DLD PROJECT REPORT

Group Members

M. Arham (BSCE22007)

Ahmad Waleed Akhtar (BSCE22003)

Title

4-bit multiplier using full adders.

Abstract

The project is about making a 4-bit multiplier using full- Adders and AND-gates.

Introduction

This project focuses on creating a 4-bit multiplier using Integrated Circuits (ICs), a crucial element in digital systems for efficient binary multiplication. Through the use of ICs, known for their modularity and ease of integration, we aim to demonstrate practical applications of digital logic. The project involves selecting suitable ICs, designing the circuit, and implementing it on a prototyping board. This hands-on experience in digital circuit design will showcase the significance of efficient multipliers in various digital applications, laying the foundation for understanding advanced arithmetic units and digital system design principles.

Components Required

1. 4-bit adder 7483 IC (x3)
2. 2 input and gate 7408 IC(x4)
3. Wires
4. Veroboard
5. LEDs
6. 2 pin connectors
7. Push buttons

Truth Table

0000	0000	00000000
0000	0001	00000000
0000	0010	00000000
0000	0011	00000000
0000	0100	00000000
0000	0101	00000000
0000	0110	00000000
0000	0111	00000000
0000	1000	00000000
0000	1001	00000000
0000	1010	00000000
0000	1011	00000000
0000	1100	00000000
0000	1101	00000000
0000	1110	00000000
0000	1111	00000000
0001	0000	00000000
0001	0001	00000001
0001	0010	00000010

0001	0011	00000011
0001	0100	00000100
0001	0101	00000101
0001	0110	00000110
0001	0111	00000111
0001	1000	00001000
0001	1001	00001001
0001	1010	00001010
0001	1011	00001011
0001	1100	00001100
0001	1101	00001101
0001	1110	00001110
0001	1111	00001111
0010	0000	00000000
0010	0001	00000010
0010	0010	00000100
0010	0011	00000110
0010	0100	00001000
0010	0101	00001010
0010	0110	00001100

0010	0111	00001110
0010	1000	00010000
0010	1001	00010010
0010	1010	00010100
0010	1011	00010110
0010	1100	00011000
0010	1101	00011010
0010	1110	00011100
0010	1111	00011110
0011	0000	00000000
0011	0001	00000011
0011	0010	00000110
0011	0011	00001001
0011	0100	00001100
0011	0101	00001111
0011	0110	00010010
0011	0111	00010101
0011	1000	00011000
0011	1001	00011011
0011	1010	00011110

0011	1011	00100001
0011	1100	00100100
0011	1101	00100111
0011	1110	00101010
0011	1111	00101101
0100	0000	00000000
0100	0001	00000100
0100	0010	00001000
0100	0011	00001100

0100	0100	00010000
0100	0101	00010100
0100	0110	00011000
0100	0111	00011100
0100	1000	00100000
0100	1001	00100100
0100	1010	00101000
0100	1011	00101100
0100	1100	00110000
0100	1101	00110100
0100	1110	00111000

0100	1111	00111100
0101	0000	00000000
0101	0001	00000101
0101	0010	00001010
0101	0011	00001111
0101	0100	00010100
0101	0101	00011001
0101	0110	00011110
0101	0111	00100011
0101	1000	00101000
0101	1001	00101101
0101	1010	00110010
0101	1011	00110111
0101	1100	00111100
0101	1101	01000001
0101	1110	01000110
0101	1111	01001011
0110	0000	00000000
0110	0001	00000110
0110	0010	00001100

0110	0011	00010010
0110	0100	00011000
0110	0101	00011110
0110	0110	00100100
0110	0111	00101010
0110	1000	00110000
0110	1001	00110110
0110	1010	00111100
0110	1011	01000010
0110	1100	01001000
0110	1101	01001110
0110	1110	01010100
0110	1111	01011010
0111	0000	00000000
0111	0001	00000111
0111	0010	00001110
0111	0011	00010101
0111	0100	00011100
0111	0101	00100011
0111	0110	00101010

0111	0111	00110001
0111	1000	00111000
0111	1001	00111111
0111	1010	01000110
0111	1011	01001101
0111	1100	01010100
0111	1101	01011011
0111	1110	01100010
0111	1111	01101001
1000	0000	00000000
1000	0001	00001000
1000	0010	00010000
1000	0011	00011000
1000	0100	00100000
1000	0101	00101000
1000	0110	00110000
1000	0111	00111000
1000	1000	01000000
1000	1001	01001000
1000	1010	01010000

