

“Design of Speed–Area–Power Optimized Ternary Logic Gates Based on Standard MOS Technology”

K Nagalakshmi ¹, T Anusha ², T.Sajida ³

¹PG Scholar, ²Assistant Professor, ³Assistant Professor, Dept of ECE, Anantha Lakshmi institute of technology & sciences, Anantapuramu.

Abstract— Previous papers have focused solely on designing ternary logic gates or constructing adders using predesigned ternary logic gates. However, this paper takes a novel approach by designing ternary logic gates using typical MOSFETs and employs them to construct a ripple carry adder. Furthermore, this paper conducts a comprehensive comparison between the resulting ripple carry adder and binary ripple carry adder in terms of speed, power(energy) consumption, and area. This paper demonstrates that the ternary adder designed by the author, using PSpice and Verilog with VPI, can exhibit differences 75% compared to the conventional binary adder.

Keywords—ternary logic gates, ternary ripple carry adder, speed, power(energy) consumption, area

I. INTRODUCTION

The use of ternary allows for the representation of larger numbers with fewer digits compared to binary [1]. Moreover, it is known to have advantages in various aspects. So, utilizing PSpice, a new architecture can be designed using the typical MOS transistors technology to create ternary NOT, AND, and OR logic gates. Subsequently, these gates are used to construct a ternary ripple carry adder and a commonly known binary ripple carry adder, which are then compared in terms of speed, power(energy) consumption, and area. The speed of the adders is measured using the delays of each gate. Power(energy) consumption is calculated as the product of power and time, applying [3] as a reference, and the area is measured through the size and quantity of transistors.

The circuits designed through PSpice are implemented in Verilog HDL. Each logic gate, along with the measured delays, is incorporated into the design. Random numbers are generated from C and received via the VPI interface between Verilog and C. This allows for actual simulation of the operational speed of ternary and binary ripple carry adders.

II. PROPOSED METHOD

$$\text{trit} = (\text{bit}) * \log_3 2 \quad (1)$$

Equation (1), it can be understood that a 21-trit ripple carry adder needs to be composed in order to construct a 32-bit ripple carry adder. First, let's design a ternary NOT gate. In ternary, unlike binary, there are three possible cases: 0, 1, and 2. When the input is 0, the output of the NOT gate should become 2 due to the pull-up voltage, and when the input is 2, the output should be set to 0 by the pull-down voltage. However, when the input is 1, all MOSFETs will be turned on, and for the output to become 1, the voltage needs to be appropriately distributed. Therefore, to facilitate the role of

voltage distribution, MOSFETs were added. Both MOSFETs need to serve as resistors, so they must operate in the triode region. Therefore, gate voltages of 0V and 2V were respectively applied to the PMOS and NMOS. The AND and OR gates are each composed of combinations of NAND gates with NOT gates and NOR gates with NOT gates. For the NAND and NOR gates as well, voltage distribution is necessary in situations where the output should be 1. To achieve this, PMOS and NMOS operating in the triode region were added. Also their gate voltages are 0V and 2V, respectively. For voltage distribution purposes, the additional PMOS and NMOS transistors were designed with lower threshold voltages and significantly shorter lengths and widths. This was done to fulfill the role of voltage distribution while minimizing resistive properties and thus minimizing the delay of ternary logic gates. Fig. 1 shows the circuit diagrams presented earlier along with the lengths, widths, and threshold voltages of the components. Using the designed ternary logic gates in this way, a ternary ripple carry adder is constructed. To create a ternary full adder, Karnaugh Map was adopted [1]. The utilized Karnaugh Map and the designed full adder are depicted in Fig. 2 and Fig. 3, respectively. By connecting 21 of these full adders together, a ternary ripple carry adder will be formed.

Fig. 4.(a) represents the results obtained by examining all possible cases of output changes for each logic gate to calculate the delay. So, in Verilog, we implemented both binary and ternary adders using obtained logic gate delays. Inputs were assigned random numbers within the same range from C, and the time taken to complete the operation through a total of 500 iterations was measured. As a result, it took 60.984ns for the binary case, and 15.098ns for the ternary case. The results obtained through Verilog confirm an operational speed difference of approximately 75% between the two.

$$E_{dynamic} = \int \alpha f_{clk} C V_{DD}^2 dt \quad (2)$$

The equation for dynamic energy consumption of a logic gate is given by (2). $E_{dynamic}$ depends on five main factors the ratio between the time it takes for a logic gate to operate once and the time it takes for the clock signal to operate α , clock frequency f , capacitance C , the supply voltage V_{DD} , delay of each logic gate t . If we assume a clock signal frequency of 1GHz, it can be observed from Fig. 4.(a) that ternary logic gates and binary logic gates operate in 2 clocks and 1 clock, respectively. Therefore, α for each is 0.5 and 1. And 't' in (2) can be referenced from Fig. 4.(a). The overall dynamic energy consumption should be multiplied by the number of logic gates. The number of logic gates used can be

determined using the critical path shown in Fig. 4.(b). As a result, for the ternary ripple carry adder is 4.74×10^{-7} [J], whereas the result for the binary ripple carry adder is 2.3×10^{-7} [J].

$$E_{overall} = \int p_{static} dt + E_{dynamic} \quad (3)$$

The equation for overall energy consumption is given by (3). As previously discussed, dynamic energy consumption may be less favorable for the ternary adder. However, static energy consumption ceases when the adder's operation stops. Therefore, the ternary adder, which performs calculations more quickly, is expected to have an advantage in terms of static energy consumption. Based on this concept, we have highlighted the range in Fig. 5 where the overall energy consumption of the ternary adder can be advantageous, represented by the yellow area. The x-axis and y-axis of Fig. 5 represent the static energy consumption of the ternary adder and the binary adder, respectively.

