

A Logical Circuit Optimization in Balancing Delay and Energy Consumption

Qihang Shan*

The Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University,
Blacksburg, 24061, USA
sqihang8@vt.edu

ABSTRACT

The fast-developing chip manufacturing technique and scaling of transistors allow us to fit more transistors on a small chip. The scaling down process, however, is facing a challenge. The smaller transistors are, the more influential quantum channeling and silicon atom size limit become. To improve efficiency, the solution of scaling down is no longer an option. Therefore, to further improve the efficiency of a chip without scaling down transistors, this paper presents a combinational circuit and focuses on an optimization approach where energy consumption is reduced in exchange for increasing delay. By adjusting the size of transistors, energy is saved while maintaining delay to an acceptable range. This approach manages to reduce energy consumption by about 56% while increasing delay by 50%. This paper represents one of many possible approaches that researchers had and has been working on and this tradeoff can benefit some circuit designs depending on the circuit's purpose and hope to bring some insights on further optimization.

CCS CONCEPTS

• **Hardware** → Power and energy; Power estimation and optimization; Circuits power issues; Integrated circuits; Logic circuits; Combinational circuits.

KEYWORDS

Optimization, Tradeoff, Energy Consumption, Delay

ACM Reference Format:

Qihang Shan*. 2023. A Logical Circuit Optimization in Balancing Delay and Energy Consumption. In *6th International Conference on Information Technologies and Electrical Engineering (ICITEE 2023)*, November 03–05, 2023, Changde, HUNAN, China. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3640115.3640128>

1 INTRODUCTION

With the advancement of technology, computers have brought us great convenience. The development of computing technology was a process of fitting more transistors on a chip by scaling transistors down [1]. However, as the sizes between transistors and silicon atoms are closing in, it is becoming challenging to further scale down transistors [2]. At first, a transition from BJT to MOSFET is favored to adapt the fast-developing chips since MOSFETs have

better efficiency when power is low and switching frequency is high [3]. However, the chip industry is approaching the new upper limit. As a result, finding an alternate solution to improve the performance of nanoscale-integrated chips becomes a trend nowadays. Currently, different designs of transistors and biochips are being explored to increase performance. This paper presents a 4-bit absolute value comparator optimization: an approach of balancing energy consumption and delay to improve performance.

This comparator consists of two 4-bit inputs: A and B, where both inputs are in its 2's complement form if is negative. This logical function block is made possible from two parts: converter and comparator. The purpose of the converter is to generate a 3-bit absolute value for both 4-bit input A and B, whereas the purpose of the comparator is to generate 1 if 3-bit input A is larger than 3-bit input B and 0 if otherwise. After the topology phase, the minimum delay is calculated, and the voltage supply is set to 1V. Then, delay and voltage are adjusted to reduce its energy consumption. In the optimization phase, energy consumption is reduced by a significant amount in exchange for a delay increased to 1.5 minimum delay. This trade-off allows us to further reduce the computer's energy consumption without further scaling down transistors. 2's complement form is a way of expressing negative figures using binary bits where the first bit always indicates its sign (positive or negative). To convert a positive bit to its negative counterpart, all bits are negated and then add 1; To convert a negative bit to its positive counterpart, the same conversion is used [4].

2 CIRCUIT DESIGN

Assuming the input capacitance of all inputs is less than or equal to 2-unit sized inverters; C load = 32 unit-sized inverter; Gamma (C parasitic/C gate) = 1; Vt=0.2V.

2.1 Converter

This converter should convert 2 4-bit inputs A (A3, A2, A1, A0) and B (B3, B2, B1, B0) to 3-bit absolute values A' and B'. This example shows the conversion of 4-bit input A.

4-bit input A should generate a 3-bit output A' (A2', A1', A0'). Since all numbers are in its 2's complement form if the first bit is 0, no conversion is needed, and if the first bit is 1, negate all bits and add 1. The truth table of this converter is found and shown below in Table 1.

A3 is the most significant bit and A0 is the least significant bit.

2.2 Comparator

Comparators have 2 3-bit inputs A (A2, A1, A0) and B (B2, B1, B0), and 1 1-bit output S. S is 1 if A is larger than B, and S is 0 if



This work is licensed under a Creative Commons Attribution International 4.0 License.

