MIPS ALU Energy Consumption Calculation in Reference to Various RNG Circuit Designs

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

Abstract **-** This paper studies various different random number and stochastic number generator circuits using p-bit designs. The ALU energy consumption from these designs is then utilized to calculate the total energy of a program, which detects the number of times two words, entered by the user, occurs in a user-entered string. Furthermore, the project design also calculates the index at which the word is matched. The lowest energy consuming design is reference [1] at 0.49 micro joules.

Keywords – RNG, SNG, ALU, Magnetic Tunnel Junction (MTJ), Phase Change Memory (PCM), P-bit

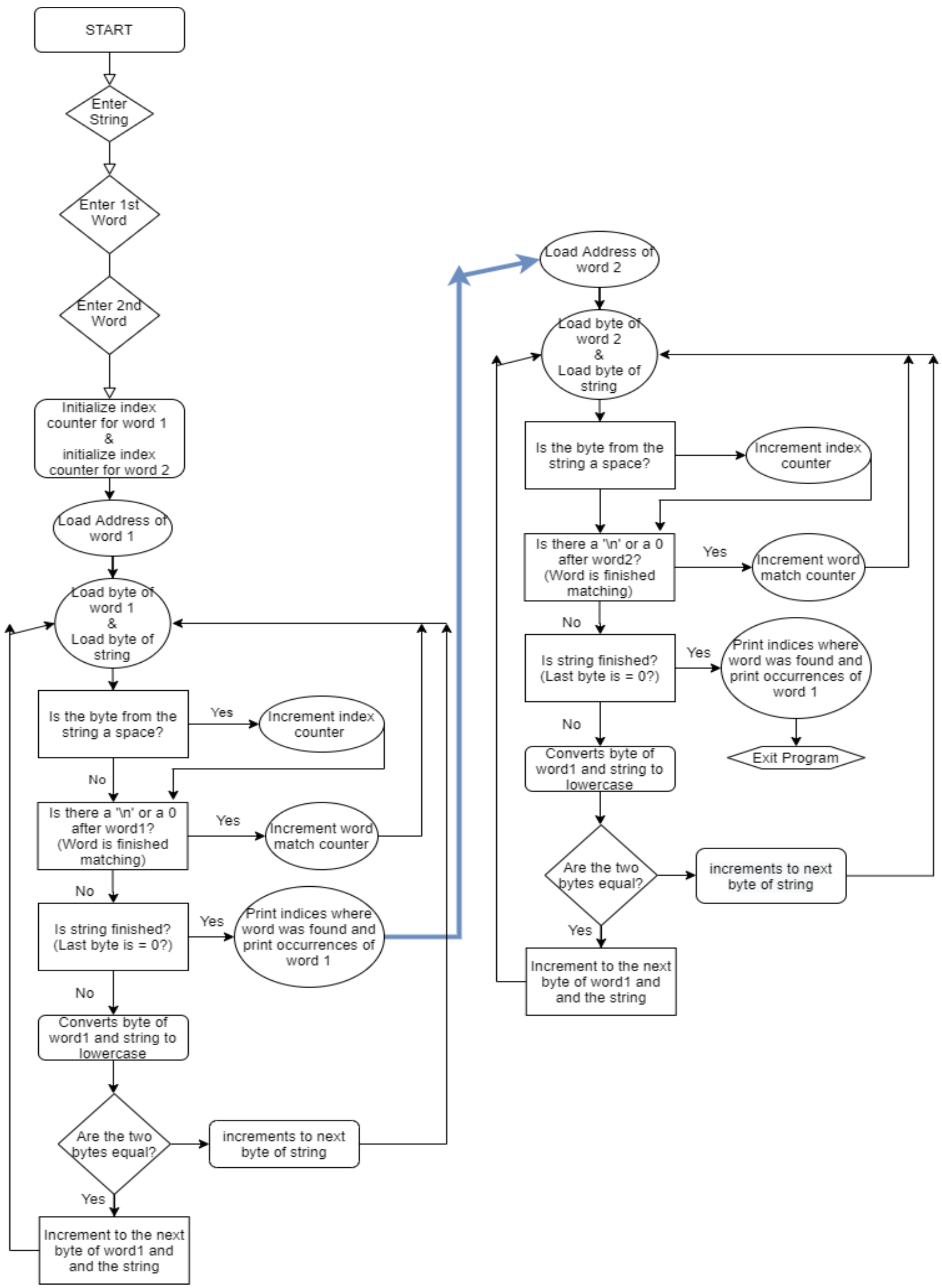
I. Introduction

The objective of this project was to identify the occurrence of two specific words in a string entered by the user utilizing MIPS assembly code in the MARS simulator. The program also returns the index of where the words were found within the string. Three different sentences, that include different inputs will be utilized to test the program's outputs. Furthermore, the program's energy consumption will be calculated utilizing provided references.

