Table of Contents

Arithmetic Logic Unit (ALU)

Introduction:

An arithmetic-logic unit (**ALU**) is the part of a computer processor (CPU) that carries out arithmetic and logic operations on the operands in computer instruction words. We were given to design a 5-Bit ALU with following functionalities:

- Addition of two 5-bit numbers
- Subtraction of two 5-bit numbers
- Multiplication of two 5-bit numbers [Assuming user will enter the values such that output won't exceed 5-bit number]
- Division of two 5-bit numbers [Outputs: Quotient & divisor]
- Comparison of two 5-bit numbers, whether they are equal, greater or less.

5-Bit Addition:

Addition is the adding of bits. Bits can be added to in each other and generates the carry. The simple addition of two-bit truth table will be as:

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

The above given table is for addition that by adding what results, by using the above table we are going to make a two-bit full adder and then will be combining them to have 5-bit addition. Their combination will be done by giving the carry-out of previous as carry-in of upcoming.

Truth Table and Expressions:

Carry-In	A	B	Sum	Carry-Out
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

By simplifying the expressions, we get are:

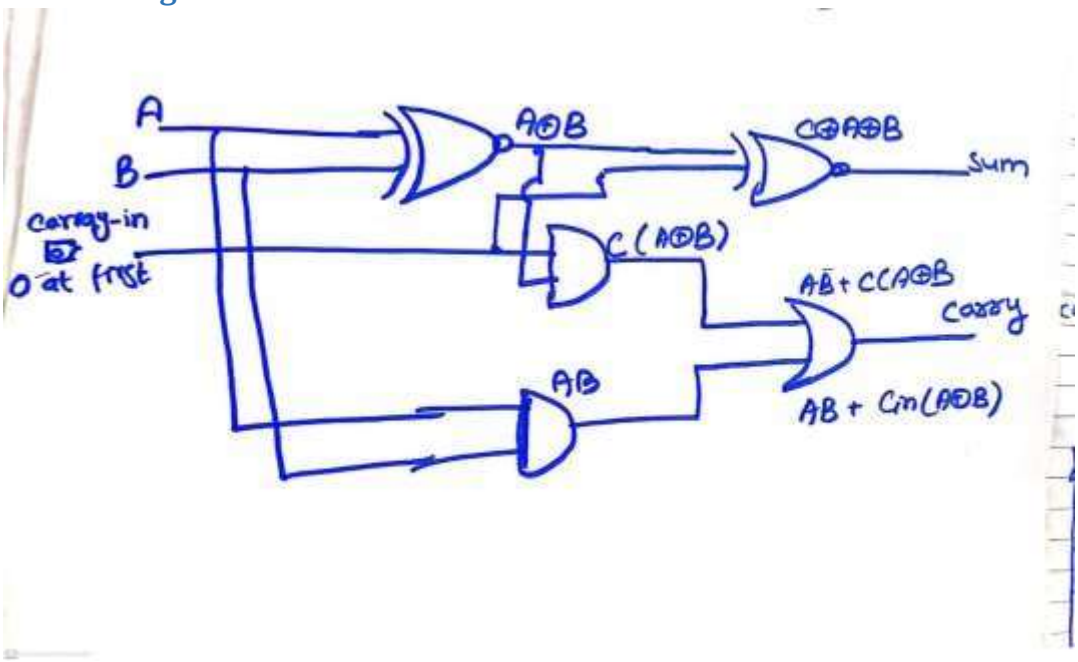
SUM = X0R (A, B, Cin).

Carray_Out = AB +Cin (XOR (A, B))

In the start Cin is given as 0, in start there is no carry in.

