



Design and Implementation of Unsigned Serial Divider Using TG Logic

G. Kavya¹✉, K. Shreshta Reddy¹, D. Vishnu Prasad¹, J. V. R. Ravindra¹,
and Himanshu Rajeshbhai Dodiya²

¹ Center for Advanced Computing Research Laboratory (C-ACRL), Department of Electronics and Communication Engineering, Vardhaman College of Engineering Hyderabad, Hyderabad, India

kavyag39147@gmail.com, jayanthi@ieee.org

² RK University, Rajkot, Gujrat, India

Abstract. The paper details the design and implementation of a 4-bit unsigned binary serial divider using 90 nm CMOS technology. It utilizes Transmission Gate (TG) logic for the division process and offers a comparison with an alternative CMOS logic approach featuring a Kogge Stone adder. The TG Logic implementation boasts a lower transistor count (402) but exhibits higher propagation delay (379.6 femtoseconds) and power consumption (18.42 micro-Watts). In contrast, the CMOS Logic with Kogge Stone adder employs more transistors (792) but results in a shorter propagation delay (494.9 femtoseconds) and lower power consumption (5.987 micro-Watts). The choice between these two methods should be based on the specific application's needs and priorities.

Keywords: power consumption · high-speed · TG logic · CMOS

1 Introduction

In the realm of digital systems, division stands as a fundamental operation. Binary dividers can be categorized into two types: serial and parallel dividers. The process of serial division involves repetitive subtraction. Consider the scenario where we aim to divide 18 by 3. We iteratively subtract 3 from 18. After six such subtractions, the remainder reaches zero, indicating that it's now smaller than the divisor. As a result, the further subtraction process ceases. Consequently, the quotient is six, and the remainder is zero. To mitigate area and power usage, various techniques such as CMOS, TG, and PT logic are employed [1]. While these techniques have aimed to reduce area and power consumption, they've encountered limitations such as low logic levels and increased circuit complexity.

2 The Proposed Algorithm – Motivation

The advancement of digital technology has made mathematical operations more efficient, but there's a noticeable focus on improving methods for addition and multiplication, rather than division [2]. This bias exists because creating efficient division logic is more

complex. In terms of speed, addition and multiplication usually require only a few clock cycles, while division takes much longer, spanning many cycles. Neglecting to improve division for modern computer applications can lead to a decline in overall computer performance.

2.1 One’s Complement Binary Subtraction Algorithm

Assuming that Y and X are both nonzero values and X is greater than Y, we initially observe a high carry output from the adder. This is because we are adding X with the two’s complement of Y (Fig. 1).

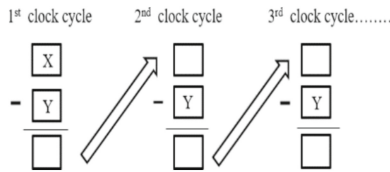


Fig. 1. Execution flow of the algorithm

Upon examining the adder’s output after the first clock cycle, we find the result of X - Y, which is then stored within the data register. Meanwhile, the counter’s output is incremented to 0001. Subsequently, in the following cycle, if the condition (X - Y) is still greater than Y, we proceed to subtract Y from X. This action leads to an increment in the counter’s output, now reflecting the value 0010. This process of subtraction continues iteratively until the output of the adder becomes high [4]. This occurrence marks the point where the subtraction process stops.

2.2 Execution Flow of Algorithm with Example Sum

The One’s complement method of subtraction is used to divide binary values 1010 (decimal 10) 0010 (decimal 2). In the serial divider circuit, this action is repeated. When the difference is positive, the ultimate carry is 1, which is added to the actual difference. When the difference is negative, the carry is zero, and the real result is one’s complement of the sum output (Fig. 2).