ICITEE 2023, November 03–05, 2023, Changde, HUNAN, China

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0829-9/23/11

<https://doi.org/10.1145/3640115.3640128>

Table 1: Truth table of converter

A3A2A1A0	A2'A1'A0'	A3A2A1A0	A2'A1'A0'
0000	000	1000	000
0001	001	1001	111
0010	010	1010	110
0011	011	1011	101
0100	100	1100	100
0101	101	1101	011
0110	110	1110	010
0111	111	1111	001

otherwise. The truth table of this comparator is shown below in Table 2.

Comparators compare the most significant bits first, then compare the second most significant bits and until to the least significant bits. A binary number comparator is almost identical to a decimal number comparator. When comparing two decimal numbers, the most significant digits are compared first, if two digits are equal, the second most significant bits are compared, and so on.

3 CIRCUIT IMPLEMENTATION

With the truth table of both the converter and comparator found, the circuit can be built and simulated using Quartus. The circuit was originally built using AND and OR gates, and every AND gate and OR gate is converted to NOR and NAND gate using Boolean algebra for easier calculation of logical effort and circuit design [5, 6]. For easier demonstration, the whole circuit is divided into two parts: The converter and the Comparator.

3.1 Converter

The converter is built in Quartus and is shown in Figure 1.

Two converters have the same design and convert 4-bit inputs A and B respectively.

Take the top converter as an example. this converter takes 4-bit A (A3, A2, A1, A0) as input and generate 3-bit output (A2_Converted, A1_Converted, A0_Converted).

A3 indicates a number sign, therefore, it acts as a selection bit in this circuit. If A3 is 1, multiplexers output their input A, and if A3 is 0, multiplexers output their input B. If A3 is 0, A is positive and input directly goes to output. If A3 is 1, A is negative and each bit is calculated using a logical expression, and logical expressions of 3 bits of output A are:

$$A2_{Converted} = (\overline{A2 \& A1}) \mid (\overline{A2 \& A1 \& A0}) \mid (A2 \& A1 \& A0) \quad (1)$$

$$A1_{Converted} = A1 \oplus A0 \quad (2)$$

$$A0_{Converted} = A0 \quad (3)$$

Note: \bar{X} means NOT gate, $\&$ means AND gate, \mid means OR gate and \oplus means XOR gate

3.2 Comparator

Comparator is built in Quartus and showed in Figure 2.

The comparator takes 2 3-bit inputs A and B. The most significant bits are compared first through logic $A2 \& \overline{B2}$, and this will output 1 if A2 is 1 and B2 is 0. Since these are the most significant bits, if A2 is larger than B2, A is larger than B. If A2 is equal to B2, seconds bits are compared using the same comparison logic. If A1 is equal to B1, then the third bits are compared. By comparing these bits from the most significant bits to the least significant bits, the comparator function is fulfilled.

4 CIRCUIT VALIDATION AND EVALUATION

4.1 Validation Using Quartus

4.1.1 Converter. The converter validation waveform is shown in Figure 3.

Note: A3, A2, A1, and A0 are 4-bits inputs of this converter. A3 turns from 0 (digital low) to 1(digital high) at approximately 4096us. A2_Converted, A1_Converted, and A0_Converted are 3-bits output.

Table 2: Truth table of comparator

A2A1A0	B2B1B0	S	A2A1A0	B2B1B0	S	A2A1A0	B2B1B0	S	A2A1A0	B2B1B0	S
000	000	0	010	000	1	100	000	1	110	000	1
000	001	0	010	001	1	100	001	1	110	001	1
000	010	0	010	010	0	100	010	1	110	010	1
000	011	0	010	011	0	100	011	1	110	011	1
000	100	0	010	100	0	100	100	0	110	100	1
000	101	0	010	101	0	100	101	0	110	101	1
000	110	0	010	110	0	100	110	0	110	110	0
000	111	0	010	111	0	100	111	0	110	111	0
001	000	1	011	000	1	101	000	1	111	000	1
001	001	0	011	001	1	101	001	1	111	001	1
001	010	0	011	010	1	101	010	1	111	010	1
001	011	0	011	011	0	101	011	1	111	011	1
001	100	0	011	100	0	101	100	1	111	100	1
001	101	0	011	101	0	101	101	0	111	101	1
001	110	0	011	110	0	101	110	0	111	110	1
001	111	0	011	111	0	101	111	0	111	111	0

