design and optimization.
- 2) Character manipulation through adding or subtraction on ASCII value.
- 3) String navigation through altering the base address.
- 4) Components and different implementations of Full Adders.

5) Approximate computation through reducing number of transistors or the complexity of transistors.

6) Spintronic computation involving nonvolatile memory.

REFERENCES

- [1] A. A. Naseer, R. A. Ashraf, D. Dechev, and R. F. DeMara, "Designing energy-efficient approximate adders using parallel genetic algorithms," *SoutheastCon 2015*, Fort Lauderdale, FL, 2015, pp. 1-7.
- [2] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: imprecise adders for low-power approximate computing," *In Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design (ISLPED '11)*, Piscataway, NJ, USA, 409-414.
- [3] E. Deng, Y. Zhang, J. O. Klein, D. Ravelsona, C. Chappert and W. Zhao, "Low Power Magnetic Full-Adder Based on Spin Transfer Torque MRAM," in *IEEE Transactions on Magnetism*, vol. 49, no. 9, pp. 4982-4987, Sept. 2013.
- [4] Ahmed M. Shams, Tarek K. Darwish, and Magdy A "Performance Analysis of Low-Power 1-Bit CMOS Full Adder Cells," *IEEE Transactions on Very Large-Scale Integration (VLSI) systems*, vol. 10, no. 1, February 2002
- [5] Hao Meng, Jianguo Wang, and Jian-Ping Wang, "A Spintronics Full Adder for Magnetic CPU," *IEEE Electron Device Letters*, vol. 26, no. 6, June 2005.
- [6] Partha Bhattacharyya, Bijoy Kundu, Sovan Ghosh, Vinay Kumar, and Anup Dandapat "Performance Analysis of a Low-Power High-Speed Hybrid 1-bit Full Adder Circuit" *IEEE Transactions On Very Large-Scale Integration (VLSI) systems*, vol. 23, no. 10, October 2015