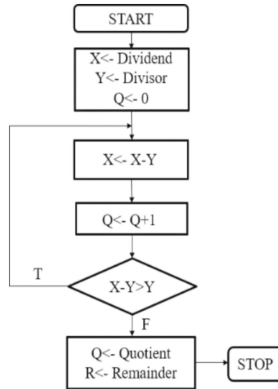


Fig. 2. Flow Chart of 1's Complement Binary Subtraction algorithm

3 Design of Unsigned Serial Divider

Serial hardware architecture employs a sequential approach to implement the various components of an algorithm [5]. This type of architecture finds its utility primarily in general-purpose applications, whether in central processing units (CPUs) or field-programmable gate arrays (FPGAs).

3.1 Design Approach

This section outlines the strategy for constructing a 4-bit unsigned binary serial divider via the technique of iterative subtraction involving two 4-bit unsigned binary numbers. The divisor, denoted as $Y_3 Y_2 Y_1 Y_0$, is subtracted from the dividend $X_3 X_2 X_1 X_0$ using the one's complement subtraction method. The fundamental building blocks employed in this design are as follows:

- Adder: This component handles the addition of two 4-bit binary numbers.
- 2:1 Multiplexers (MUX): A group of four of these multiplexers is utilized.
- D Flip-Flops: A set of four D flip-flops.
- 4-bit Binary Synchronous Up Counter: Used to count synchronously (Fig. 3).

3.2 Design of the 4-bit Adder

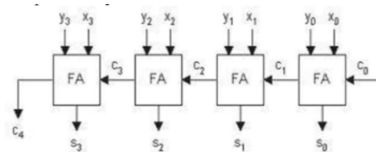


Fig. 3. 4-bit Ripple Carry Adder

The design incorporates a ripple carry adder, which is constructed by utilizing multiple full adders [10]. A depiction of the straightforward implementation of the 4-bit ripple carry adder is provided in Fig. 4. In this setup, C0 signifies the input carry, while X0 through X3 and Y0 through Y3 represent the two 4-bit input binary numbers.

3.3 Design of 2:1 Multiplexer

The divider’s multiplexer, composed of two transmission gates and an inverter, switches between inputs A and B according to control signal S. This mechanism allows controlled and flexible data selection, making it a vital component in digital circuit operations.

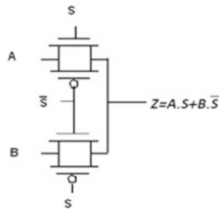


Fig. 4. Transmission gate 2:1 MUX

3.4 Kogge Stone Adder Design

The Kogge Stone Adder is fundamentally a parallel prefix adder, recognized for its remarkable speed in addition operations based on design considerations [7]. This design approach stands out for its exceptional speed during the design phase. It operates by calculating propagate signals denoted as ‘Pi’ and generate signals denoted as ‘Gi’ using the ith bit of the provided input [9] (Fig. 5).

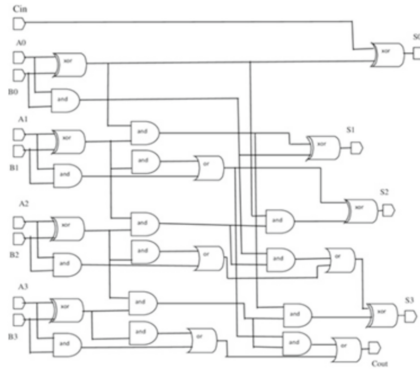


Fig. 5. 4-bit Kogge Stone Adder

3.5 Design of D Flip-Flop

The negative edge-triggered master-slave D flip-flop utilizes D latches with an enable input to synchronize changes with a clock signal's falling edge. During the high clock phase, it stores the D input in the master latch, while the slave latch remains unchanged. As the clock transitions from high to low in the low clock phase, the master latch's output transfers to the slave latch, allowing the flip-flop's output to change [11]. This design ensures that the output can only change during the clock's high-to-low transition. The master latch tracks the D input when the clock is high, and the inactive slave latch maintains the output. This flip-flop is crucial for precise timing in digital circuits [15]. Incorporating the clear (clr) and enable (CE) inputs enhances the flip-flop's functionality:

- When the clear input is activated (high), regardless of the D input and enable signal, the output is forced low.
- When the enable input is deactivated (low), regardless of the D input and clear signal, the output maintains its previous state (Fig. 6).

