Design and Optimization of a 4-bit Binary Synchronous Counter Based on Logical Effort

Hongjia Lin[1]

23307130025@m.fudan.edu.cn [1]

College of Integrated Circuits & Micro-Nano Electronics, Fudan University, Shanghai, China[1]

Abstract. Synchronous binary counters are essential for digital timing and control, yet optimizing them for both speed and energy efficiency remains challenging. While Logical Effort theory lays a solid foundation for delay minimization, its application often overlooks energy efficiency. This paper presents the design and optimization of a 4-bit synchronous binary counter, leveraging Logical Effort. The work introduces a two-stage methodology: first, transistor sizes are determined for minimal delay; second, an energy-driven optimization phase employs Energy-Delay Product metric to adjust device sizing and supply voltage. Results show a combined strategy of scaling both transistor widths and VDD achieves an Energy-Delay Product reduction of 27.82%, outperforming optimizations using only width scaling (-22.15%) or only voltage scaling (-9.89%). The structured approach successfully extends Logical Effort analysis from pure delay minimization to a balanced energy-delay optimization, offering a practical framework for designing energy-efficient VLSI circuits where both speed and power are critical.

Keywords: Logical Effort; Energy-Delay Product; Synchronous Counter; CMOS Design; VLSI Optimization.

1 Introduction

Synchronous binary counters are foundational components in digital circuits, critical for timing, sequencing and control in applications ranging from analog-to-digital converters, phase-locked-loop frequency synthesizers, frequency dividers and so on [1]. With higher stability and lower delay, synchronous counters are really suitable for scenarios requiring precise time control [2]. As semiconductor technology scales, it has become a great challenge to optimize circuits for both high speed and low power under increasing design constraints. Traditional synchronous counter designs, while superior to asynchronous counterparts in speed and reliability, often rely on empirical methods. With huge fan-outs and extensive carry chains, they are limited in counting rate, mainly because the size of the counter is not modest [3]. With fixed fan-out scaling, counters cannot achieve an optimal balance between delay and energy consumption. Considering the specific complexity and load conditions of a given circuit path systematically is the key to this problem.

ICMEEA 2026, April 24-26, Singapore, Singapore

DOI 10.4108/eai.24-4-2026.2364888

The Logical Effort methodology provides a rigorous theoretical framework for delay minimization by modeling path delay as a function of logical and electrical effort. It enables the determination of optimal transistor sizes to achieve minimal propagation delay for a given topology [4]. Although well-established for simple logic chains, its systematic application to multi-bit synchronous counters offers significant potential for improvement. This is particularly true for such counters since their critical paths exhibit non-uniform logical effort so energy efficiency must be the key design objective. Prior work has solely focused on delay optimization, which can lead to excessive power consumption [5]. This leaves a gap for a holistic design flow that extends Logical Effort analysis to optimizing Energy-Delay Product (EDP), a key metric for Very Large Scale Integration (VLSI) systems [6].

This paper presents the design and optimization of a 4-bit binary synchronous counter using the Logical Effort theory. The primary goal is to demonstrate a complete methodology that transits from theoretical delay minimization to a balanced optimization of energy efficiency. The work includes a systematic critical path analysis using Logical effort to decide initial transistor sizes, followed by an energy-driven optimization phase. In this phase, strategies such as voltage scaling and device sizing are evaluated using the EDP as the core metric to identify the best performance-power trade-off. The optimized design is verified through simulation and compared against a conventional baseline to quantify its advantages.

The contribution of this work mainly lies in two aspects. First, it provides a structured design flow that systematizes the application of Logical Effort for transistor sizing in a multi-bit counter, which is inspiring [7]. Second, it introduces a two-stage optimization process. The initial stage focuses on minimal delay, while the second stage employs EDP-guided fine-tuning to achieve a superior balance between speed and energy consumption, advancing the design towards a more practical, multi-objective optimum.

The rest of the paper is organized as follows. Chapter 2 reviews the theoretical background of Logical Effort and synchronous counters. Chapter 3 details the critical path of the design and the complete optimization process. Chapter 4 concludes the work and proposes future research directions.

2 Theoretical Background

This chapter establishes the theoretical foundation for the work presented in this paper. It details the core principles of the Logical Effort theory for digital circuit optimization. It also demonstrates how synchronous counter works briefly.

2.1 Fundamentals of Logical Effort

The Logical Effort methodology provides a formal, technology-independent framework for analyzing and minimizing the delay of CMOS digital circuits, a technique applied for accurate delay estimation in modern circuit design [8]. Its primary strength lies in decoupling the intrinsic properties of logic gates from the effects of loading and fanout, enabling a systematic approach to transistor sizing.

