4-bit Comparator

I. INTRODUCTION

The objective of this project is to design and implement a 4-bit digital comparator, a fundamental combinational logic circuit that takes two 4-bit binary inputs and determines their relative magnitude, outputting signals to indicate whether the first input is greater than, less than, or equal to the second input. The design involves logic gates to create the necessary circuitry based on the truth table and Boolean expressions, enabling accurate comparison of 4-bit binary numbers, a crucial functionality in various computing and electronic systems, and providing valuable learning experience in digital logic design and combinational circuit implementation.

II. BOOLEAN EXPRESSIONS

The design of a 4-bit comparator circuit requires three Boolean expressions to represent the desired outputs, utilizing the logical operations of AND, OR, and XNOR. The three output expressions for a 4-bit comparator circuit are as follows: the "greater than" output, the "less than" output, and the "equal to" output. These outputs are represented by Boolean expressions that leverage the logical operations of AND, OR, and XNOR, applied to the corresponding bits of the input operands. The expressions are structured to capture the desired comparative relationships between the 8 input values, from the most significant bit to the least significant bit:

A. The initial output will be represented by the subsequent Boolean expression; it will only transition to a high state when the first 4-bit binary number exceeds the value of the second 4-bit binary number:

$$A_3B_3' + A_2B_2'(A_3 \oplus B_3)' + A_1B_1'(A_3 \oplus B_3)'(A_2 \oplus B_2)' + A_0B_0'(A_3 \oplus B_3)'(A_2 \oplus B_2)'(A_1 \oplus B_1)'$$

B. The secondary output will be represented by the subsequent Boolean expression; it will only generate a high signal when both 4-bit binary numbers are identical:

$$(A_3 \oplus B_3)'(A_2 \oplus B_2)'(A_1 \oplus B_1)'(A_0 \oplus B_0)'$$

C. Regarding the final output, it will transmit a high signal exclusively when the initial 4-bit binary number is numerically inferior to the other. Notably, there exist two distinct methods for formulating the Boolean expression representing the outcome of this condition:

- The first approach involves formulating a general expression that represents the desired output without considering the preceding two outputs. This expression can be expressed as follows:

$$A_3'B_3 + A_2'B_2(A_3 \oplus B_3)' + A_1'B_1(A_3 \oplus B_3)'(A_2 \oplus B_2)' + A_0'B_0(A_3 \oplus B_3)'(A_2 \oplus B_2)'(A_1 \oplus B_1)'$$

- The alternative method involves negating the final two outputs and performing an AND operation on the negated values. While the written expression may be lengthier compared to the first approach, the actual circuit design becomes more simplified, as demonstrated in the subsequent design section:

$$[A_3B_3' + A_2B_2'(A_3 \oplus B_3)' + A_1B_1'(A_3 \oplus B_3)'(A_2 \oplus B_2)' + A_0B_0'(A_3 \oplus B_3)'(A_2 \oplus B_2)'(A_1 \oplus B_1)']'$$
$$[(A_3 \oplus B_3)'(A_2 \oplus B_2)'(A_1 \oplus B_1)'(A_0 \oplus B_0)']$$

III. TRUTH TABLE

Given the circuit design with 8 inputs, generating a comprehensive truth table to represent all possible input-output combinations would be an arduous task. Such a truth table would require 11 columns (8 inputs, 3 outputs) and 256 rows to account for the full range of input permutations. Instead, a sample truth table will be provided to illustrate the relationship between the inputs and their corresponding outputs. This approach offers a practical alternative to documenting the complete input-output mapping, while still conveying the essential characteristics of the circuit's behavior.

A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	A>B	A=B	A<B
0	1	1	1	0	1	0	0	1	0	0
1	0	0	1	0	0	1	1	1	0	0
1	1	0	1	1	0	1	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0
1	0	1	0	1	0	1	0	0	1	0
1	1	1	1	1	1	1	1	0	1	0
0	1	0	0	0	1	1	1	0	0	1
0	0	1	1	1	0	0	1	0	0	1
1	0	1	0	1	1	0	1	0	0	1