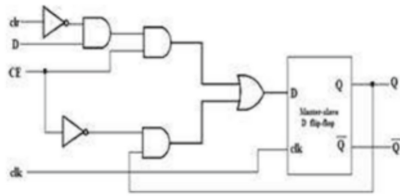


Fig. 6. D-Flip Flop with clear and CE inputs

3.6 Design of 4-bit Synchronous up Counter

The synchronous 4-bit up counter is composed of three AND gates, four XOR gates, and four master-slave D flip-flops. Each flip-flop receives the same clock pulse, leading to synchronized operation. Here's how it functions:

- The counter is designed as an up counter, commencing from the value 0000.
- With every clock pulse, the counter increments by one step. This means it progresses from 0001, 0010, 0011, 0100, and so on, until it reaches 1111.
- After reaching 1111, it resets back to 0000, thus completing one full counting cycle (Fig. 7).

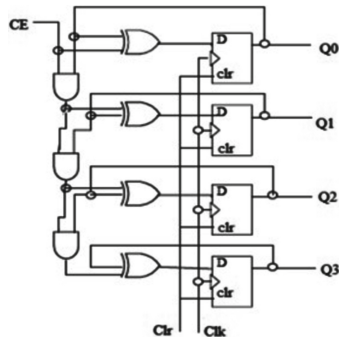


Fig. 7. 4-bit Synchronous up counter

4 Simulation Results

Using the Cadence EDA tool, the design of the final divider circuit is simulated and the transient responses are examined.

4.1 Schematic Diagrams and Simulations

Figures 8, 9, 10 and 11.

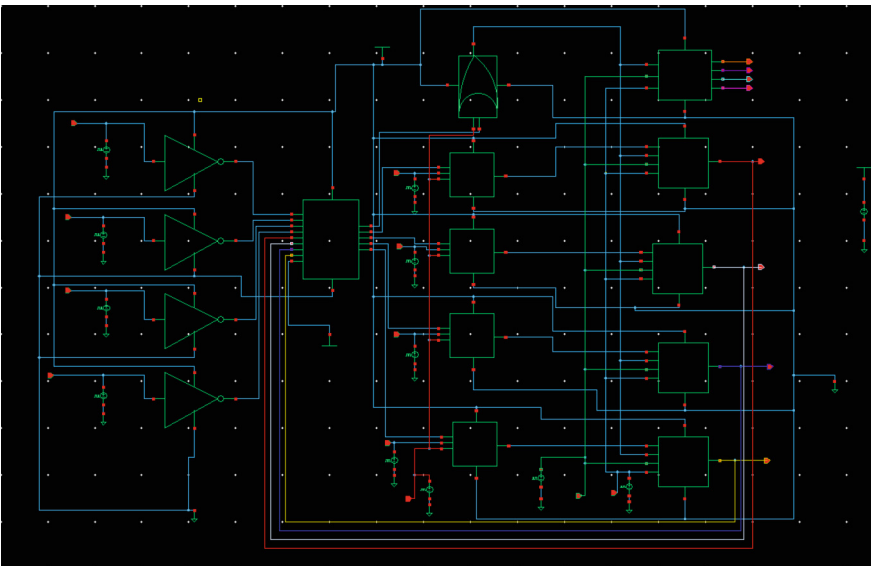


Fig. 8. Schematic of 4-bit Unsigned Serial divider circuit

Table 1. Power and Delay Analysis of CMOS and TG based serial divider

Conventional CMOS				TG Logic		
Circuits	Transistor Count	Power	Delay	Transistor Count	Power	Delay
Full Adder	28	1.036 μ W	37.31 ps	20	0.497 μ W	20 ps
RCA	112	6.31 μ W	298.5 ps	80	1.94 μ W	20.08 ps
2:1 MUX	20	41.76 μ W	277.6 ps	6	0.097 μ W	16.71 ps
Master Slave D Flip Flop	62	106.9 μ W	71.31 ns	46	500.8 μ W	498.5 fs
Counter	210	2.049 μ W	211.4 ns	114	1.72 μ W	499.6 fs
Serial Divider	664	5.7 μ W	85.68 ps	402	18.42 mW	379.6 fs
Full Adder	28	1.036 μ W	37.31 ps	20	0.497 μ W	20 ps