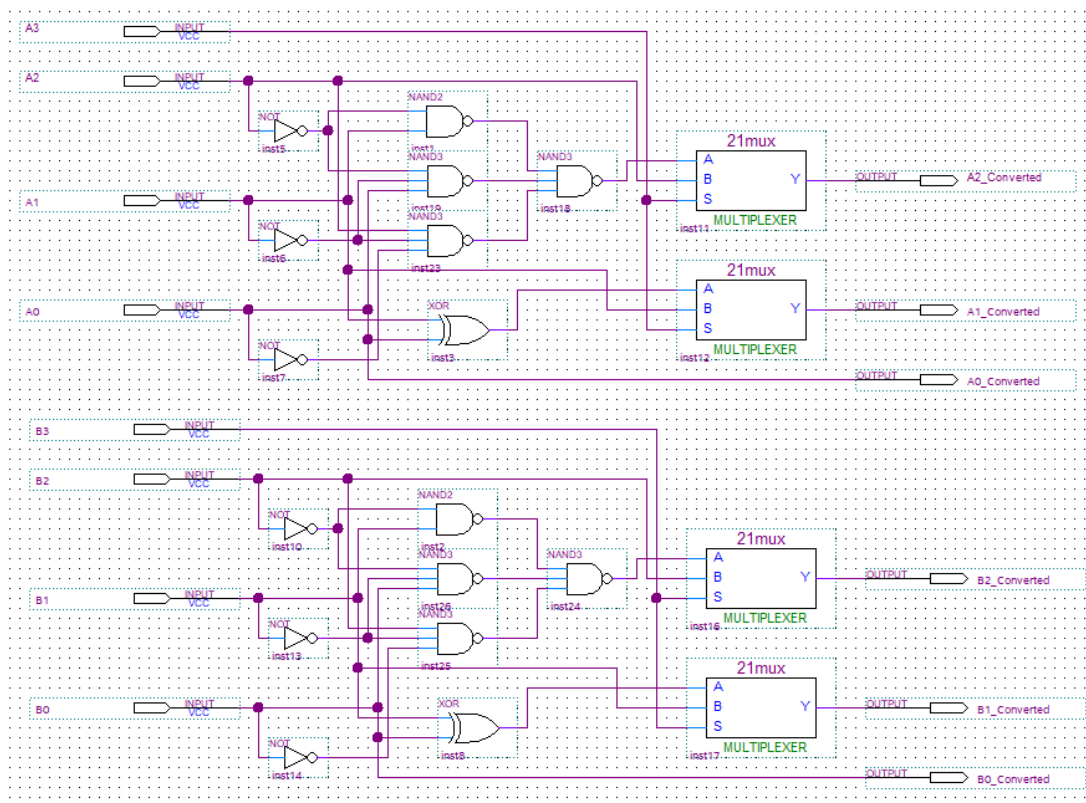


Figure 1: Converter design in Quartus (Photo/Picture credit: Original)

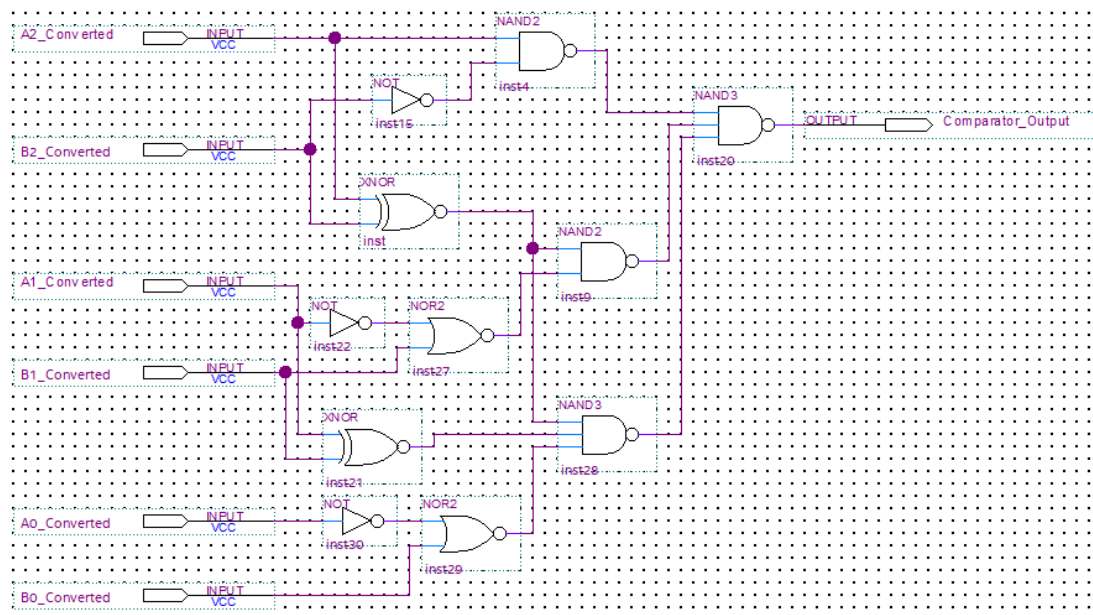


Figure 2: Comparator design in Quartus (Photo/Picture credit: Original)