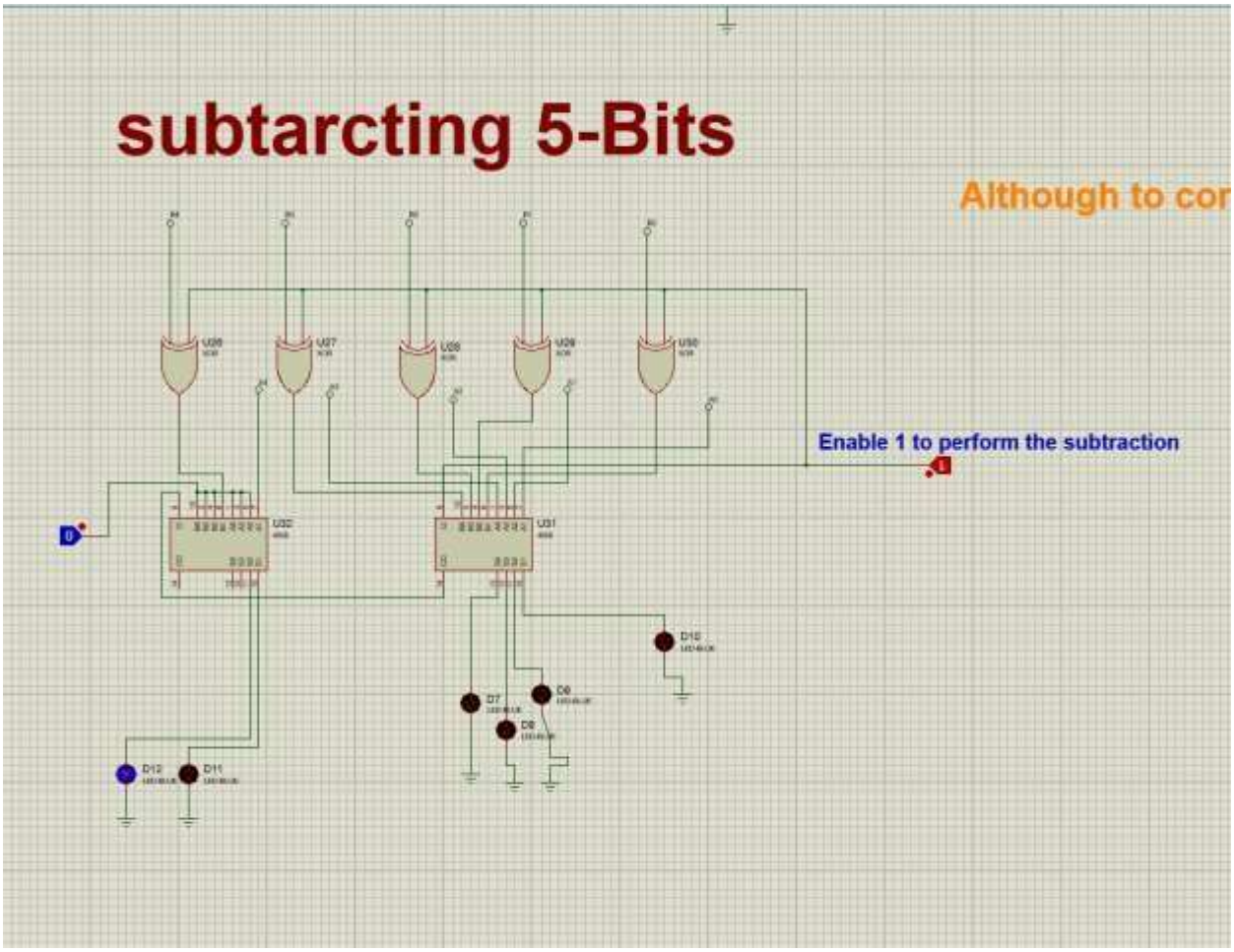
Since it gives how to add the two bits, as we have 5-bits to add we will add step by step first the addition of A_0 & B_0 will be done and its carry out will be feed as the carray_in in the addition of A_1 & B_1 and so on. Thus by making the Carray_Out as the carray_in of other will add the numbers accordingly and 5-bit two numbers will be added.