Finally, utilizing the sizes mentioned in Fig. 1, I can calculate the sizes of logic gates and then estimate the approximate sizes of each ripple carry adder. The result is that the ternary ripple carry adder's size is approximately 12445×10^{-10} [m²], whereas the size of the binary ripple carry adder is approximately 5616×10^{-10} [m²]. The ternary ripple carry adder's size is about twice as large as the binary counterpart. However, considering the percentage, they are still sufficiently small to have minimal impact.

III. CONCLUSION

In terms of speed, an improvement of 75% has been achieved. Although dynamic energy consumption and area were less favorable in the ternary adder, but when considering the improvement percentage in speed, the difference is reasonable. Moreover, by taking static energy consumption into account, we were able to identify conditions where the ternary adder can be advantageous.

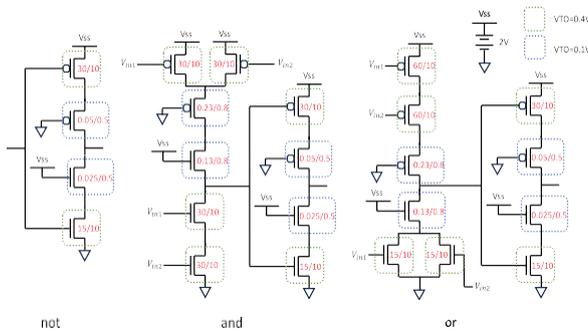


Fig. 1. Circuit diagrams of ternary logic gates (units:[μm])

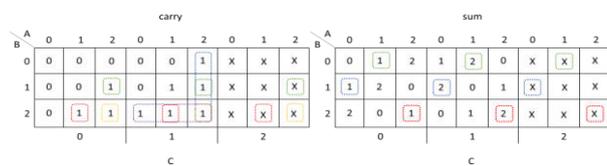


Fig. 2. Karnaugh Map for designing a ternary full adder

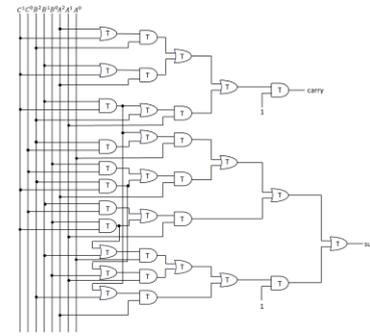


Fig. 3. Circuit diagram of a ternary full adder

gate	type	input	delay	average	
and	ternary	1	0->1 1.8164	1.35015	
		0->2 1.2556			
		1->0 1.1796			
		2->0 1.3527			
		0->1 1.3461			
	0->2 1.2050				
	1->2 0.8033				
	1->0 1.4018				
	2->0 1.3756				
	2->1 1.7654				
binary	1	0->2 0.2564	0.26535		
	2->0 0.2743				
	0->1 1.6719				
	0->2 1.5766				
	1->2 1.3568				
or	ternary	0	1->0 1.0027	1.40709	
		2->0 1.1518			
		2->1 1.5113			
		0->2 1.3437			
		1->2 1.1193			
	binary	1	2->0 1.2718		0.27695
		2->1 2.065			
		0->2 0.2564			
		2->0 0.2743			
		0->2 0.0403			
not	binary	0	2->0 0.0399	0.0401	

critical path	
b	$33t_{not} + 64t_{and} + 64t_{or}$
t	$43t_{and} + 64t_{or}$

(a) Delays of logic gates (b) Propagation delays of binary and ternary carry adder

Fig. 4. Delays of logic gates (units:[ns]) and critical paths

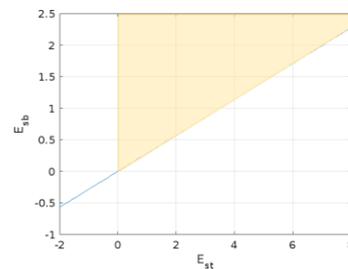


Fig. 5. Range of static energy consumptions

ACKNOWLEDGMENT

This study was supported by (4199990113966), (NRF-2018R1A6A1A03025109), (NRF-2022R111A3069260), (2020M3H2A1078119), (No. 2021-0-00944), (No. 2022-0-01170), (No. RS-2023-00228970), (No. RS-2022-00156389). The EDA tool was supported by the IC Design Education Center (IDEC), Korea.

REFERENCES

- [1] V. Ghiye, S. Bonde and A. Dhande, "Investigation of Ternary Function Minimization," 2014 Fourth International Conference on Communication Systems and Network Technologies, Bhopal, India, 2014, pp. 1054-1058
- [2] J. Park and Y. Kim, "Design and Implementation of Ternary Carry Lookahead Adder on FPGA," 2021 International Conference on Electronics, Information, and Communication (ICEIC), Jeju, Korea (South), 2021, pp. 1-2
- [3] V. T. Gaikwad and P. R. Deshmukh, "Design of CMOS ternary logic family based on single supply voltage," 2015 International Conference on Pervasive Computing (ICPC), Pune, India, 2015, pp. 1-6