The Comparison of Various Read Circuits and Sense Amplifier Operations Based on Energy Consumption

Corey Hogue

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

***Abstract*— Energy consumptions is a crucial factor when dealing with memory reading and writing efficiency. The main task is to compare the memory read instructions for the EASA, VISA, PWSA, and BVSC signal amplifier designs. In regards of finding the most efficient design memory reading with different sense amplifiers, an input word is gained from the program user. The program then searches for this word within a given string and outputs the number of times the users input occurred within the string. The EASA design required the least amount of energy to complete this task with only 75,027.04 fJ needed for the tested input word 'knight'.**

Keywords—Read-Memory, Bit-Cells, CMOS, MIPS, Energy Consumption, Sense Amplifiers, Efficiency, EASA, VISA, PWSA, BVSC

I. PROJECT DESIGN

The design for this project is very straightfoward. First, a prompt asking for an input word is asked from the user. This input is then stored into a string that was declared in the `.data` section. This string is then loaded into address `$s0` of which the first byte (the first letter in the string) is then loaded into `$t0`. This byte is then compared to ascii 10, which is a new line, that if equal will result in a full word match. If this is not equal, the program then stores the first byte of the provided string saved in address `$s1` to `$t1`. If `$t1` is equal to zero, the program has reached the end of the provided string and will print the result. If this byte is not equal to zero, 32 is added and subtracted and stored in `$t2` and `$t3` respectively. This allows the program to work with both captal and lowercase letters. If the bytes were equal, the program would branch to '`matchletter`' of which would increment both the input word and the string by one byte. This would continute until either the bytes did not match or until the full word was found. If the word was found, the program would branch to '`matchword`' as stated before of which increments the counter as well as the word and the string positions. Once the end of the string was reached, the program would print the input word along with the corresponding interger stating the number of times that word occurred in the sentence.

The sample inputs chosen for testing this program were strategically picked to ensure the code would function properly. The first input used was "knight" in all lowercase letters. This is basically a benchmark test that was also given in the project

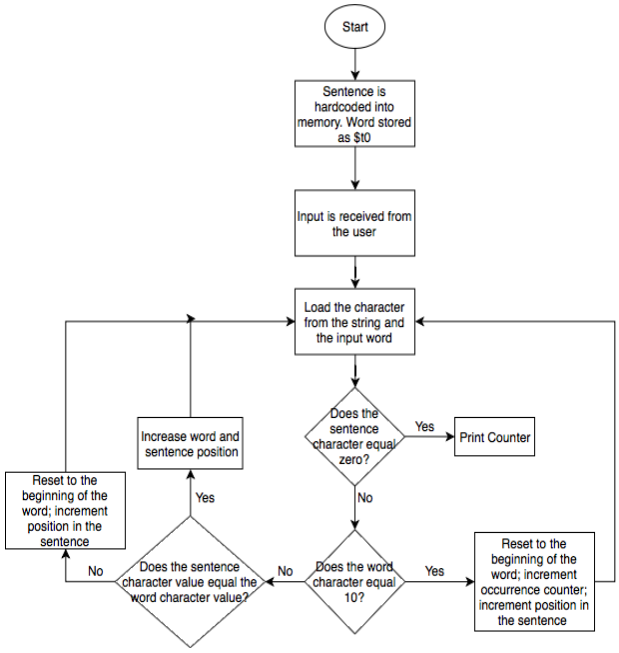

Fig.1: Flowchart of the assembly program.

```
 Please input the word: knight
knight
6
-- program is finished running (dropped off bottom) --


 Please input the word: KniGHt
KniGHt
6
-- program is finished running (dropped off bottom) --


 Please input the word: if
if
0
-- program is finished running (dropped off bottom) --
```

Fig.2: Sample outputs of assembly program.

prompt to ensure the correct output would be displayed. The second input tested was "*KniGHt"*. This ensured that the program could find the word within the sentence regardless if the word was given in upper or lowercase letters. Also, this second test was the same word used in the first test, therefore if the outputs did not match, it would be obvious that there was an error in the code. The third input used was "if". This input was chosen because the string being searched through contained zero "if"s within it, therefore this input was a good test to see how the code would react if there were no matches found within the string.