The delay of a single logic gate is modeled as the sum of two key components. The first is effort delay (f), which is the product of logical effort (g) and electrical effort (h). g quantifies the

intrinsic driving capability of a gate's topology relative to a simple inverter, which is defined to have a g of 1. More complex gates, such as NAND or NOR gates, have larger logical efforts because their internal transistors deliver less current for the same input capacitance. h, also known as the fanout, is defined as the ratio of the load capacitance the gate drives to its input capacitance. The second component is the parasitic delay (p), which represents the delay intrinsic to the gate due to its own parasitic capacitance, independent of the load.

For a path consisting of N logic gates,

$$G = \prod_{i=1}^{N} g_i \tag{1}$$

$$H = \frac{C_{out}}{C_{in}} \tag{2}$$

$$F = GBH \tag{3}$$

Where path logical effort (G) is the product of g in all stages, path electrical effort (H) is the ratio of Cout to Cin, and the final path effort (F) is the product of G, H and the branching effort (B). B often takes the value of 1 if there is no branching in the path.

Once the value of F is known, optimization by sizing each logic gates can start to put into practice.

$$f^* = \sqrt[N]{F} \tag{4}$$

$$h = \frac{f^*}{g} \tag{5}$$

$$C_{in} = \frac{C_{out}}{h} \tag{6}$$

As F and N are determined, so the optimal stage effort (f*) can be calculated. Then h of each stage is also clear since the value of g in each stage is already known. Finally, the input capacitance for each logic gate can be obtained. Therefore, by applying these equations sequentially from the output load back to the beginning of the path, the optimal transistor widths for every gate in the critical path can be determined, minimizing its overall propagation delay [4].

2.2 Principles of Synchronous Counters

A synchronous binary counter is a sequential digital circuit that advances through a binary sequence upon each active edge of a common clock signal. The critical feature of a synchronous counter, in contrast to an asynchronous (ripple) counter, is that the clock signal is fed simultaneously to all state-holding elements (flip-flops). This eliminates the cumulative propagation delay in ripple structures, allowing for significantly higher maximum operating frequencies and output decoding without glitches, which is critical for precise timing and control applications [9].

The functional implementation of a 4-bit synchronous binary up-counter typically consists of two core parts: a state register formed by four D flip-flops (one for each bit), and a combinational next-state logic circuit. The outputs of the D flip-flops (Q_0 to Q_3) directly represent the current 4-bit value. During each clock cycle, the combinational logic block computes the next binary value based on the current value and presents it to the D inputs of the flip-flops. When the next

active clock edge comes, all flip-flops synchronously update their outputs to the new value, thus advancing the count. This synchronous update mechanism is the defining characteristic of the design. The central task of it consists of the specific logic equations that define the counting sequence, the detailed transistor-level implementation and critical path optimization of the combinational circuit. All of these will be elaborated in chapter 3.

3 Design and Optimization of the Counter

3.1 System Architecture

This section demonstrates the top-level design of the 4-bit binary synchronous up-counter. The design should meet the demand of giving the correct output during each clock cycle by the 4 D flip-flops and computing the correct value through the combinational logic circuit. Naming the input of the 4 D flip-flops to be D0, D1,D2 and D3, there should be:

$$D_0 = Q_0{}' \tag{7}$$

$$D_1 = Q_1 \oplus Q_0 \tag{8}$$

$$D_2 = Q_2 \oplus (Q_1 Q_0) \tag{9}$$

$$D_3 = Q_3 \oplus (Q_2 Q_1 Q_0) \tag{10}$$

With these equations giving the logical expression of D0, D1,D2 and D3, the design can be depicted:

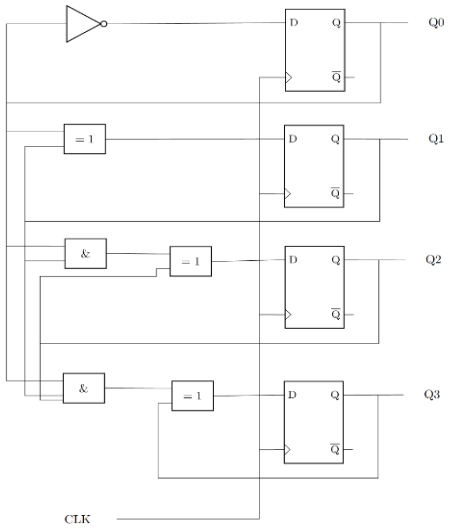

Fig. 1. Design of the 4-bit Binary Synchronous Counter [10].