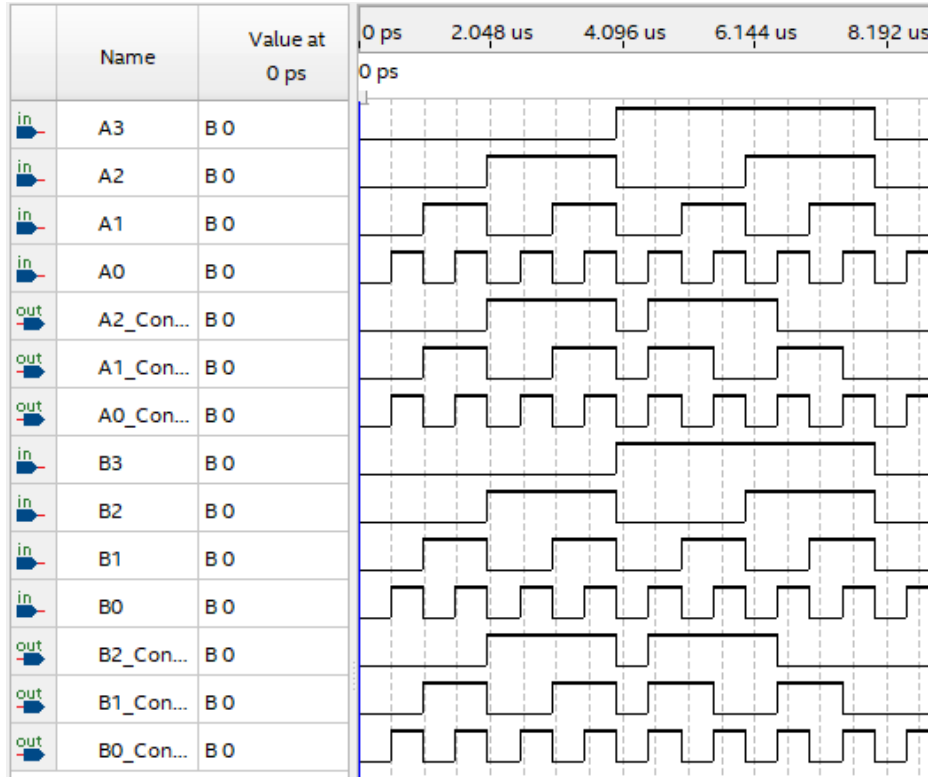


Figure 3: Simulation and validation of converter in Quartus using ModelSim (Photo/Picture credit: Original)

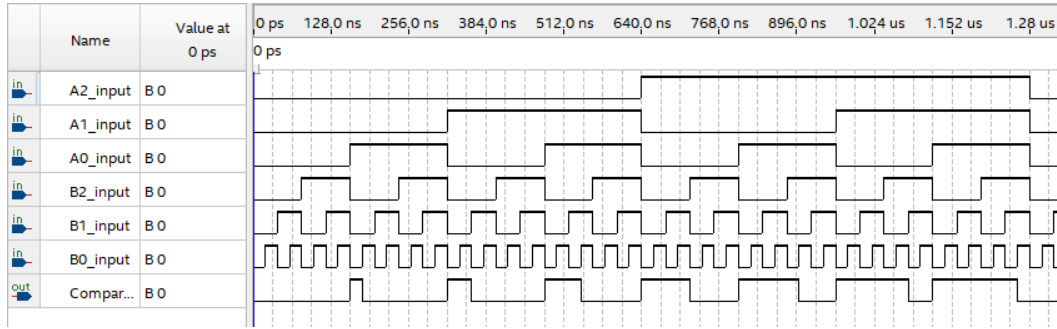


Figure 4: Simulation and validation of comparator in Quartus using ModelSim (Photo/Picture credit: Original)

4.1.2 *Comparator.* The comparator validation waveform is shown in Figure 4.

