Princess Sumaya University for Technology

King Abdullah II Faculty of Engineering

Electrical Engineering Department

Princess Sumaya جامعـــة
University الأميـــرة سميّـــة
for Technology للتكنولوجيا

DIGITAL ELECTRONICS 21332
FINAL PROJECT

Authors:

Kareem AbuSharkh 20230995 Computer Engineering

Nooreldeen Alkomi 20230632 Computer Engineering

Supervisor:

Dr. Mohammad Abdel-Majeed

January 12, 2026

Abstract

This report delves into the design and implementation of a 2-bit adder/subtractor using custom transistor-level logic modules. A 2-bit adder/subtractor is a fundamental digital circuit that processes two 2-bit binary inputs, A_1A_0 and B_1B_0, along with a 1-bit Mode input, to produce a 3-bit output denoted as ($S_2S_1S_0$). The design process involves developing the logic for a 1-bit adder module, constructed primarily using XOR and NAND gate modules implemented at the transistor level. These transistors are specifically sized to ensure equal rise and fall delays, optimizing the circuit for speed. The report outlines the step-by-step construction and verification process, beginning with the creation of these optimized gate modules and then cascading the 1-bit adder units to implement the full logical structure of the 2-bit adder/subtractor.

TABLE OF CONTENTS

1 Introduction 2
1.1 Objectives 2
1.2 Theory 2
1.2.1 Sum/Difference 2
1.2.2 Carry-out/Borrow-out 3
2 Procedure and Methods 3
2.1 Designing 1-bit XOR module 3
2.2 Designing 1-bit NAND module 4
2.3 Designing 1-bit Full Adder module 5
2.4 Designing 2-bit Adder/Subtractor module 6
3 Results and Discussions 7
4 Conclusions 8
5 References 8

1 INTRODUCTION

In digital electronics, the 2-bit adder/subtractor is a critical building block used in arithmetic operations involving both binary addition and subtraction within various systems. It is designed to process two 2-bit binary numbers (A_1A_0 and B_1B_0) and a Mode control bit, producing a 3-bit result ($S_2S_1S_0$). This report presents the implementation of a 2-bit adder/subtractor using transistor-level design. The primary goal is to demonstrate the creation of the circuit by cascading 1-bit adder modules built from optimized XOR and NAND gates, emphasizing its logical structure and timing efficiency. The design has been verified using input waveforms, and the correctness of the implementation was confirmed by examining the output waveforms for all possible input combinations.

1.1 OBJECTIVES

- **Design and implement a 2-bit Adder/Subtractor at the Transistor Level:** To create a 2-bit adder/subtractor circuit by cascading 1-bit adder modules built from optimized XOR and NAND gates. This involves sizing all transistors to ensure that rise and fall delays are equal while accurately modeling the logical operations for both binary addition and subtraction**.**
- **Develop Comprehensive Input Waveforms for Verification:** To construct a complete set of input waveforms that simulate all possible input combinations for the operands (A_1A_0, B_1B_0) and the Mode bit, ensuring the correct testing of the adder/subtractor design.
- **Analyze Output Waveforms and Timing:** To verify the correctness of the implementation by analyzing the resulting output waveforms, confirming that the logical outputs ($S_2S_1S_0$) match expected values and that the transistor sizing has successfully achieved the required delay balancing.

1.2 THEORY

A 2-bit adder/subtractor is a versatile digital circuit used to perform both binary addition and subtraction, which are essential operations in arithmetic logic units (ALUs). The design is constructed by cascading 1-bit Full Adder modules. Each 1-bit module processes three inputs: the significant bits A and B, and a Carry-in (Cin) from the previous stage (or the Mode bit for the first stage). It produces two outputs: a Sum bit (S) and a Carry-out bit (Cout). By manipulating the B input using the Mode signal (typically via XOR logic), the same circuit can perform subtraction using 2's complement logic.