Figure 1 reveals the up-level architecture of the whole design. With the architecture, it's time to determine the critical timing path of the counter, which determines its maximum operating frequency.

As it is obvious that the logical expression of D3 is the most complicated, the critical path should originate from the output of the LSB flip-flop (Q_0), propagates through the AND3 gate, and finally through the XOR gate to the D input of the MSB flip-flop (D_3).

This critical path shows the propagation delay of the combinational chain deciding the minimum clock period, making it the primary target for the transistor-level optimization detailed in the following sections.

3.2 Transistor-Level Circuit Design of Key Modules

Now the block-level design gives out the clear depiction of the critical timing path, the next step should go down to the specific transistor-level design of all the logic gates on the critical path.

Before starting to detail the structure, one thing to confirm is that this paper determines to make use of Complementary Metal Oxide Semiconductor (CMOS) and the transmission gate to complete the implementation of all the logic gates on the critical path, including the AND3 gate and the XOR gate. The reason is that fundamental logic gates such as inverters (INV) and multi-input NAND gates are typically implemented using static CMOS topology, while transmission gates provide an efficient alternative for implementing specific functions like XOR, according to the classics [11]. What's more, in consideration of the need of optimization for logic gates sizing, using CMOS and transmission gates can achieve the explicit value of g, h and p. This will simplify the calculation of logical effort and the optimal capacitance of each logic gates in the critical path to a great extent, thus achieving the goal of minimal delay through the whole design flow.

One more thing have to mention is that if static CMOS topology is used to struct the AND3 gate, the gate should be divided into two part: a NAND3 gate and an INV. It is because the NAND and NOR structures are the fundamental and most efficient primitives due to their simple series (NMOS) and parallel (PMOS) transistor arrangements. A direct, monolithic CMOS implementation of an AND3 function would result in an inefficient and slow transistor network. Therefore, the standard and optimal method to realize an AND3 is to cascade a NAND3 gate followed by an INV, which together form a high-performance, standard composite cell.

After this clarification, the N of the critical path is determined as 3, meaning a 3-stage logic chain: a NAND3 gate (Stage 1), followed by an INV (Stage 2), driving an XOR2 gate (Stage 3).

Now it's time to detail the transistor-level design of all the three stages.

Setting up a baseline first is really important. The INV in stage 2 should take on this role since it has the simplest CMOS structure. An important prerequisite is the paper assume that the PMOS-to-NMOS width ratio is 2:1. It is because in the Logical Effort methodology, the unit inverter serves as the foundational reference for delay and capacitive load. To establish a fair baseline, its PMOS and NMOS transistors are sized to provide symmetric rising and falling times, achieving balanced pull-up and pull-down drive strengths. This symmetry is critical

because the hole mobility in PMOS transistors is typically 2 to 3 times lower than the electron mobility in NMOS transistors in standard CMOS processes. Therefore, a PMOS width twice that of the NMOS is a common and well-justified design choice to compensate for this mobility difference, ensuring the unit inverter itself represents a performance-neutral point for subsequent stage-effort calculations and transistor sizing optimizations [4].

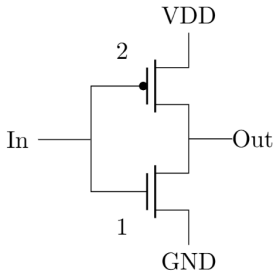

Fig. 2. Unit Inverter [11].

As shown in figure 2, if the width of NOMS in the unit inverter is marked as 1, then the width of PMOS will be 2.

This unit inverter can actually be directed implemented as the stage 2 design in the critical path.

With the baseline locked, the structure of NAND3 and XOR2 can be carried out.

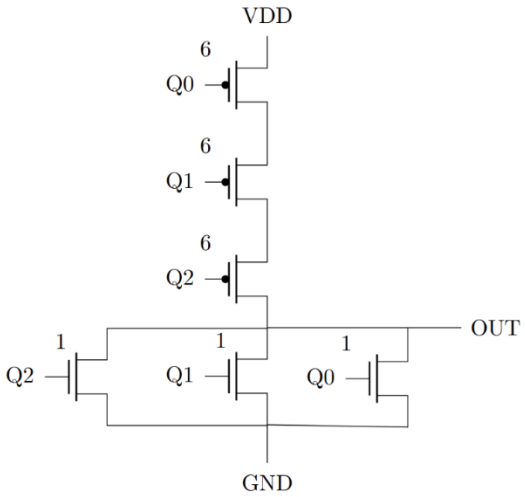

Fig. 3. NAND3 Gate [11].