Note 1: A2_input, A1_input, A0_input and B2_input, B1_input, B0_input are 2 3-bit inputs of this comparator. A2_input turn from 0 (digital low) to 1(digital high) at approximately 640ns. Comparator_Output is the output of this comparator. If Comparator_Output is digital high, A is larger than B, and Comparator_Output is digital low if otherwise. The final circuit is the combination of converter and comparator.

4.2 Minimum Delay and Energy Consumption Calculation

The critical path is found using the method from and shown in Figure 5 [7].

Reference parameters for calculating logical effort and parasitic delay are shown in Table 3.

Input capacitance and Gamma are 1, and load capacitance is 32. On this critical path: (Formulas are referenced from [8].)

$$\text{Total logical effort : } G = \prod g_i = 4 * 2 * 4 * \frac{5}{3} * \frac{5}{3} \approx 133.33 \quad (4)$$

$$\text{Electrical effort } H = \frac{C_{out}}{C_{in}} = 32 \quad (5)$$

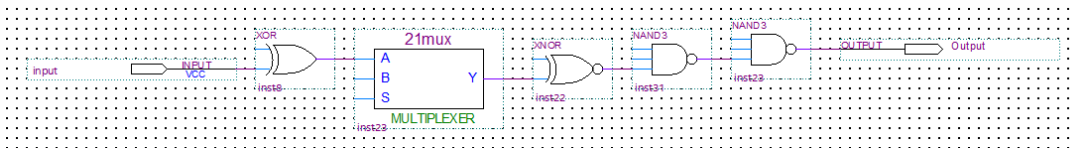


Figure 5: Critical Path in Combined Circuit (Photo/Picture credit: Original)

Table 3: Logical Effort g and Parasitic Delay for Different Logic Gates [8]

Gate Type	g for different numbers of inputs		
	1	2	3
Inverter	1		
NAND		4/3	5/3
NOR		5/3	7/3
Multiplexer		2	2
XOR, XNOR		4	12
Gate Type	Parasitic delay		
Inverter	p_{inv}		
n-input NAND	np_{inv}		
n-input NOR	np_{inv}		
n-way Multiplexer	$2np_{inv}$		
n-input XOR, XNOR	$n2^{n-1}p_{inv}$		

$$\text{number of stages : } N = 5 \quad (6)$$

$$\text{Path effort : } f^N = GH = 133.33 \times 32 = 4266.67 \quad (7)$$

$$f = \sqrt[5]{4266.67} \approx 5.3213 \quad (8)$$

$$\text{Parasitic delay : } \sum P_i = 4 + 4 + 4 + 3 + 3 = 18 \quad (9)$$

$$\text{Size of each stage : } C_i = \frac{C_{i-1}f}{g_{i-1}}, \text{ and } C_1 = 1 \quad (10)$$

$$\text{Fanout of each stage : } F_i = \frac{C_{i+1}}{C_i} \quad (11)$$

$$\text{Approximate delay of each stage : } D_i = \text{Gamma} + F_i \quad (12)$$

$$\text{Energy consumption of each stage : } E_i = C_i \times \text{Gamma} + C_{i+1} \quad (13)$$

$$\text{Total delay : } D = \sum D_i \quad (14)$$

$$\text{Total energy consumption } E = \sum E_i \quad (15)$$

After parameters of each stage are calculated, data of each stage is listed below in Table 4.

Energy initial = $\sum \text{Energy} = 82.22$, Delay initial = $\sum \text{Delay} = 24.44$. This Energy consumption is when delay is at minimum delay.

4.3 Optimization

Since energy is given by $E \propto CV_{dd}^2$ (16) and $E = \sum E_i$, energy consumption calculation becomes to below formula [9].

$$\text{Energy consumption : } E \propto \sum E_i V_{dd}^2 \quad (16)$$

If assume

$$\text{Delay} \propto \frac{V_{dd}}{(V_{dd} - V_t)^2}, \text{ where } V_t = 0.2V \quad (17)$$

Then

$$\text{Delay : } D = \sum D_i \frac{V_{dd}}{(V_{dd} - 0.2)^2} \quad (18)$$

After logging these formulas and values into Excel, excel built-in solver function can be used to find minimum energy consumption by changing sizes and voltage supply and limiting delay to 1.5 minimum delays [10]. Adjusted size, Vdd, energy consumption, and delay are shown in Table 5.

Energy Consumption = $\sum \text{Energy} = 35.73$, Total Delay = $\sum \text{Delay} = 36.66$. This Energy consumption is when delay is at 1.5*minimum delay. Energy Consumption Compared with energy consumption at the minimum delay and initial Vdd, energy consumption is decreased by 56.54% and delay is increased by 50%.