The logical operations of the core 1-bit adder module can be described using the following equations:

1.2.1 Sum/Difference

The Sum/Difference bit is obtained by performing an XOR operation on the two input bits (A and the modified B) and the Carry-in bit.

$$S(\text{or } D) = A \oplus B \oplus Cin$$

1.2.2 Carry-out/Borrow-out

The Carry-out/Borrow-out represents the overflow logic, generated when the combination of inputs produces a value exceeding a single bit. Using our specific logic gates (XOR/NAND), this is logically equivalent to:

$$\text{Cout} = \neg\left(\neg(A \cdot B) \cdot \neg((A \oplus \text{B}) \cdot Cin)\right)$$
$$\text{Bout} = (\neg A \cdot B) + (\neg A \cdot Bin) + (B \cdot Bin)$$

2 PROCEDURE AND METHODS

Designing the 2-bit adder/subtractor consists of using optimized XOR and NAND gate modules to construct the underlying 1-bit full adder logic. To implement the adder/subtractor in this manner, a transistor-level structural approach is utilized, where custom logic modules must be designed, XOR and NAND gates.

2.1 DESIGNING 1-BIT XOR MODULE

The XOR module is constructed at the transistor level, serving as the primary component for Sum generation and controlled inversion logic. The design process is based on precise calculations for transistor aspect ratios (W/L) to balance the pull-up and pull-down networks, with the reference width for the nmos is 0.1µ.

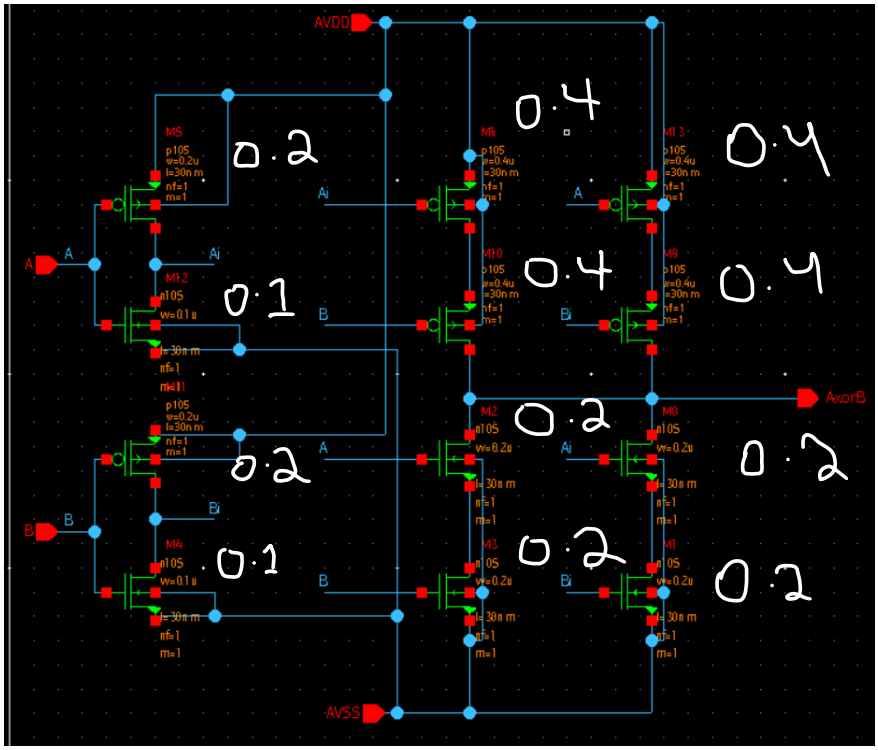


Figure 1: XOR Gate-Transistor level

The XOR that is shown in Figure 1, consists of pmos and nmos, where the reference is 0.1 µ for nmos, the nmos of PDN are set to 0.2 µ; where the longest path have two nmos in series (0.1 * 2 = 0.2). For the pmos of the PUN, they are set to 0.4 µ; where the longest path has two pmos in series, and due to the difference of the mobility, the overall sizing must be double the nmos overall sizing, which is 0.1*2 = 0.2, therefore the size of each pmos is 0.4 (0.2 * 2 = 0.4).

