Analysis of a MIPS Assembly Program Total Energy Consumption and Full-Adder Circuit Designs

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

***Abstract*— In a world that relies on embedded devices, the demand is growing for faster technology. To achieve this, research must be conducted and applied to modern devices. Energy dissipation and speed are the two primary factors that are carefully analyzed among researchers and engineers. In this paper, we will be taking an analysis of a MIPS Assembly Program's total energy consumption and Full-Adder circuit designs to show how limiting the transistor count can increase speed but have effects on energy consumption. Through careful analysis of instruction statistics, Design 1 consumes the least amount of energy, while Design 3 consumes the most energy.**

Keywords—Floating Full-Adder, Transistor, ALU, Full-Adder, MIPS, Assembly, Static Energy, CMOS Full-Adder, Conventional Mirror Adder, Static Energy Recovery Full-Adder, Hybrid CMOS, Nobel Multiplexer Based Full-Adder.

I. Introduction

A. *Project Design*

The program begins by creating 2 strings, one for the hard-coded string where the user will be searching for matching words, and an allocated 10 byte long string used as the maximum size the user is allowed to search for. Next, the user is asked to input a string they wish to search for in the hard-coded statement. Strings to place a colon and new-line are also used. Next, the main function will ask the user to enter the string they want to search for. This must be a maximum of 10 bytes. Once the user presses enter, the program reads the length of both $t2 and $t3. The program then takes the base address of each letter of the inputted string and compares it to the address of each occurrence in the hard-coded string. Loops in the program then count these occurrences, looping until the hard-coded string is complete. The program also contains jump and link, and this is used to convert anything the user may enter into lowercase letters. The convert to lowercase loop uses the Ascii values of capital letters as the range, and then proceeds to subtract immediate '32' from them to convert to lowercase. Once the conversion is done, we jump register back and begin comparing occurrences. Using bridge not equal, the program goes through each instance and will jump if there isn't a comparison. Should there be an equal comparison, that number is incremented to the respective word. Since the letters are converted to lowercase, every instance should be

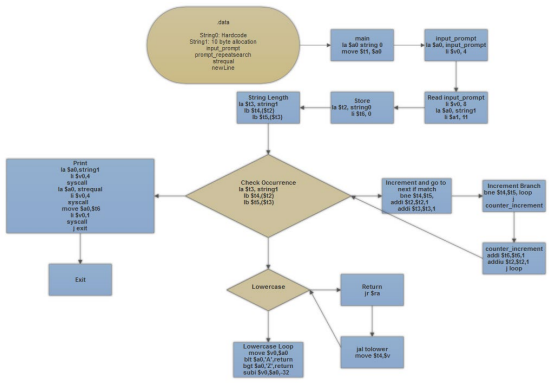

Fig.1: Flowchart of the assembly program.

```
Please input first word: knight
knight
: 6
Repeat Search? <Y/N>?
```

Output 1

```
Please input first word: ucf
ucf
: 3
Repeat Search? <Y/N>?
```

Output 2

```
Please input first word: university
university
: 2
Repeat Search? <Y/N>?
```

Output 3

Fig.2: Sample outputs of the assembly program.

accounted for. Finally, the program prints out the counter for the word the user entered. This leaves on the screen the proper number of occurrences. Refer to Figure 1 for detail in regard to the register names and branches.

B. *Test Cases*

Developing the test cases for this program involved analyzing a few things. The first thing I wanted to test was a word that occurred several times, in this case 6 times. This word was knight. By selecting this word, I can verify the program loops correctly several times and that the logic is proper. The next test case that I used was UCF. This test case was used because short words satisfy that the program correctly loops a minimum amount of times. The final test case chosen was university. By using a word that is the maximum amount of characters, the code correctly terminates properly and counts every instance once the program reads the inputted string.

II. Full-Adder Circuit

The primary purpose of a Full-Adder circuit is to add three one-bit numbers together, one being a carry value. The output sum results in two binary values, one being the carry value. Without a Full-Adder, arithmetic operations would be difficult and would make CPU's slower. A fast Full-Adder will use a minimal number of transistors to reduce calculation time but will consume more power in the process.

The Full-Adder Circuit stated in Reference [1] illustrate a design of a Conventional Mirror Adder. This type of Full-Adder uses 24 transistors and can achieves very low power consumption. While this circuit has very low power consumption, it is an approximation adder, and therefore not as accurate as other circuits. The Full-Adder Circuit in Reference [2] is a Hybrid CMOS type Full-Adder, with a minimal transistor count. It takes the ideas presented in Reference [1], and reduces the transistor count further, and using the same number of inputs and outputs. The Full-Adder Circuit demonstrated in Reference [3] is a Floating Full-Adder Circuit, using a minimal 8 transistors, but has much higher power consumption. This Full-Adder works by utilizing 5 inputs instead of the typical 3 for a normal Full-Adder. Analyzing another design with similar transistor count in Reference [5], possible research into Recovery Full-Adder circuits can introduce a method to recover Static Energy, and therefore reduce energy consumption. In Reference [4], the circuit is a Power-Aware Full-Adder, utilizing a higher count of 20 transistors, but instead achieves a lower power delay product using a new hybrid structure. No inputs that are inverted are required for this Full-Adder. Due to this, there is no ground output for the circuit.