A. Project Design

The program will be designed to identify how many times two specific words occur in a string entered by the user and also returns the index at where it was found. The program will first ask the user to enter a string utilizing a syscall and then asks the user to enter two different words, word1 and word2, to be found in the string. The program first focuses on determining the occurrences of the first word and then replicates the method using the same registers with the second word. The first byte from Word1 and the first word of the string are loaded and stored into registers $t0 and $t2. Word1 and the first word of the string are then converted to either upper or lowercase. These two bytes are checked to see if they match. If they match the program increments to the next byte of word1 and checks if it matches again with the next byte of string. However, if the bytes do not match, word1 is reset to its initial byte and continues to search for a match of its first byte or first letter. When all matches are found of the word1, which means that all letters match with a letter in the string, a counter is incremented. This counter will be utilized to determine the occurrences. If, for example, 6 matches are found, the counter, initialized at zero, will be incremented 6 times. This will be how many matches are found. Furthermore, in order to determine the index of where the word is found, we can implement a counter that continuously counts the number of spaces after each word. This counter keeps track of which word it is on. The counter can be printed when a match is found, thus returning the location or index of where the matched word is located. The same process is replicated using word2 and then the values are printed.

Figure 1: Flowchart of Project Design



B. Test Cases

The first input I am deciding to go with is a simple string, consisting of 8 words, to test the basic functionality of the design. I will enter the string "The dog is faster than the other dog" I will search for the words "dog" and "fast."

The second input will be a longer string consisting of 26 words. with more words that are the same. The next input string will be "A peck of pickled peppers Peter Piper picked; If Peter Piper picked a peck of pickled peppers, Where's the peck of pickled peppers Peter Piper picked?" I will search for the repeated words of "peck" and "pick."

The third input will be a large string consisting of 122 words: "The Knights Graduation and Grant Initiative is a UCF award to help undergraduate students who cannot pay their tuition and their difficulty would not allow them to finish their degree. The Knights Success Grant is the most well-known program inside the Knights Graduation and Grant Initiative. In order to be awarded the Knights Success Grant, you need to be referred but it does not mean that all students who are referred will be awarded the grant. The students who want to apply for the Knights Success Grant need to submit a required application and complete the Knights Success Grant web course. For more information, you can stop by their office in the Registrar's Office on the main campus of UCF." I will search for the words kNigHt and UcF. This time, we will see if the design adjusts the casing of the input words entered and match them to the words from the string.

Figure 2: Test Outputs of Design

```
Please type in your input statement:
The dog is faster than the other dog.

Please type in a word (up to 10 characters) that you are looking for (e.g. Knight or KnIGhT, knight, &):
dog

Now please enter a second word:
fast

------------------------------------------------------------
Indexes of the matches found: 2, 8,

Number of times the word dog
was found in the input statement: 2

------------------------------------------------------------
Indexes of the matches found: 4,

Number of times the word fast
was found in the input statement: 1

-- program is finished running --
```

```
Please type in your input statement:
A peck of pickled peppers Peter Piper picked; If Peter Piper picked a peck of pickled peppers, Where's the

Please type in a word (up to 10 characters) that you are looking for (e.g. Knight or KnIGhT, knight, &):
peck

Now please enter a second word:
pick

------------------------------------------------------------
Indexes of the matches found: 2, 15, 22,

Number of times the word peck
was found in the input statement: 3

------------------------------------------------------------
Indexes of the matches found: 4, 8, 12, 16, 22, 26,

Number of times the word pick
was found in the input statement: 6

-- program is finished running --
```

```
Please type in your input statement:
The Knights Graduation and Grant Initiative is a UCF award to help undergraduate students who cannot pay t

Please type in a word (up to 10 characters) that you are looking for (e.g. Knight or KnIGhT, knight, &):
kNigHt

Now please enter a second word:
UcF

------------------------------------------------------------
Indexes of the matches found: 2, 32, 42, 53, 85, 97,

Number of times the word kNigHt
was found in the input statement: 6

------------------------------------------------------------
Indexes of the matches found: 9, 122,

Number of times the word UcF
was found in the input statement: 2

-- program is finished running --
```

II. P-BIT CIRCUIT

Three references each discuss a method of creating a Random Number Generator (RNG) or Stochastic Number Generator (SNG); both utilize P-bit or probabilistic bit, which fluctuates between the 0 state and 1 state, to function as a program that produces a string of random bits. The program relies on an intrinsic stochasticity in physical variables, which is the source for the generated randomness. Three different designs that produce random bits are showcased

The first p-bit design [Reference 1] discusses an energy-efficient SNG design used for spin-based neuromorphic circuits. The design utilizes a transistor and a magnetic tunnel junction (MTJ). The MTJ resistance fluctuates two resistive states, this causes the output voltage to fluctuate as well. The transistor allows the input voltage to amplify from the threshold voltage, which produces a stochastic sigmoidal output. This SNG design is used for nanomagnetic devices within neuromorphic structures.