To check our calculations and the overall design, we used PrimeWave as a TestBench and here are the results:

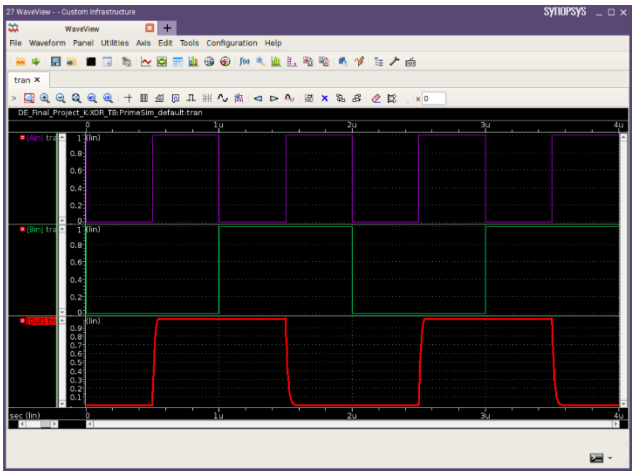


Figure 2: XOR TestBench

2.2 Designing 1-bit NAND module

The NAND module is implemented at the transistor level to efficiently handle the critical carry generation and propagation logic. The design procedure relies on precise transistor aspect ratio (W/L) calculations to equalize the pull-up and pull-down network delays, utilizing a standard reference NMOS width of 0.1µ.

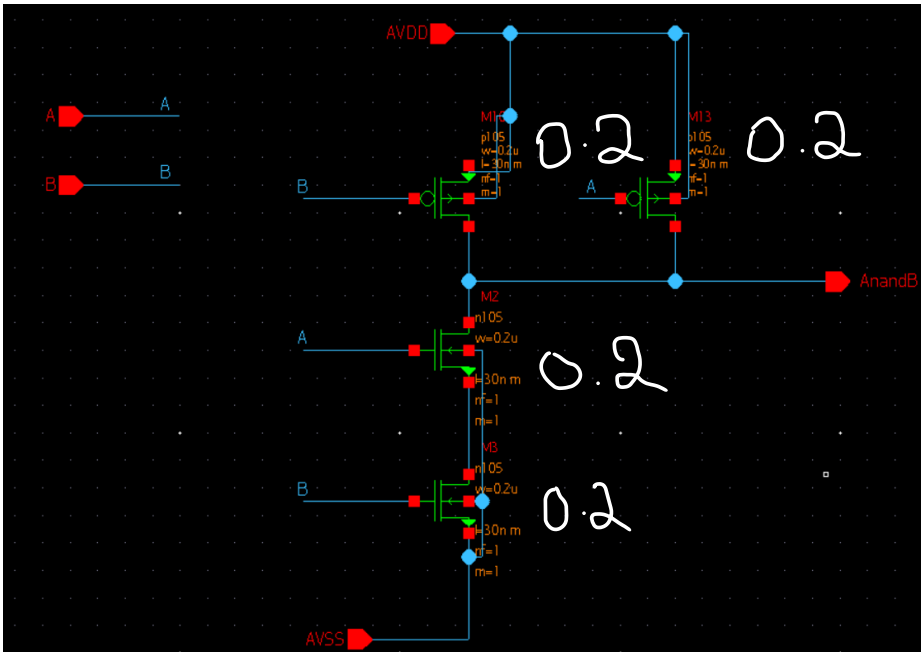


Figure 3: NAND Gate-Transistor level

The XOR that is shown in Figure 1, consists of pmos and nmos, where the reference is 0.1 µ for nmos, the nmos of PDN are set to 0.2 µ; where the longest path have two nmos in series (0.1 * 2 = 0.2). For the pmos of the PUN, they are set to 0.2 µ; where the longest path has only one pmos, and due to the difference of the mobility, the overall sizing must be double the nmos overall sizing, which is 0.1*2 = 0.2, therefore the size of each pmos is 0.2 (0.2 * 1 = 0.2).