Table 2. Power and Delay Analysis of serial divider with Kogge Stone Adder

Circuits	Transistor Count	Power	Delay
Kogge Stone Adder	240	8.314 μ W	500 fs
Serial Divider	792	5.987 μ W	494.9 fs

**Fig. 12.** Power and Delay graphs

5 Conclusion

Division, a fundamental arithmetic operation, has posed implementation challenges. Focus on improving its efficiency spans digital circuits, computers, and embedded systems, aiming to optimize area, hardware, and latency. Efforts have introduced novel algorithms with simplified logic for diverse applications, but sequential implementation offers simplicity at the expense of latency and conversion time. In specific results, a TG Logic divider showcased 402 transistors, 379.6 fs delay, 18.42 μ W consumption,

while a CMOS/Kogge Stone approach yielded 792 transistors, 5.987 μ W, 494.9 fs delay. Balancing efficiency factors drives division implementation strategies.

References

1. Soheil Hashemi, R.I.B., Reda, S.: A low-power dynamic divider for approximate applications. *IEEE Design Automation Conference (DAC)*, pp. 1–6 (2016)
2. Liu, W., Tao, X., Li, J., Wang, C., Montuschi, P., Lombardi, F.: Design of unsigned approximate hybrid dividers based on restoring array and logarithmic dividers. *IEEE Trans. Emerg. Top. Comput.* **10**(1), 339–350 (2022)
3. Santhosh Reddy, G., Sai Vandana, K., Subramanya Sai, S.V.R., Ravindra, J.V.R.: Efficient Design of MISTO CDL Adder Cell for High Throughput and Power-Efficient Array Divider. *Soft Computing and Signal Processing 3rd ICSCSP 2020, Volume 1*, pp. 595–605 (2020)
4. Parhami, B.: *Computer Arithmetic: Algorithms and Hardware Designs*. Oxford series in electrical and computer engineering. Oxford University Press (2010). ISBN: 9780195328486
5. Patankar, U.S., Koel, A.: Review of basic classes of dividers based on division algorithm. *IEEE Access*, 23035–23069 (2021)
6. Purohit, S., Laddha, P., Parekh, R.: Implementation and Physical Design of 8/4-Bit Signed Divider. *8th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 829–834 (2021)
7. Kommineni, R., Arunsai, G., Ravindra, J.V.R.: Low Power, Noise-Immune High Performance Arithmetic Adder Circuit Design Using Modified Parallel Prefix Adders. *Inventive Communication and Computational Technologies* (2023)
8. Deepak, S., Ganesan SaiKrishnan, D.R., Ravindra, J.V.R.: Seek-method based 2's complement circuit for low power circuit and high-speed operation. *AIP Conference Proceedings*, vol. 2407. AIP Publishing (2021)
9. San, A.M., Yakunin, A.N.: Reducing the hardware complexity of a parallel prefix adder. *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 1348–1351 (2018)
10. Koyada, B., Meghana, N., Jaleel, O., Jeripotula, P.R.: A comparative study on adders. *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 2226–2230 (2017)
11. Panda, S., Sharma, S., Asati, A.R.: Clock gating analysis of TG based d flip-flop for different technology nodes. *IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pp. 1–6 (2020)
12. Robertson, J.E.: A new class of digital division methods. *IRE Transactions on Electronic Computers EC-7.3*, pp. 218–222 (1958)
13. Mitchell, J.N.: Computer multiplication and division using binary logarithms. *IRE Transactions on Electronic Computers EC-11.4*, pp. 512–517 (1962)
14. Aoki, T., Nakazawa, K., Higuchi, T.: High-radix parallel VLSI dividers without using quotient digit selection tables. *Proceedings 30th IEEE International Symposium on Multiple-Valued Logic*, pp. 345–352 (2000)
15. Sharma, M., Noor, A., Tiwari, S.C., Singh, K.: An area and power efficient design of single edge triggered DFlip flop. *International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 478–481 (2009)