In the future, we may see even more unique ways to design Full-Adder circuits. Some unique ways that have been researched include Reference [6], which shows a Nobel Multiplexer-Based Full-Adder, which uses 12 transistors and six identical multiplexers to conserve additional electrical charges that are generated, effectively reusing them. This leads to an increase in speed, and due to the multiplexer utilization, energy consumption is reduced. In Reference [7], a CMOS Full-Adder with 14 transistors is used. This is comparable to [6], which used ideas from a SERF (Static Energy Recovery Full-Adder). Compared with this Circuit, [7] uses a different approach, mainly because a CMOS Full-Adder has issues regarding threshold loss.

The Full-Adder circuits use various numbers of inputs and outputs to achieve what they are designed for. Every Full-Adder presented has a primary goal, and that is to reduce transistor count. By reducing transistor count, a Full-Adder circuit will run much faster, but will consume more energy. Full-Adder's analyzed in Reference [5]-[7] present more ways to conserve power consumption and give us a vision for the future.

III. Results and Discussion

The energy values provided in Table 1 below show the energy consumption of Design 1-4's ALU Instructions. Table 2 below shows the total Energy Consumption for the Assembly program, using the word "Knights" as the input string. Using the following energy values, the total energy can be calculated.

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

Table I: Energy consumption for a single ALU Instruction in the designs provided in [1-4].

Design	Energy Consumption For Each ALU Instruction
[1]	5 fJ
[2]	16.2 fJ
[3]	27.7 fJ
[4]	6.15 fJ

Table II: Total Energy consumption for the assembly program using designs provided in [1-4].

Design	Total Energy Consumption
[1]	191,124 fJ
[2]	239,698.4 fJ
[3]	289,573.9 fJ
[4]	196,111.55 fJ

IV. Conclusion

In conclusion, several Full-Adder circuit designs were investigated and compared to each other. Each design had a primary goal to reduce transistor count, to create a faster Full-Adder circuit. The designs that were compared for Energy Consumption showed that Design 1 had the least energy consumption of the 4 designs, while Design 3 had the most energy consumption. When comparing the 2, we can see that Design 1 had the most transistors, while Design 3 had the least. Takeaways from the analysis of the Full-Adder circuits and Assembly Code include:

- Lower transistor count will make a circuit run faster, however the energy consumption must be analyzed because it will be higher.
- Reducing the amount of memory instructions will have a far greater effect on energy consumption than any other instruction. (100 fJ compared with the highest ALU instruction count of 27.7 fJ). This is something to consider when writing Assembly Code.
- While lower transistor count generally will mean higher power consumption, various methods to reuse power or take static power have been studied, as seen in [5] and [7].
- Optimizing Assembly Code and getting instruction counts at their minimal can save time and money when it comes to designing new methods of creating ALU's.
- Designing the fastest circuit does not mean it is the best circuit for a project. Low power ALU's may be the optimal design, when speed isn't the priority.
- Algorithm analysis and creation in MIPS Assembly.

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] Hassoune, D. Flandre, I. O'Connor and J. Legat, "ULPFA: A New Efficient Design of a Power-Aware Full Adder," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 8, pp. 2066-2074, Aug. 2010.

[3] M. J. Zavarei, M. R. Baghbanmanesh, E. Kargaran, H. Nabovati and A. Golmakani, "Design of new full adder cell using hybrid-CMOS logic style," 2011 18th IEEE International Conference on Electronics, Circuits, and Systems, Beirut, 2011, pp. 451-454.

[4] N. Shiri Asmangerdi, J. Forounchi and K. Ghanbari, "A new 8-transistors floating full-adder circuit," 20th Iranian Conference on Electrical Engineering (ICEE2012), Tehran, 2012, pp. 194-197.

[5] Shalem, R & John, Eugene & John, Lizy. (1999). A novel low power energy recovery full adder cell. 380-383. 10.1109/GLSV.1999.757461.

[6] Yingtao Jiang, A. Al-Sheraidah, Yuke Wang, E. Sha and Jin-Gyun Chung, "A novel multiplexer-based low-power full adder," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 51, no. 7, pp. 345-348, July 2004.
doi: 10.1109/TCSII.2004.831429

[7] Vigneswaran, T & Mukundhan, B & Subbarami, P. (2018). A Novel Low Power, High Speed 14 Transistor CMOS Full Adder Cell with 50% Improvement in Threshold Loss Problem.