As shown in figure 3, due to the unit width of the INV, the width of 3 NOMS in the NAN3 gate is 1 while the width of 3 PMOS is 6 according to the series-parallel relationship.

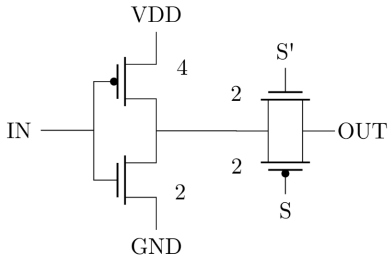

Fig. 4. XOR Gate [11].

As shown in figure 4, the size of each NMOS and PMOS can all be determined based on the unit INV. What's more, the usage of transmission gate can partly reduce the g and p of the XOR gate, thus optimizing time delay.

3.3 Logical Effort Analysis and Gate Sizing

With all the detailed structure, calculations can be done to have all the g and p of each stages.

For NAND3,

$$g_1 = \frac{6+1}{2+1} = \frac{7}{3} \tag{11}$$

$$p_1 = \frac{6+1+1+1}{2+1} = 3 \tag{12}$$

For INV,

$$g_2 = \frac{2+1}{2+1} = 1 \tag{13}$$

$$p_2 = \frac{2+1}{2+1} = 1 \tag{14}$$

For XOR,

$$g_3 = \frac{4+2}{2+1} = 2 \tag{15}$$

$$p_3 = \frac{2+2}{2+1} = \frac{4}{3} \tag{16}$$

Assume H as 8, and B is obviously 1 because there is no branching. With all the data, according to formula (3), the result of F is 112/3. According to formula (4), f* is 3.342. According to formula (5) and (6), there is finally the table of gate sizing.

Table 1. Gate sizing of each stages in the critical path.

Stage	3 XOR	2 INV	1 NAND3
C_{out}	8	4.787	1.432
g	2	1	2.333
h	1.671	3.342	1.432
C_{in}	4.787	1.432	1

As shown in table 1, all the input capacitance are known now, whose ratio are also the ratio of gate sizing to each stage of the critical path.

$$D = \sum_{i=1}^{3}(g_i h_i + p_i) = 15.36\tau \tag{17}$$

As shown in formula (17), the delay of the whole design flow (D) is 15.36τ.

3.4 Energy-Delay Optimization

With all the input capacitance known, so does the size ratio of each logic gates. Before starting the energy-delay optimization, there are some assumptions that have to be made so that calculations can be simplified and the results can get closer to real situations.

First, γ is assumed to be 1, so that the delay of each stage is:

$$d = gh + \gamma p = gh + p \tag{18}$$

Second, the Clock Enable to Clock-to-Data ratio (CE/CD) is assumed to be 1.2, so that the energy consumption of each stage is (only considering "0" → "1"):

$$e_i = \frac{1}{2}\frac{CE}{CD}(c_i\gamma + c_{i+1})VDD^2 = 0.6(c_i + c_{i+1})VDD^2 \tag{19}$$

Third, VDDref and VT are assumed to be 1V and 0.2V respectively, so that the total delay is:

$$D = \frac{\sum d_i VDD}{(VDD-V_T)^2} * \frac{(VDD_{ref}-V_T)^2}{VDD_{ref}} = \frac{0.64\sum d_i VDD}{(VDD-0.2)^2} \tag{20}$$

After all the assumptions, the energy and delay baseline can be calculated in table2:

Table 2. Energy and Delay Baseline.

Stage	1	2	3	Total
Energy	1.4592	3.7314	7.6722	12.8628
Delay	6.3413	4.3429	4.6757	15.3600

So the baseline EDP is 197.5717.

The paper decides to discuss three approaches of optimization: only adjusting the width, only adjusting the VDD, and adjusting width and VDD at the same time. By limiting the delay region below 150% of the baseline delay, the final results can be given:

Table 3. Optimized Size, Energy, Delay and EDP.

Design	Stage 2	Stage 3	VDD	Energy	Delay	EDP
Baseline	1.432	4.787	1	12.8628	15.3600	197.5717

W only	0.637	1.974	1	8.5334	18.0235	153.8019
W+VDD	0.722	2.205	0.833	6.1893	23.0400	142.6009
VDD only	1.432	4.787	0.775	7.7268	23.0400	178.0261

As shown in Table 3, all three approaches have optimized the key metric EDP to varying degrees, with the "W+VDD" approach having the best optimization ratio, the "W only" approach following closely, and the "VDD only" approach being the least optimized.