1000	1011	01011000
1000	1100	01100000
1000	1101	01101000
1000	1110	01110000
1000	1111	01111000
1001	0000	00000000
1001	0001	00001001
1001	0010	00010010
1001	0011	00011011
1001	0100	00100100
1001	0101	00101101
1001	0110	00110110
1001	0111	00111111
1001	1000	01001000
1001	1001	01010001
1001	1010	01011010
1001	1011	01100011
1001	1100	01101100
1001	1101	01110101
1001	1110	01111110

1001	1111	10000111
1010	0000	00000000
1010	0001	00001010
1010	0010	00010100
1010	0011	00011110
1010	0100	00101000
1010	0101	00110010
1010	0110	00111100
1010	0111	01000110
1010	1000	01010000
1010	1001	01011010
1010	1010	01100100
1010	1011	01101110
1010	1100	01111000
1010	1101	10000010
1010	1110	10001100
1010	1111	10010110
1011	0000	00000000
1011	0001	00001011
1011	0010	00010110

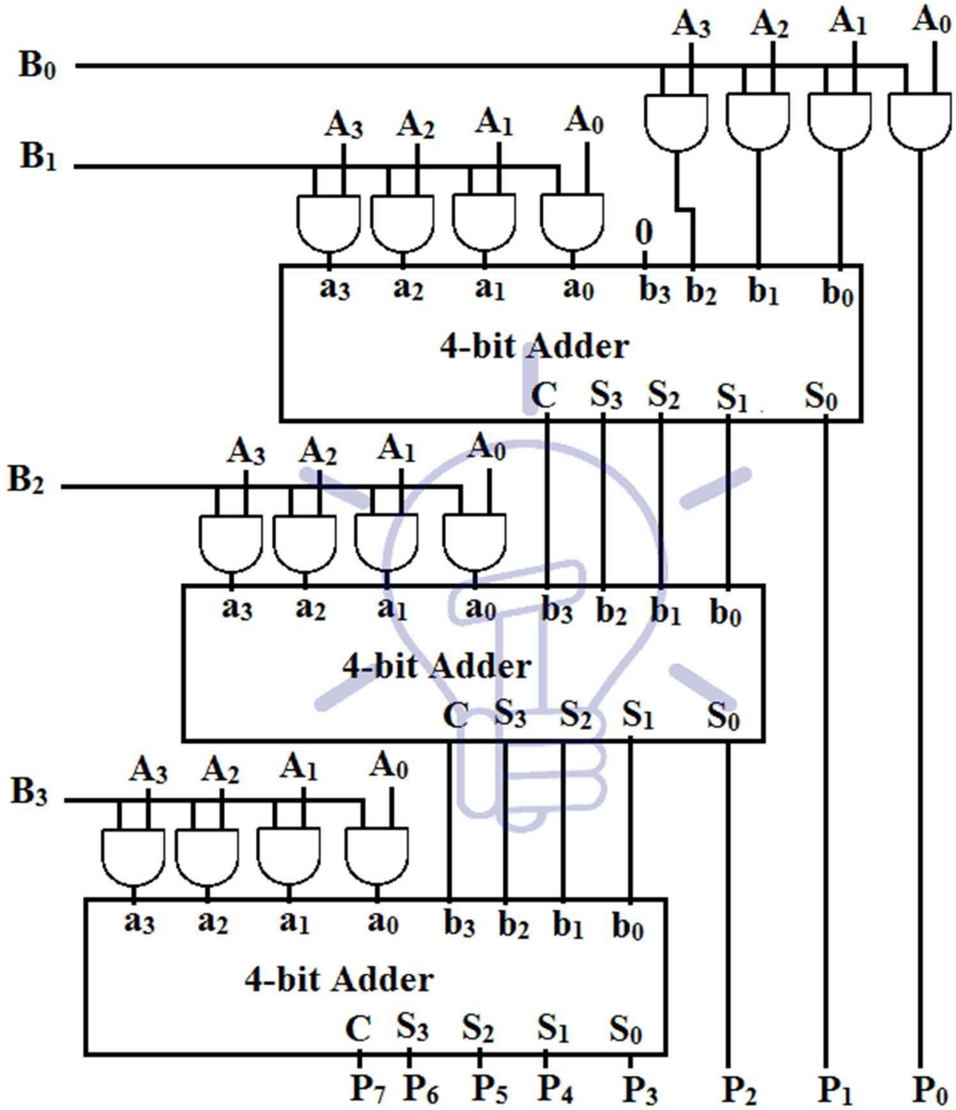
1011	0011	00100001
1011	0100	00101100
1011	0101	00110111
1011	0110	01000010
1011	0111	01001101
1011	1000	01011000
1011	1001	01100011
1011	1010	01101110
1011	1011	01111001
1011	1100	10000100
1011	1101	10001111
1011	1110	10011010
1011	1111	10100101
1100	0000	00000000
1100	0001	00001100
1100	0010	00011000
1100	0011	00100100
1100	0100	00110000
1100	0101	00111100
1100	0110	01001000