To check our calculations and the overall design, we used PrimeWave as a TestBench and here are the results:

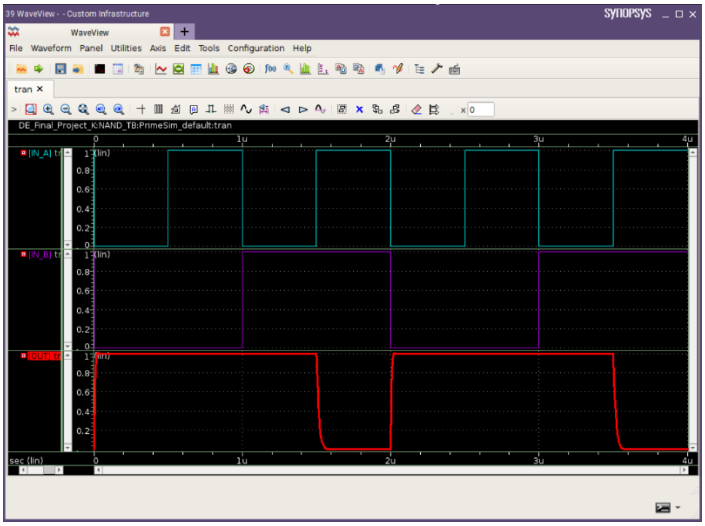


Figure 4: NAND TestBench

2.3 Designing 1-bit Full Adder module

The 1-bit Full Adder module is constructed by interconnecting the optimized XOR and NAND logic blocks to realize the arithmetic Sum and Carry-out functions. This modular design serves as the fundamental cascading unit, enabling the scalable construction of the complete 2-bit adder/subtractor system.

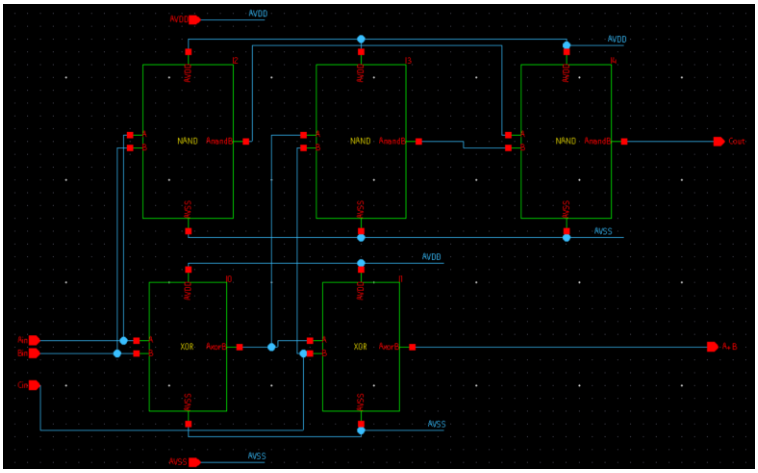


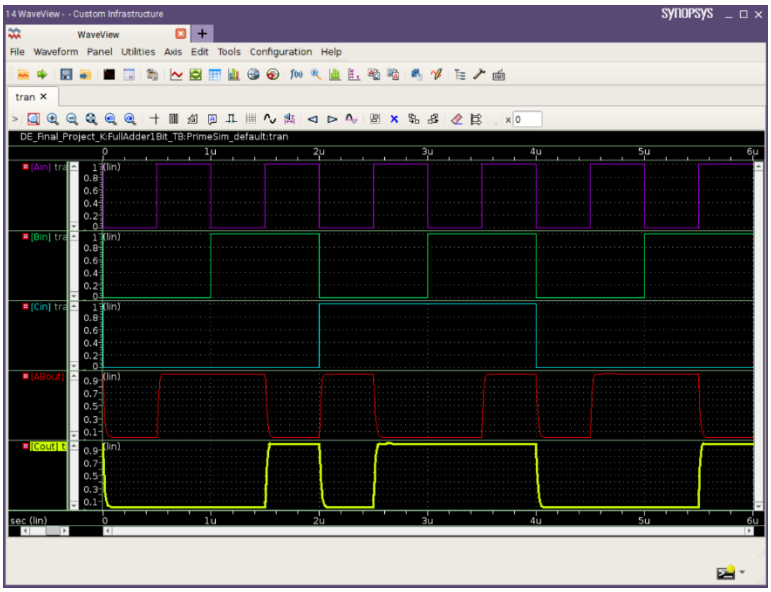
Figure 5: Full Adder – Gate Circuit

The full adder produces two values, Sout and Cout, these values come from the following equations:

$$S = A \oplus B \oplus Cin$$

$$Cout = \neg\,(\neg(A \cdot B) \cdot \neg((A \oplus B) \cdot Cin))$$

To check our design for the adding functionality, we used PrimeWave as a TestBench and here are the results:



Inputs			Outputs	
Cin	**B**	**A**	**Sout**	**Cout**
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

Figure 6: Full Adder TestBench, Truth Table for Full Adder

2.4 Designing 2-bit Adder/Subtractor module

The complete 2-bit adder/subtractor is realized by cascading two 1-bit Full Adder instances in a ripple-carry configuration. The design integrates the Mode control input to dynamically invert the operands, allowing the circuit to seamlessly compute the 3-bit result ($S_2S_1S_0$) for both addition and subtraction operations.

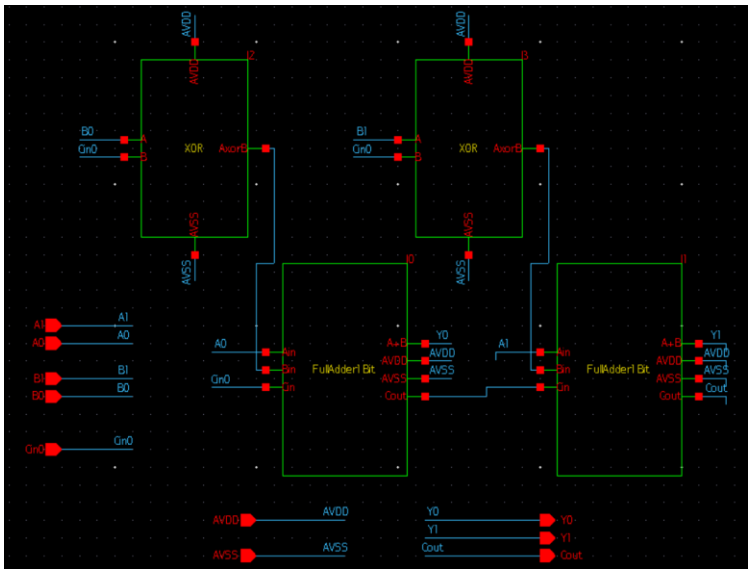


Figure 7: 2-Bit Adder/Subtractor – Gate Circuit

To check our design functionality, we used PrimeWave as a TestBench and here are the results:

