Controlling Power Dissipation in Full-Adders by Enhancement of Logic Gate Design and Transistor Reduction

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

***Abstract*— A Full Adder circuit takes in two to three inputs and produces, at a maximum, two outputs. These inputs and outputs are required to compute binary addition. A standard Full Adder will successfully produce one of eight possible outputs, according to its truth table. Approximate Mirror Adders are enhancements made to the simple Full Adder. The purpose of enhancing a Full Adder is to decrease power dissipation, lower energy consumption, and reduce hardware area being used. The Approximate Mirror Adders also cause some error to occur on the output side, which is not seen in a standard Full Adder; this is due to the removal of transistors. AMA3 consumes the least amount of energy, 3.5 pico-Joules for this design, compared to the Conventional Full Adder, which consumes 143.3 pico-Joules.**

Keywords—Full Adder, Approximate Mirror Adder, Conventional Mirror Adder, Carry-out, Carry-in, Truth Table, Inversion, Hardware Area, Logic Gate, Sum

I. Project Design

The purpose of this design was to calculate the amount of times a user-inputted word occurred in a given sentence. The sentence is pre-loaded into the data. The first step in the code is to prompt the user to enter their word of choice. This input is then loaded into a register. The string, the given sentence, is also loaded into a register. Next, the program enters the start loop. Initially the first byte of the string and the word are loaded into different registers. Then the program checks if the end of the string has been reached. If so, the code will jump to the exit line of code. If the string is not finished, then the program will next check for a completed word. If this is true, then the word counter will be incremented, the word will be reset, the string byte location will be incremented, and the code will jump back to the start of the loop. If there isn't a word match at this point, then the code will check for a character match. If no match, then the code will add 32 to the word character value and check again. Then it will subtract 64, if still no match is found, and check for the last time. If there is no character match, then the program will increment the string byte, reset the word byte, and jump back to the start. If there is a character match, then both the word and string registers will be incremented, and the loop is restarted. The only way for this code to finish is if the end of the string has been reached.

This program was tested by using inputs of varying case and characters. The first input used was 'KNIGHT', in order to verify that the uppercase was being detected, which gave a result of six. The next test input was 'knight', used to check for the opposite lowercase condition, also producing a result of six occurrences. To check that upper and lower case were working

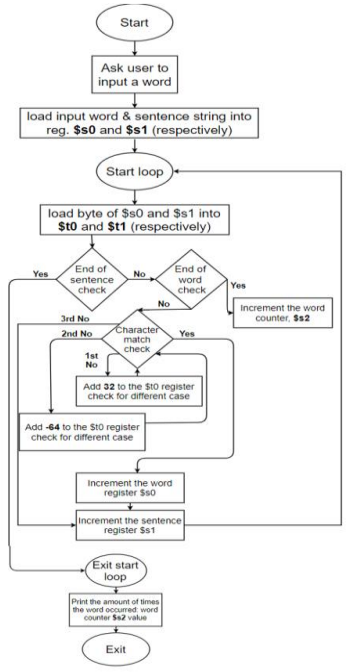


Fig.1: Flowchart of the assembly program.

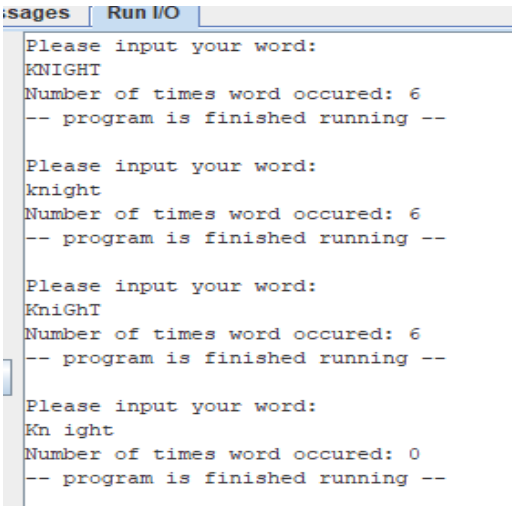


Fig.2: Sample outputs of the assembly program.