1100	0111	01010100
1100	1000	01100000
1100	1001	01101100
1100	1010	01111000
1100	1011	10000100
1100	1100	10010000
1100	1101	10011100
1100	1110	10101000
1100	1111	10110100
1101	0000	00000000
1101	0001	00001101
1101	0010	00011010
1101	0011	00100111
1101	0100	00110100
1101	0101	01000001
1101	0110	01001110
1101	0111	01011011
1101	1000	01101000
1101	1001	01110101
1101	1010	10000010

1101	1011	10001111
1101	1100	10011100
1101	1101	10101001
1101	1110	10110110
1101	1111	11000011
1110	0000	00000000
1110	0001	00001110
1110	0010	00011100
1110	0011	00101010
1110	0100	00111000
1110	0101	01000110
1110	0110	01010100
1110	0111	01100010
1110	1000	01110000
1110	1001	01111110
1110	1010	10001100
1110	1011	10011010
1110	1100	10101000
1110	1101	10110110
1110	1110	11000100

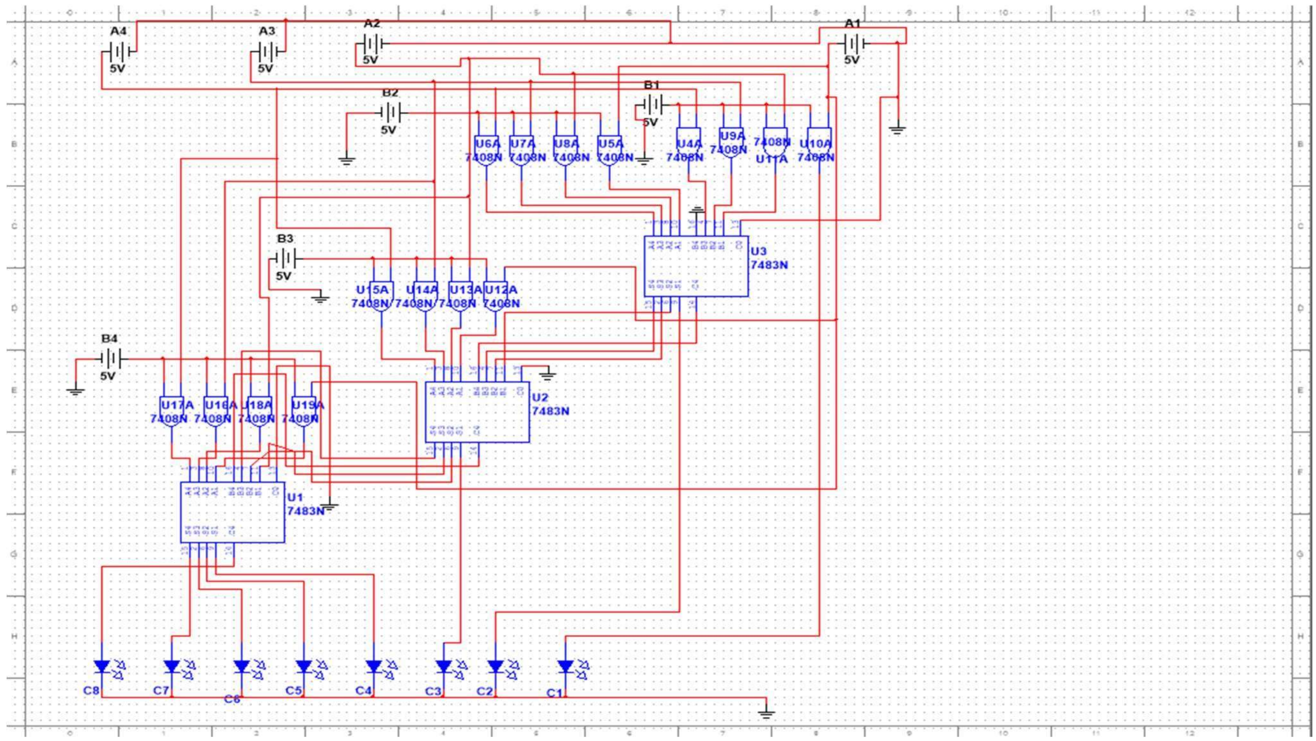
1110	1111	11010010
1111	0000	00000000
1111	0001	00001111
1111	0010	00011110
1111	0011	00101101
1111	0100	00111100
1111	0101	01001011
1111	0110	01011010
1111	0111	01101001
1111	1000	01111000
1111	1001	10000111
1111	1010	10010110
1111	1011	10100101
1111	1100	10110100
1111	1101	11000011
1111	1110	11010010
1111	1111	11100001