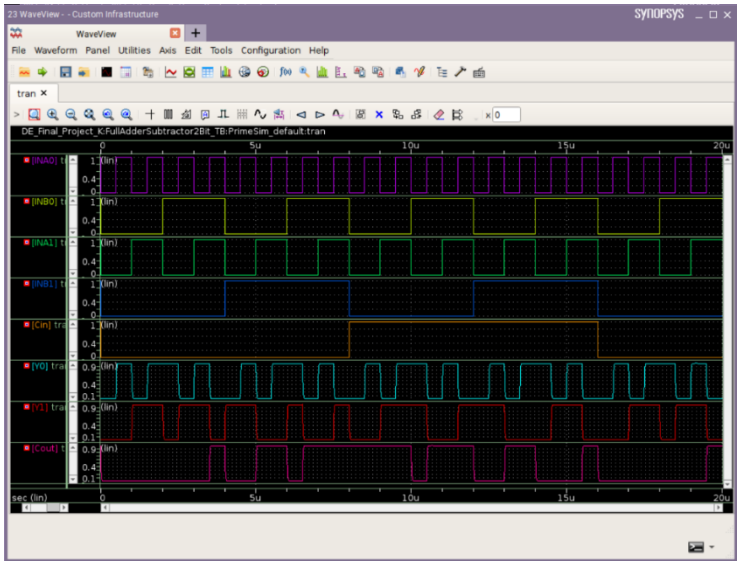


Figure 7: 2-Bit Adder/Subtractor TestBench

3 RESULTS AND DISCUSSIONS

The 2-bit adder/subtractor was analyzed using 1:1(bottom) and 1:2(top) transistor sizing. The 1:1 sizing shows slower rise/fall times and higher propagation delay. In contrast, 1:2 sizing improves switching speed and carry propagation, resulting in faster output transitions. Although 1:2 sizing increases area and power consumption, it provides better overall performance for high-speed operation.

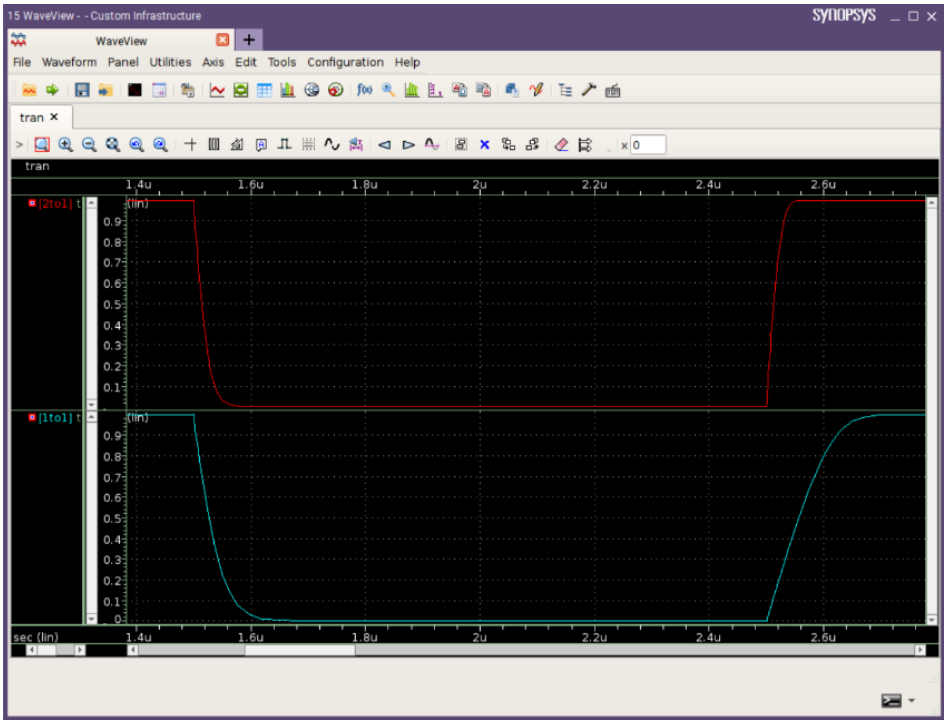


Figure 7: 2-Bit Adder/Subtractor TestBench

4 CONCLUSIONS

In conclusion, this project successfully demonstrated the design and implementation of a 2-bit adder/subtractor using a transistor-level structural approach. By developing custom XOR and NAND modules with precisely calculated aspect ratios, the design achieved the critical requirement of equalized rise and fall delays. The construction of the final circuit, achieved by cascading 1-bit Full Adder units, allowed for the accurate processing of 2-bit binary inputs and the Mode control signal. The simulation results, verified through comprehensive waveform analysis, confirmed that the circuit correctly produces the 3-bit output ($S_2S_1S_0$) for all input combinations, validating both the logical correctness and the timing optimization of the design.

5 REFERENCES

[1] Fundamentals of Microelectronics 2nd edition Sedra/Smith, Microelectronics Circuits, Edition 5th or 6th DeMassa & Ciccone, Digital Integrated Circuits

[2] EE 421L Digital Integrated Circuit Design Laboratory Fall 2014, University of Nevada, Las Vegas

[3] Course Slides