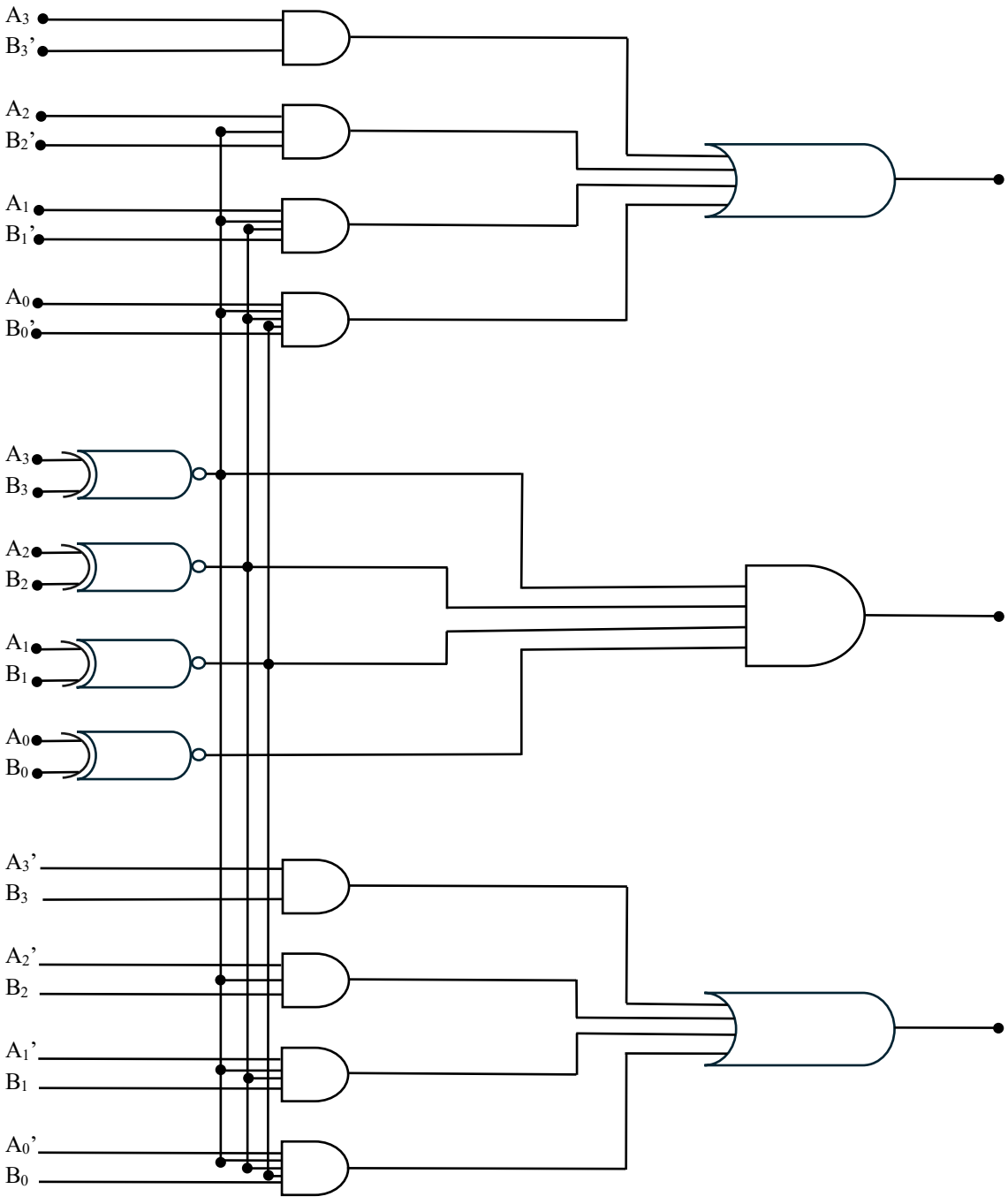
The truth table presented above demonstrated a sample set of 8 input conditions and their corresponding 3 output states. Rather than enumerating all 256 possible input combinations, this selective approach effectively captures the essential relationships between the input variables and output signals. Specifically, the first row illustrates that when the binary-encoded input A represents the decimal value 7, and input B represents the decimal value 4, the output A>B is high while the other outputs A=B and A<B are low. This aligns with the expectation that when A is greater than B, the appropriate comparison output should be activated. The subsequent two rows exhibit a similar pattern, where the greater-than relationship between the input values (9 vs. 3, and 13 vs. 10) is correctly reflected in the high state of the A>B output. The middle three rows demonstrate the case where the input values A and B are equal, resulting in a high output for A=B while the other comparison outputs remain low. Finally, the last three rows depict the scenario where the value of A is less than the value of B, appropriately triggering the high state of the A<B output while the other outputs remain low. Based on this analysis, a more general truth table can be constructed to abstractly represent the relationship between the input variables A and B, and their corresponding comparison outputs:

A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

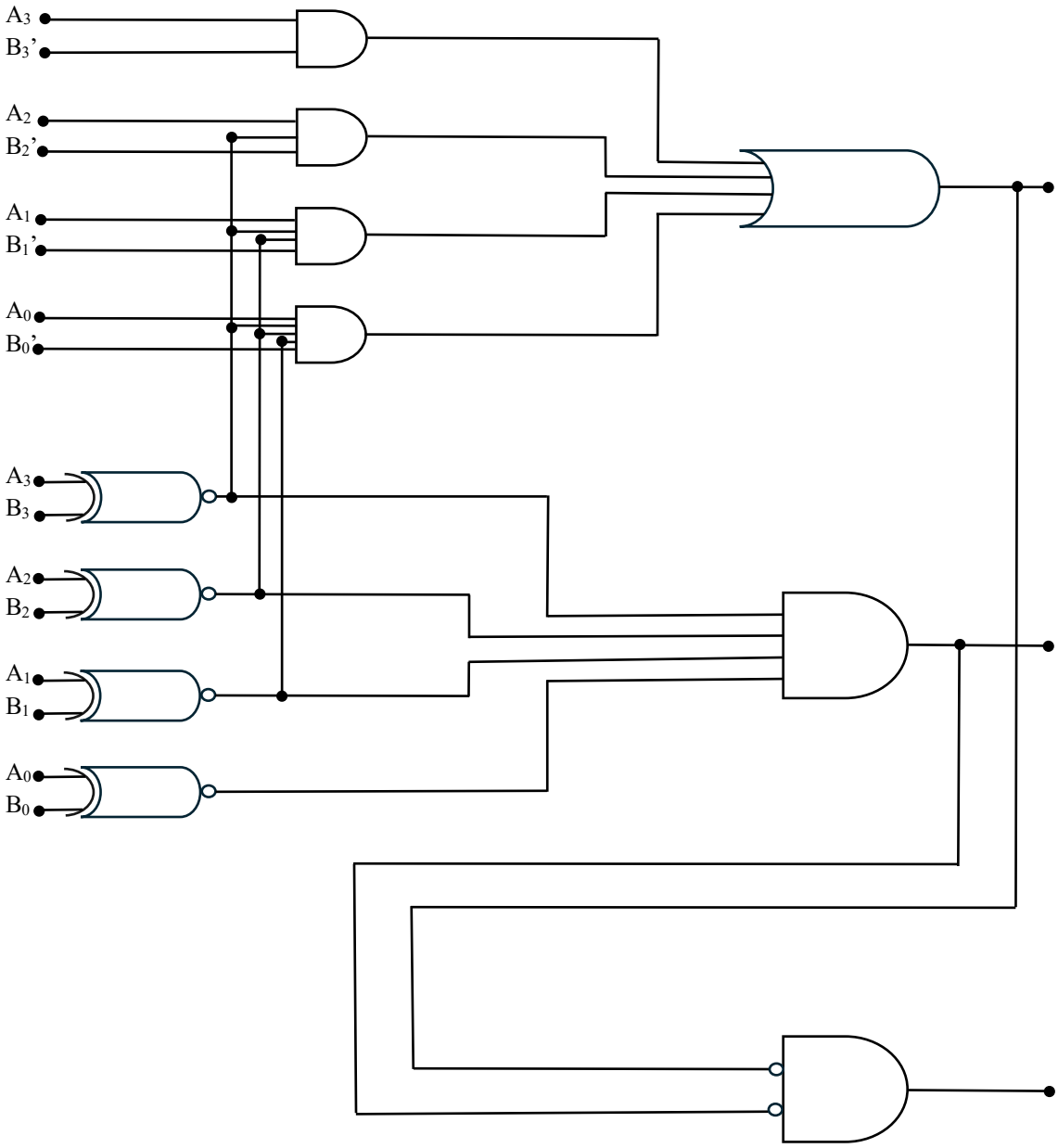
This generalized truth table, presented last, offers a concise and comprehensive overview of the circuit's functional characteristics. It eliminates the need to list all possible input combinations or provide samples of them. This abstract representation facilitates a clear understanding of the underlying logic governing the circuit's operation.