Block Diagram of Circuit

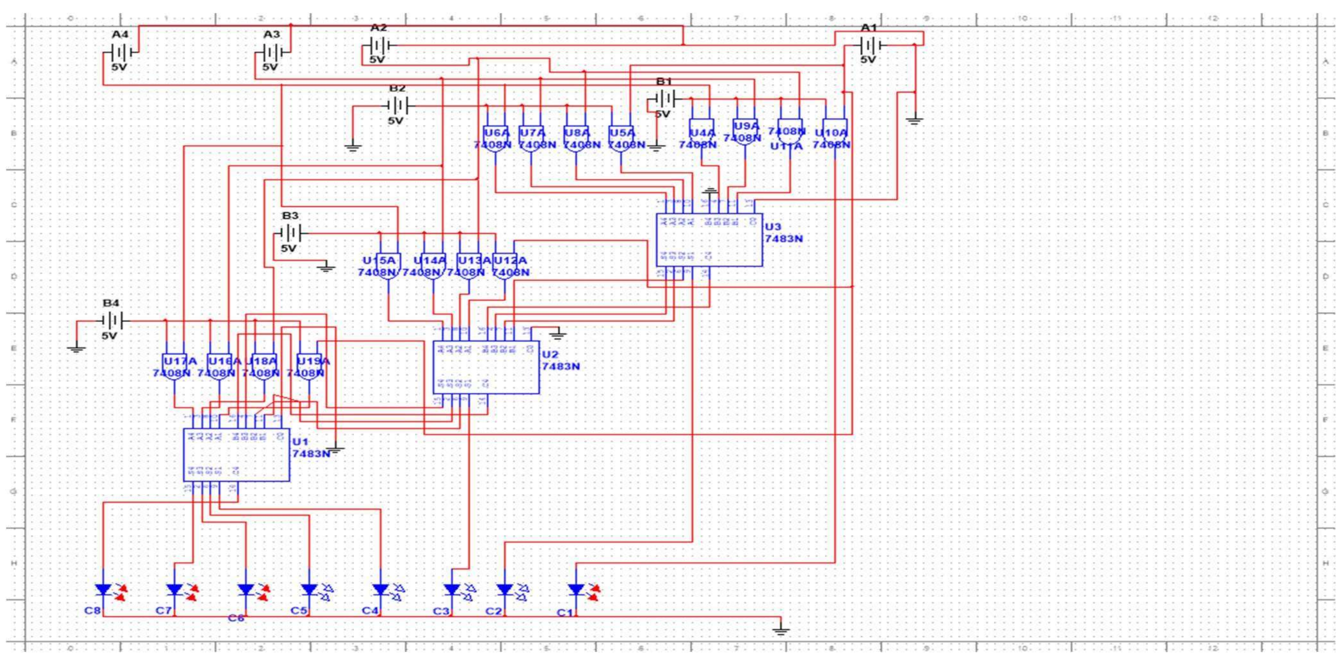


Circuit Simulation

The circuit diagram on Multisim is shown below



When all the input bits are turned on, A = 1111 (15 in decimal) and B = 1111 (15 in decimal) , the output (A x B) is 11100001 (225 in decimal) as shown below