5 CONCLUSION

The density of transistors on microchips had skyrocketed. With more and more transistors, the power consumption of these chips is becoming much more significant than previous. In this design, circuit parameters are leveraged to reduce energy consumption by 56.54%. In this approach, a circuit is drawn to achieve desired objectives. Without changing this circuit design, a tradeoff between energy consumption and delay is proposed and found. By relaxing the delay from a minimum delay to a 1.5 minimum delay, energy consumption is reduced by a significant amount. This design approach shows how optimization works in a computer at the logical gate level. The traditional method of reducing energy consumption is to make the transistor smaller. The optimization process conducted in this experiment had shown an alternate solution to reduce

Table 4: Initial Energy Consumption and Delay

Stage	Size	Fanout	Delay	Energy	Logical Effort (g)	Vdd
1	1	1.33	2.33	2.33	4	1
2	1.33	2.66	3.66	4.87	2	
3	3.54	1.33	2.33	8.25	4	
4	4.71	3.19	4.19	19.74	1.67	
5	15.03	2.13	3.13	47.03	1.67	
Load	32					

Table 5: Adjusted size of each stage and voltage for minimum energy consumption at 1.5 minimum delay

Stage	Size	Fanout	Delay	Energy	Logical Effort (g)	Vdd
1	1	0.54	4.54	1.54	4	0.8356308
2	0.81	1.17	5.17	1.17	2	
3	1.02	1.91	5.91	1.84	4	
4	1.87	3.32	6.32	5.21	1.67	
5	5.39	7.99	10.99	36.01	1.67	
Load	32					

energy consumption without changing the existing circuit design and without scaling down or reducing the number of transistors.

This approach represents one of the many efforts in improving chip efficiency. The advent of new materials could change the whole chip industry. Furthermore, researchers are investing their effort in biochips. The combination of exploration in different paths will further help to reduce energy consumption while maintaining an acceptable delay.

REFERENCES

- [1] BITS & CHIPS. ANALYSIS Chip shrinking in the era of system scaling. Retrieved July 13, 2023 from <https://bits-chips.nl/artikel/chip-shrinking-in-the-era-of-system-scaling/>
- [2] Schulz, M. 1999. The end of the road for silicon? *Nature* 399, (June 1999), 729–730. <https://doi.org/10.1038/21526>
- [3] CIRCUIT DIGEST. Understanding the Difference Between BJT and MOSFET and How to Select the Right One for Your Designs. Retrieved July 13, 2023 from [https://circuitdigest.com/article/understanding-the-difference-between-bjt-and-mosfet-and-how-to-select-the-right-one-for-your-designs#:~:text=\\$BJTs%20have%20switching%20frequencies%20of,power%20loss%2C%20MOSFET%20is%20preferred](https://circuitdigest.com/article/understanding-the-difference-between-bjt-and-mosfet-and-how-to-select-the-right-one-for-your-designs#:~:text=$BJTs%20have%20switching%20frequencies%20of,power%20loss%2C%20MOSFET%20is%20preferred)
- [4] Thomas Finley. 2000. Two's Complement. Retrieved July 13, 2023 from <https://www.cs.cornell.edu/~simstomf/notes/cps104/twoscomp.html>
- [5] J. Eldon Whitesitt. 2010. Boolean Algebra and Its Applications.
- [6] Robert G. Plantz. 2021. Introduction to Computer Organization: ARM Assembly Language Using the Raspberry Pi.
- [7] M. Abramovici, P. R. Menon, D. T. Miller. 1983. CRITICAL PATH TRACING - AN ALTERNATIVE TO FAULT SIMULATION. In *Proceedings of the IEEE 20th Design Automation Conference Proceedings*. IEEE. <https://ieeexplore.ieee.org/document/1585651/references#references>
- [8] Ivan Sutherland, Robert F. Sproull, David Harris. 1999. Logical Effort: Designing Fast CMOS Circuits.
- [9] University of California, Berkeley. digital Integrated Digital Integrated Circuits. Retrieved July 13, 2023 from http://bwrce.eecs.berkeley.edu/Classes/icdesign/ee141_f10/Lectures/Lecture12-Delay_Power-6up.pdf
- [10] Microsoft. Define and solve a problem by using Solver. Retrieved July 13, 2023 from <https://support.microsoft.com/en-gb/office/define-and-solve-a-problem-by-using-solver-5d1a388f-079d-43ac-a7eb-f63e45925040>.