IV. DESIGN

In this section, a schematic diagram of the 4-bit magnitude comparator circuit is going to be presented, illustrating its precise operation in generating an accurate result from two 4-bit binary number inputs. The total input will consist of 8 bits, while the output will comprise three bits, with only one of the three bits being high at any given time. Consequently, the output will be either 100, 010, or 001:

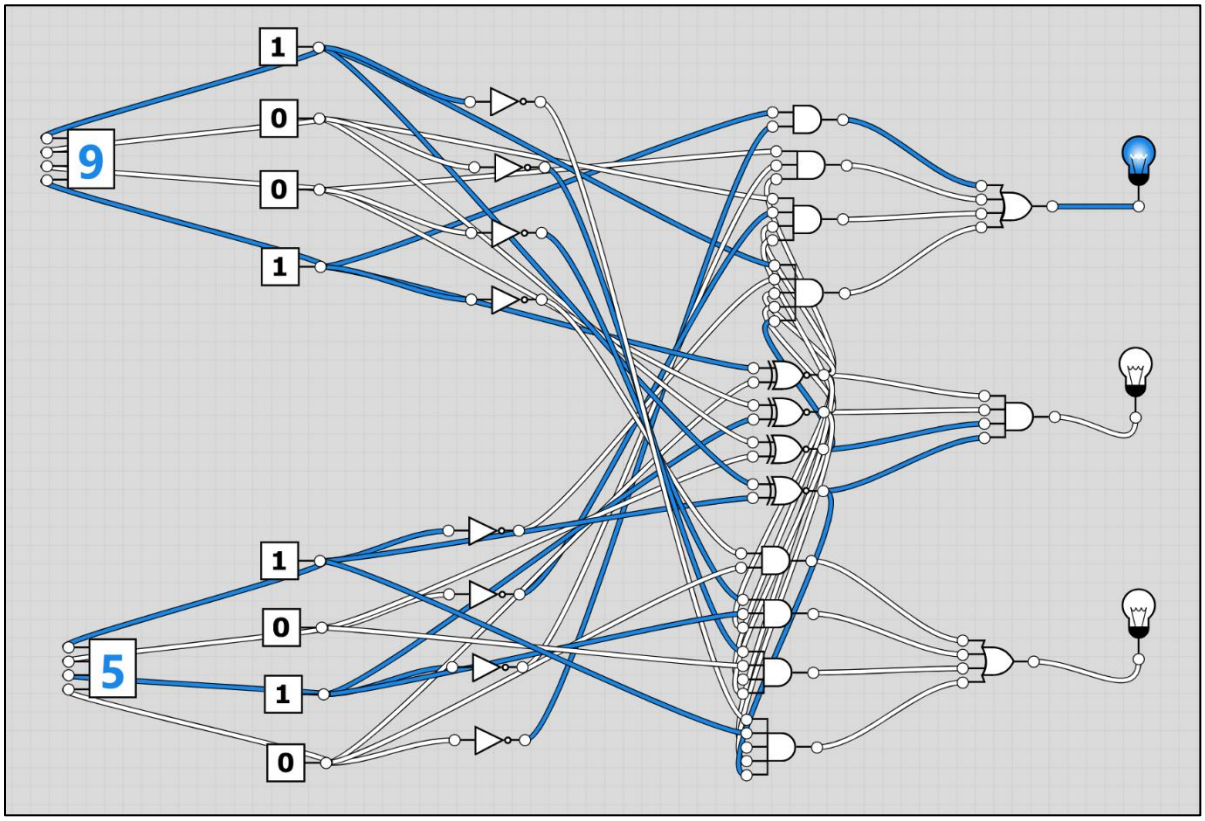


The previous schematic of the circuit provides accurate outputs; however, it can be further simplified while maintaining the same accurate results. This simplified version of the circuit schematic would be more efficient and potentially more cost-effective, making it a desirable alternative to the original design:



V. SIMULATION

In the provided circuit simulation scenario with A = 9 and B = 5, the A>B output is activated as expected, since 9 is greater than 5. This observed behavior aligns with the underlying logic depicted in the circuit's truth table, validating the design and implementation.



VI. CONCLUSION

The 4-bit comparator circuit is a device that utilizes a combination of AND, OR, and XNOR gates to generate output signals that indicate the relative magnitude of two 4-bit binary numbers. This design efficiently executes the comparison function, enabling effective processing and analysis of digital data.

VII. Bibliography

i. M. M. Mano, M. D. Ciletti, Digital Design (Forth Edition), Prentice-Hall 2020

ii. Logic.Ly, logic.ly/demo/. Accessed 1 May 2024.

iii. Sayed, J.F. (2022) *A DIGITAL 8-BIT MAGNITUDE COMPARATOR USING 1-BIT MAGNITUDE COMPARATOR BLOCK*, *ResearchGate*. Available at: https://www.researchgate.net/publication/364928446_A_DIGITAL_8-BIT_MAGNITUDE_COMPARATOR_USING_1-BIT_MAGNITUDE_COMPARATOR_BLOCK (Accessed: 02 May 2024).