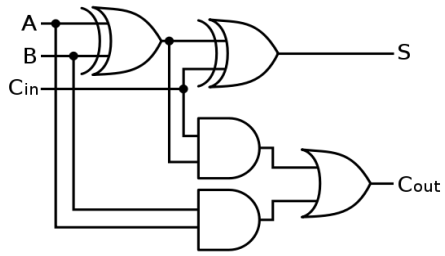
harmoniously, the input given was 'KniGhT'. The results were a success. A final test used the input 'Kn ight' to verify that the program accounted for spaces, which correctly produced zero results.

II. Full-Adder Circuit

Full Adders (FA) are pieces of hardware that contain a combination of logic gates. The inputs for a single FA are A, B, and Carry in (C_{in}). The outputs are Sum and Carry out (C_{out}). The reason there is a C_{in} is because there will be many FAs connected to one another and the way this is done is by connecting a C_{out} to the next $C_{in.}$ The following truth table will show this process more clearly:

Inputs			Outputs	
A	**B**	**C_{IN}**	**Sum**	**C_{OUT}**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Logic gate representation of a Full Adder:



There are various approximations of a FA. A Conventional Mirror Adder (CMA) is a standard FA, as seen in the diagram above, which also consists of 24 transistors. In order to decrease the number of transistors, reduce the hardware area used, and lower power dissipation, the CMA must be altered. In Approximation 1 (AMA1), transistors are removed from the CMA and the hardware is tested for open or short circuits. In this approximation, eight transistors are removed and only 6 out of the 8 cases are still correct. This circuit makes Sum = C_{OUT} (inverse). With this technique, Sum produces three errors and C_{OUT} has one. These errors signify the differences in output compared to the CMA, there is an acceptable amount of error based on what the result is. In Approximate Mirror Adder 2 (AMA2), C_{OUT} is set to equal A, forcing C_{OUT} to have two errors. It is also noticed that the circuit can be further simplified by inverting input A and calculating C_{OUT} and Sum as in a CMA. This method results in two errors for C_{OUT} and three errors for Sum. In AMA1 and AMA2 the C_{OUT} value is calculated using an inverted C_{OUT} as input. In the case of Approximate Mirror Adder 3 (AMA3), Sum has four errors. By setting Sum equal to input B and C_{OUT} equal to input A, we get a result of only four errors out of eight total outputs. AMA 3 also uses the least amount of area, as well as the least amount of total energy (as seen in Table II).

III. Results and Discussion

The results of this program were used to calculate its energy consumption based on the consumption per instruction values below:

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

The total energy consumption for this program, not including the ALU instruction, was 135.841 nano-Joules. The overall focus of this program was to identify which type of Full Adder had the lowest power dissipated. Compared to the CMA circuit, the AMA 1 and 2 remove eight transistors which reduces the amount of power used. AMA 3 has the highest number of errors produced, but it only has 1 ferro-Joule of energy consumed for each ALU instruction. These major differences in energy may outweigh the need to minimize output error.

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

Based on the results below, it is evident that the 3rd AMA was the most energy efficient. These dynamic instruction values were obtained by accessing:

MARS4.5➔Tools➔Instruction Statistics.

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

Design	Total Energy Consumption
[1]	135.859 nJ
CMA [2]	135.984 nJ
AMA [2]	135.885 nJ
[3]	135.845 nJ

IV. Conclusion

There is more than one type of Full Adder. The various types can be used to alter the power dissipated while running a program through the Arithmetic Logic Unit (ALU), which is where the Full Adders reside. A standard Full Adder, or a Conventional Mirror Adder, contains 24 transistors and produces zero errors from any of its eight possible outputs. There are more than three different hardware changes that can be made to the CMA to improve its energy consumption, but in this paper, the ones listed are Approximate Mirror Adder 1, 2, and 3. AMAs 1 and 2 calculate the Carry out value by inverting the Carry out from the preceding Adder. AMA 3 produces the most amount of errors (4 out of 8) but it has a substantially lower energy consumption. The total energy consumption of the best design for this program was 135.845 nano-Joules from using the Approximate Mirror Adder 3, which is 139 pico-Joules less than the Conventional Mirror Adder.

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 Magnetics*, vol. 49, no. 9, pp. 4982-4987, Sept. 2013.