Table 4 turns the optimization ratio into quantified percentage:

Table 4. Optimization Ratio.

Design	dEnergy	dDelay	dEDP
Baseline	0	0	0
W only	-33.66%	17.34%	-22.15%
W+VDD	-51.88%	50.00%	-27.82%
VDD only	-39.93%	50.00%	-9.89%

4 Conclusion

This paper has successfully realized the application of Logical Effort theory to the design and optimization of a 4-bit synchronous binary counter, transiting from pure delay minimization to a balanced energy-delay optimization targeting the EDP. The two-step methodology first utilizes Logical Effort to determine transistor sizes for minimal delay along the critical path, and then introduces EDP-guided fine-tuning through device sizing and supply voltage scaling. Results show that a combined width and voltage scaling strategy achieves the most significant EDP reduction of 27.82%, substantially outperforming optimizations based solely on width (-22.15%) or voltage (-9.89%). Therefore, the study concludes that a holistic approach integrating delay optimization with energy-aware post-tuning is essential for modern VLSI circuits, where energy efficiency is as critical as speed performance.

The primary contribution of this work lies in providing a structured, generalizable design flow that systematizes the use of Logical Effort for complex multi-bit sequential circuits. By addressing the energy-delay trade-off, this methodology compensates for the shortcomings in prior delay-centric optimizations and offers a practical framework for others seeking a balance between performance and power. The demonstrated flow not only benefits the design of counters but also serves as a valuable reference for optimizing other digital circuit modules with logical efforts, such as adders or comparators, under tight power constraints.

Looking forward, several directions can extend this research. First, the methodology can be applied to counters with higher bit widths or different architectures, i.e., synchronous down-counters or loadable counters to further validate its scalability. Second, taking process-voltage-temperature (PVT) variations into consideration would enhance the robustness of the design in practical manufacturing conditions. Finally, exploring the integration of this logical-effort-based sizing flow with advanced optimization techniques, such as machine-learning-assisted parameter search or dynamic voltage and frequency scaling (DVFS) schemes, could unlock

further improvements in energy-efficient circuit design. Future work may also consider extending the EDP metric to account for leakage power and layout-dependent effects, moving closer to a full-system optimization perspective.

References

[1] Y. Hyun, I.-C. Park, Constant-Time Synchronous Binary Counter With Minimal Clock Period. IEEE Trans. Circuits Syst. II 68, 2645-2649 (2021) doi:10.1109/TCSII.2021.3054014

[2] H. Chen, Design and Implementation of MOD60 Counter Based on Multisim, in Proceedings of the 5th International Conference on Materials Chemistry and Environmental Engineering, 75-82 (2025) doi:10.54254/2755-2721/124/2025.20090

[3] V. Yadav, P. P. Bansod, D. K. Mishra, 64 Bit Binary Counter with Minimal Clock Period, in Proceedings of the ICAECT 2022 conference, ITM Web of Conferences 50, 02005 (2022) doi:10.1051/itmconf/20225002005

[4] I. Sutherland, B. Sproull, D. Harris, Logical Effort: Designing Fast CMOS Circuits (Morgan Kaufmann Publishers Inc., San Francisco, CA, 1999)

[5] Y. Aizik, A. Kolodny, Exploration of energy-delay tradeoffs in digital circuit design, in Proceedings of the 2008 IEEE International SOC Conference, 1-5 (2008)

[6] H. Seok, H. Seo, J. Lee, Y. Kim, COREA: Delay- and Energy-Efficient Approximate Adder Using Effective Carry Speculation. Electronics 10, 2234 (2021) doi:10.3390/electronics10182234

[7] M. A. S, P. Poongodi, Adaptive prairie dog optimization based variable length conditional counter for designing multiplier. Integr. VLSI J. 99, 102243 (2024) doi:10.1016/j.vlsi.2024.102243

[8] S. Goswami, V. Chandana, A. Dandapat, An Approach to Design a Low Power High-Speed Full Adder Circuit Based on Logical Effort. Electron. 28, 3 (2024) doi:10.53314/ELS2428003G

[9] V. Chioktour, A. Kakarountas, Constant delay systolic binary counter with variable size cellular automaton based prescaler. Comput. Electr. Eng. 93, 107291 (2021). DOI: 10.1016/j.compeleceng.2021.107291

[10] J. F. Wakerly, Digital Design: Principles and Practices (Prentice Hall, Upper Saddle River, NJ, 2006)

[11] N. H. E. Weste, D. M. Harris, CMOS VLSI Design: A Circuits and Systems Perspective (Addison-Wesley, Boston, 2011)