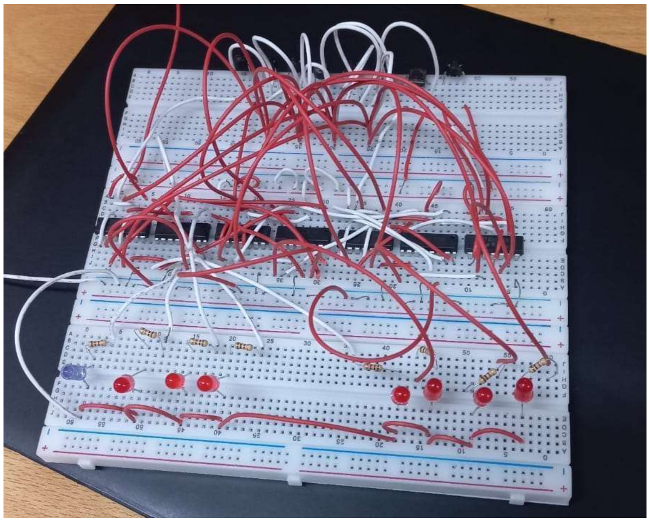


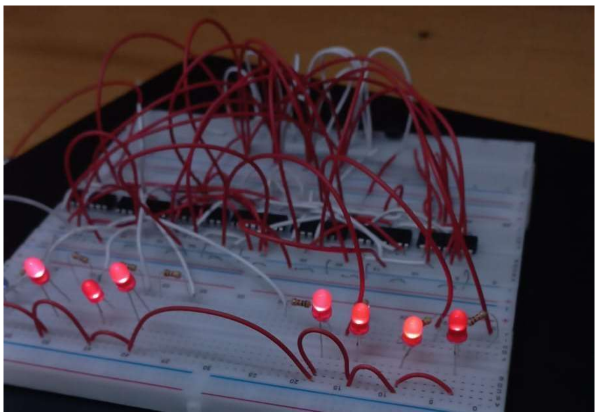
Simulated Results

When all input bits are zero, that means we are multiplying two numbers that are 0000 * 0000. This will give us an output of 00000000. That is shown by our simulation result.

When all inputs are 1s. Then that means we are multiplying 1111 * 1111. This makes 15 in decimal. 15 * 15 makes 225. We got an output of 11100001.

Model Pictures





Here we have multiplied 14 and 9 and got this output.

Working

The And gates are used to check if there is a signal at both inputs (A and B), just at one input, or no signal at both inputs. The least significant bit of the product is the output when the least significant bits of both numbers are input into an And gate, for the other bits of the product we need to use the adders.

Need of 4-bit Multiplier

A 4-bit multiplier is a specific type of digital circuit or component that is designed to perform the multiplication operation on 4-bit binary numbers. Here are some scenarios where a 4-bit multiplier might be useful:

1. Microprocessors and Microcontrollers: In the design of microprocessors and microcontrollers, arithmetic logic units (ALUs) are responsible for performing arithmetic operations. A 4-bit multiplier can be a part of the ALU to handle multiplication operations for 4-bit binary numbers.

2. Digital Signal Processing (DSP): Digital signal processors often deal with signal processing tasks that involve multiplication of binary numbers. Depending on the specific requirements of the DSP application, a 4-bit multiplier may be sufficient for certain operations.

3. Embedded Systems: In small-scale embedded systems with limited resources, a 4-bit multiplier might be preferred due to its simplicity and lower hardware requirements. This is especially relevant when dealing with applications where power consumption and space are critical factors.

4. Educational Purposes: In educational settings, a 4-bit multiplier can be used as a teaching tool to help students understand the principles of digital logic and arithmetic operations. It provides a simpler example for learning the basics of multiplication in binary.

5. Specific Applications: Some applications may have specific requirements for 4-bit operands and using a 4-bit multiplier in such cases can optimize the hardware design.

It's important to note that in many modern digital systems, especially those requiring high precision and performance, larger bit multipliers (such as 8-bit, 16-bit, or 32-bit multipliers) are more common. However, in certain situations where simplicity and resource constraints are key considerations, a 4-bit multiplier can still be relevant and useful.

Conclusion

The project ran perfectly, as expected, giving the correct value at output.