The second method [Reference 2], discusses an SNG circuit design for Phase Change Memory (PCM). The inherent stochasticity of PCM is used to design a scalable and energy efficient circuit by reducing its area. The University of Kentucky's design has roughly up to 300 times lower area, which leads to a 250 times decrease in energy consumption. This is all in comparison to a Linear Feedback Shift Register. The design takes an N-bit input, which is converted into a 2^N bit stochastic bit-vector. Furthermore, the design utilizes a PCM-based Gaussian digital-to-analog converter, which aids the reference voltage in reading a full row of 2^N PCM cells. The block diagram showcases the functionality of utilizing a 3-bit binary input (B2 B1 B0); however, the design can be generalized for any bit binary input (BN-1, B1 B0). When the N number of bits enter the GDAC, the N number of switches help connect the bits to their respective N number of voltage sources. The voltage source Vx equal to the current pulse (I READ) multiplied by the inverse of the cumulative distribution function. This allows the resistance value Rx to be obtained for the given probability number.

The third p-bit circuit [Reference 3] discusses a lightweight and energy-efficient true micro random number generator [uRNG] design. This design combines three entropy inputs A, B, and C to produce an output bit stream. These three entropy sources are linked to their respective voltages and clock grids, producing a total of 3 random bits/clock. Furthermore, this design utilizes multiple cycles of data-whitening, which allows the individual true random number generators [TRNG] to converge at their respective steady-states. This combined approach of using entropy sources along with data-whitening proves to be quite compact and energy efficient; the design has a 6.5 times reduction in total gate count and 5.4 times lower energy consumption when compared to conventional entropy extractors, which rely on only one entropy source.

III. RESULTS AND DISCUSSION

In this section, the energy consumption of the program design will be calculated using the below energy consumption per instruction values.

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

It is assumed that an RNG circuit is used in the implementation of ALU. The energy consumption of the RNG circuit is used as the energy consumption of a single ALU instruction. The energy consumption values reported from references [1-3] will be used to calculate the total energy consumption of our program design. The below table lists the required energy consumption to perform each ALU instruction based on the different technologies proposed in [1-3].

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

Design	Energy Consumption For Each ALU Instruction
[1]	0.08 pJ
LFSR [2]	0.9 pJ
GDAC [2]	90 pJ
[3]	36 pJ

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

Design	Total Energy Consumption
[1]	0.49 μJ
LFSR [2]	0.5 μJ
GDAC [2]	2.35 μJ
[3]	1.23 μJ

IV. CONCLUSION

A Three different RNG/SNG circuits using P-bit were analyzed. Their respective ALU energy consumption values were incorporated in the total energy calculation of the program design at hand. The most energy efficient design was reference 1 at 0.49 μJ.

Technical Topics Learned

- RNG Circuit Design
- ALU energy calculation
- Stochastic Design
- Comparing Bytes in MIPS
- Neuromorphic Circuits

References

[1] H. Pourmeidani, S. Sheikhfaal, R. Zand, and R. F. DeMara, "Probabilistic Interpolation Recoder for Energy-Error-Product Efficient DBNs with p-bit Devices," IEEE Transactions on Emerging Topics in Computing, 2020.

[2] S. M. Shivanandamurthy, I. G. Thakkar, and S. A. Salehi, "Work-in-Progress: A Scalable Stochastic Number Generator for Phase Change Memory Based In-Memory Stochastic Processing," in Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis Companion, pp. 1-2, October 2019.

[3] S. K. Mathew, D. Johnston, S. Satpathy, V. Suresh, P. Newman, M. A. Anders, H. Kaul, A. Agarwal, S. K. Hsu. G. Chen, and R. K. Krishnamurthy, "µRNG: A 300–950 mV, 323 Gbps/W all-digital full-entropy true random number generator in 14 nm FinFET CMOS," in IEEE Journal of Solid-State Circuits, vol. 51, no. 7, pp. 1695-1704, 2016.