Comparing Energy Consumption of Look-Up Table Implemented Arithmetic Logic Units

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

***Abstract*—The energy efficiency that can be provided through Arithmetic Logic Unit design is seemingly infinite. Identifying areas that can offer the greatest percentage of improvement is important to continue towards a greener technology sector. An example program design that checks for the occurrences of words in a sentence is first described. This program is then processed by four separate ALU designs which utilize Look-up Tables and their energy consumption is then compared. The effect of modifying an ALU's design towards energy efficiency is shown by this comparison. Different ALU designs can be shown to give a decrease of nearly 1% in total energy consumption when applied to the same program design.**

***Keywords*— Look Up Table, Energy Consumption, Arithmetic Logic Unit**

I. PROJECT DESIGN

MIPS assembly language code was generated to count the number of occurrences of a word within a paragraph. The two inputs for this program were a paragraph, and the word to be searched for within that paragraph. The output was simply a count of the number of occurrences of that word. The word to be searched for would be an input accepted from the user, while the paragraph to search was stored in memory. Several other characters and strings were stored in memory to generate user input prompts and output. Several variables were stored in registers.

The main overview of the program is that characters that could be considered troublesome for processing are first removed. These characters included newlines, period, commas and apostrophes. A space was simply put in place of any occurrence of these characters. The next portion of the program capitalized all letters in both the input and the paragraph. This simplified the search for the occurrences of the word. Finally, the paragraph is then scanned and the occurrences of the word are counted and stored in a register. Lastly, the output is produced and shown to the user. The flowchart to the right shows how the the program was designed to scan the paragraph for occurrences of a certain word.

Thorough testing was completed for this program to ensure correct word counts were being found. Three separate words were entered as inputs for the same paragraph, and the case of

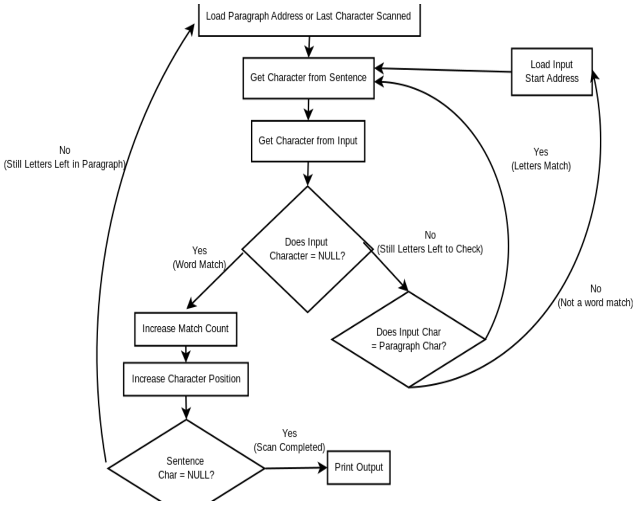
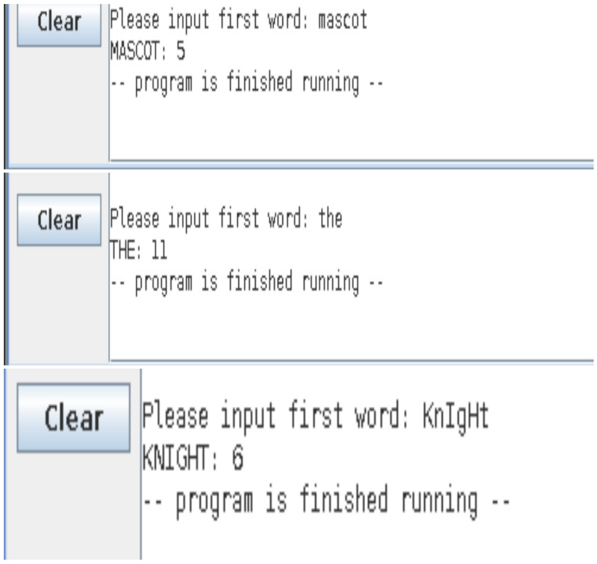


Fig. 1: Flowchart



```
Please input first word: mascot
MASCOT: 5
-- program is finished running --
```

```
Please input first word: the
THE: 11
-- program is finished running --
```

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

Fig.2: Sample outputs of the assembly program.

their letters varied randomly. The sample outputs from this testing are shown to the right in Fig. 2.