II. Memory Bit-Cells

The various sense amplifier (SA) designs being explored for this experiment were Energy Aware SA (EASA), Variation Immune SA (VISA), Pre-read and Write SA (PWSA), and the Body Voltage Sensing Circuit (BVSC). A sense amplifier is basically a part of the read circuitry that will take a low voltage signal from the *bitline* and amplify the low voltage to higher levels so the data can be understood outside of the memory. Each different sense amplifier design being tested has their individual pros and cons.

The first design being tested is the Energy Aware Sense Amplifier (EASA). The EASA design implements transmission gates (TGs) to reduce leakage energy within the circuit [1]. This helps to conserve energy as all of the energy required to process information gets utilized. A downside to the EASA design is that the usage of TGs does give added resistance which can result in read errors [1].

The second design is the Variation Immune Sense Amplifier (VISA). Process variation is reduced in this design due to the use of inverters in the circuitry which also reduces the leakage energy [1]. The inverters used within this design are the most noticeable difference between the EASA and VISA designs, however these inverters serve a crucial role in this design by lowering the chances of bit error occurrence [1].

The third tested design is called the Preread and Write Sense Amplifier (PWSA). As the name states, this design implements both reading and writing tasks within the same circuit. Since these tasks are done within the same circuit, this greatly reduces the time needed to execute read and write operations [2].

The final design to be tested was the Body Voltage Sensing Circuit (BVSC). This design heavily relies on proper resistance to become efficient [3]. If the resistance becomes too high or too low, the sense margin becomes more widespread which in return will restrict the sensing speed.

III. Results and Discussion

To calculate the energy consumption of the program, energy values for each type of instruction were provided. The input word used for the following calculations was '*knight*'. The 4,348 ALU instructions required 1fJ per instruction resulting in 4,348 fJ total. The 626 jump instructions required 3fJ per instruction, resulting in 1,878 fJ total. The 3,042 branch instructions required 2fJ per instruction, resulting in 6,084 fJ total. The 6 other instructions required 5fJ per instruction, resulting in 30 fJ total. The energy total for the memory instructions were calculated using the following table:

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

Design	Energy Consumption For Each Bit-cell's Read Operation
EASA [1]	0.23 fJ
VISA [1]	1.86 fJ
PWSA [2]	36.0 fJ
BVSC [3]	195.5 fJ

Given that the Memory = Read Energy + 50fJ (Write Energy), the 50fJ was added to each energy value provided in Table 1 and then multiplied by the number of memory instructions, which was 1248. Each one of these energy values for memory was then added to the energies for ALU, Jump, Branch, and Other separately to determine the total energy consumption for each design. These results are provided in Table 2 below:

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

Design	Total Energy Consumption
EASA [1]	75,027.04 fJ
VISA [1]	77,061.28 fJ
PWSA [2]	119,668.0 fJ
BVSC [3]	318,724.0 fJ

IV. Conclusion

The program written and tested is a commonly used algorithm used on every computer referred to as "find". Trying to develop a program that can solve this task using minimal energy is another task within its own. The program written functioned with the lowest energy consumption using the **EASA design**, requiring **75,027.04 fJ** to find the number of times the input word 'knight' occurred within the given string. This is only 23.5% of the energy required when implementing the same code and input on the BVSC design; of which was the least efficient. This project focused on instruction statistics, energy consumptions, read and write based memory, as well as code writing techniques using branches, jumps, and byte loading and referencing. Implementing a working code that will solve a task is only a small fraction of what is to be expected from engineers. Efficiency plays a huge role with software and hardware development and is typically a major factor in everyday engineering tasks.

References

[1] S. Salehi and R. F. DeMara, "Process variation immune and energy aware sense amplifiers for resistive non-volatile memories," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, 2017, pp. 1-4.

[2] H. Lee *et al*., "Design of a Fast and Low-Power Sense Amplifier and Writing Circuit for High-Speed MRAM," in *IEEE Transactions on Magnetics*, vol. 51, no. 5, pp. 1-7, May 2015.

[3] F. Ren, H. Park, R. Dorrance, Y. Toriyama, C. K. K. Yang and D. Marković, "A body-voltage-sensing-based short pulse reading circuit for spin-torque transfer RAMs (STT-RAMs)," Thirteenth International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, 2012, pp. 275-282.