II. Look-Up Table Circuit

A LUT is a device that helps reduce the processing power required for operations by storing the result of operations in memory. LUTs can greatly reduce the latency when starting hardware or applications. When constructing LUTs, the signal propagation must be strong enough to lower the chance of loss which can result in functional failure [1].

LUTs can be designed in a variety of different fashions. One design is created using the asymmetry of spin-transfer-torque magnetic tunnel junction (STT-MTJ) current switching [3]. Another design for a LUT is called a Spin-MTJ-based LUT (spin-LUT). The spin-LUT's design enables what can be called 'instant on' but still allows for reconfiguration [2]. Nano-electro-mechanical switches with CMOS technology have several advantages that help the design stand apart from some others. They are able to operate using a very low amount of power, has a very low amount of leakage, and can be configured for permanent storage of data [4].

III. Results and Discussion

Four ALU designs implemented with LUT technology will had their energy use compared. The four designs were all processed using the same MIPS assembly code. The number of instructions were counted using the MARS 4.5 Instruction Statistics Tool.

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

Design	Energy Consumption For Each ALU Instruction
[1]	0.1 fJ
[2]	1.2 fJ
[3]	0.6 fJ
[4]	0.25 fJ

Table I shows the energy consumption of each design's ALU instruction. Design [1] using the least amount of energy for an ALU instruction, while design [2] uses the most. Table II shows the total energy consumption for each of these designs. Total energy consumption takes into account the energy required to complete the ALU, Branch, Jump, Memory and Other instructions. The program used for testing consists of a total of 19,640 instructions. As expected, design [1] uses the least amount of energy to execute, while design [2] uses the most. The results of all designs is shown in Table II.

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

Design	Total Energy Consumption
[1]	775 pJ
[2]	782 pJ
[3]	778 pJ
[4]	776 pJ

IV. Conclusion

The design of code will have a heavy impact on the amount of energy expended by the system running it. Hardware can also heavily impact this result as seen in the comparison of designs. Four ALUs designed using LUT technology were compared during this study and the top design for the least amount of energy consumed was design [1].

Several technical topics were learned and implemented throughout the design of this project and creation of the report. The following list and explanations are just a few.1) Look-up Table (LUT) Circuits – The design of, advantages and disadvantages of LUT hardware designs was learned, 2) Memory Addresses - Loading and storing data to the memory with MIPS assembly code, 3) MIPS Energy Consumption Calculations - Calculating the power used through the number of clocks a program requires to finish running, 4) Energy Conservation – Methods employed at the hardware and software level for minimizing the amount of energy required to run a program, 5) MIPS Memory Contents - Implementation of a C type array using byte offsets in MIPS assembly code.

References

[1] A. Alzahrani and R. F. DeMara, "Process variation immunity of alternative 16nm HK/MG-based FPGA logic blocks," *2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Fort Collins, CO, 2015, pp. 1-4.

[2] W. Zhao, E. Belhaire, C. Chappert, F. Jacquet, P. Mazoyer, "New non-volatile logic based on spin-MTJ," *physica status solidi (a)*, vol. 205, no. 6, pp. 1373-7, 2008.

[3] D. Suzuki, M. Natsui and T. Hanyu, "Area-efficient LUT circuit design based on asymmetry of MTJ's current switching for a nonvolatile FPGA," *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS),* Boise, ID, 2012, pp. 334-337.

[4] Y. Zhou, S. Thekkel and S. Bhunia, "Low power FPGA design using hybrid CMOS-NEMS approach," *Low Power Electronics and Design (ISLPED), 2007 ACM/IEEE International Symposium on*, Portland, OR, 2007, pp. 14-19.

[5] B. Pandey, J. Yadav, Y. Singh, R. Kumar, S. Patel, "Energy Efficient Design and Implementation of ALU on 40nm FPGA," *2013 International Conference on Energy Efficient Technologies for Sustainability,* Nagercoil, India, 2013, pp. 45-50.

[6] M. Tolba, A. Madian, A. Radwan, "FPGA Realization of ALU for Mobile GPU," *Advances in Computational Tools for Engineering Applications (ACTEA), 2016 3rd International Conference,* Beirut, Lebanon, 2016, pp. 16-20.

[7] C. Feng, L. Yang, "Design and Evaluation of a Novel Reconfigurable ALU Based on FPGA," *Proceedings 2013 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC),* Shengyang, China, 2013, pp 2286-2290.