Circuit Diagrams:



5-Bit Subtraction:

Subtraction is the subtraction of the numbers. Subtraction is actually done by making a negative of other than adding them. First we make the B negative by the using the enables the circuit for that is given as below:

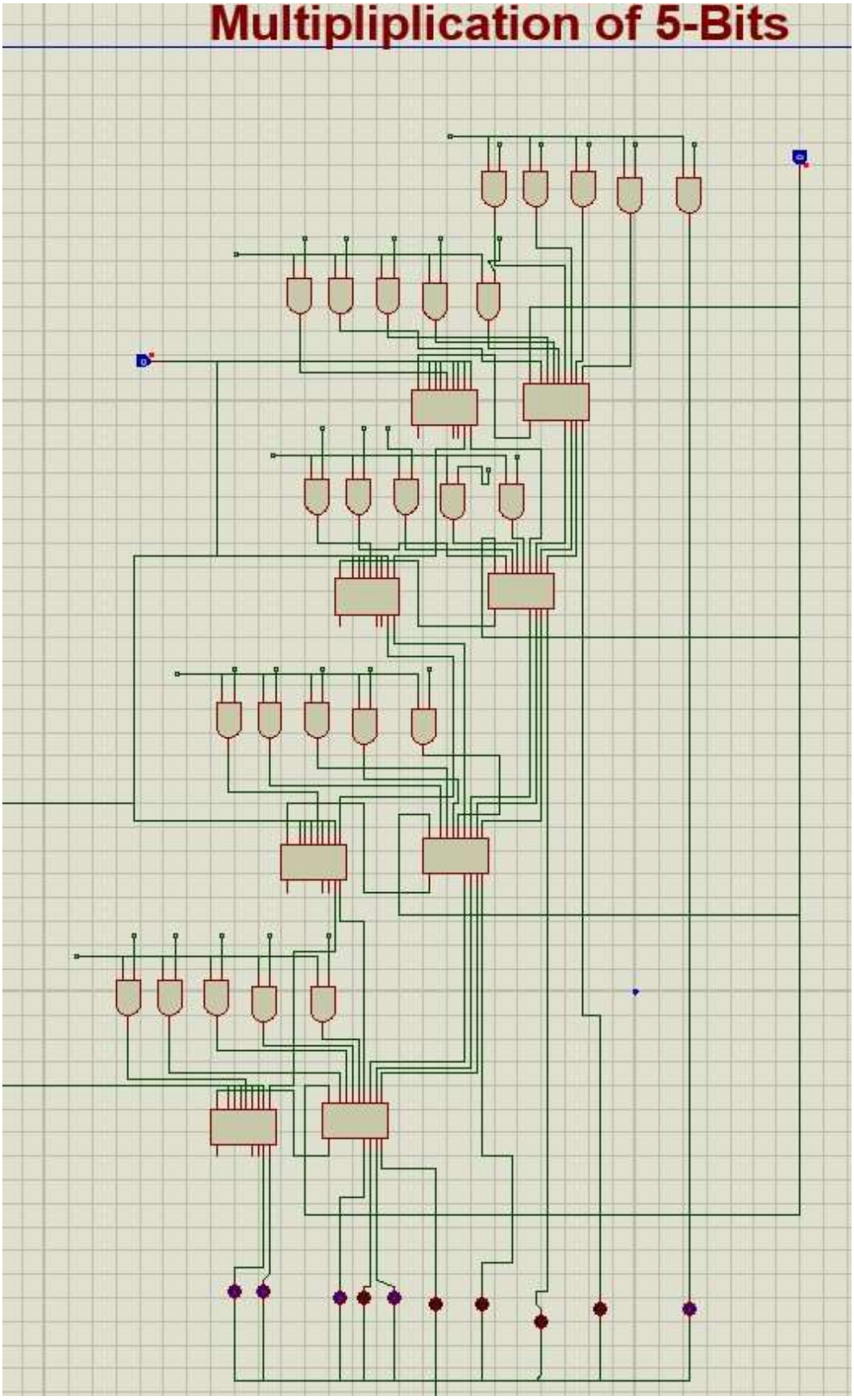


5-Bit Multiplication:

Multiplication of 5-Bit can be done by simple processes. The simple truth table of multiplication will be:

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

It is the AND gate expression, A.B since Multiplication can be done simply by Adding the bits, we will multiply the bits starting from the most significant bit, with the all bits of A and then add them by using the adders. For adding we are using the 4-bit adders and making two 4bit adders to add the 5-bits. The adder will be receiving the 5 bits and three other will be made off and will be adding 5-Bits.A circuit diagram from proteus is as follow:


Multipliplication of 5-Bits

5-Bit Comparator:

Comparator is used for the comparison between the numbers it can compare whether the numbers are equal, which is greater and which is lower. The three circuits are made when is for comparing A<B, second A==B and third A>B. Making the circuits for the two bit numbers then generalizing it. The respective truth table will be;

Truth Table and Expressions:

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

Table 1 Truth Table for comparison

Expressions	
A<B	A'B
A==B	A'B' + AB = (A'B+AB')'
A>B	AB'

Explanation for addition expression:

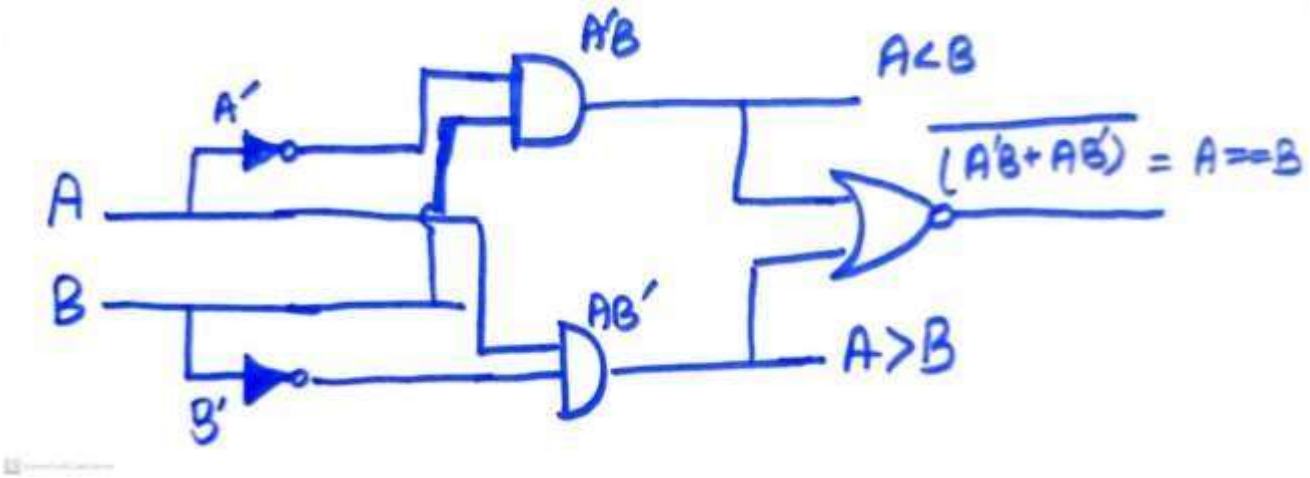
We had derived an expression (A'B + AB')', we had made this so we can have the easy implementation because terms A'B and AB' are already used in the expressions of A>B & A<B. Now proving that written expression is equal to the direct one obtained from the truth table;

$$(A'B + AB')' = (A+B') \cdot (A'+B) = AA' + AB + A'B + BB'$$

$$= AB + A'B' = \text{derived directly from truth table}$$

Thus A==B: (A'B + AB')

Circuit Diagrams:



Generalizing expressions for 5-bit numbers:

If we consider comparison in our daily life we compare the most significant numbers the starting ones, thus as we know that when a bit is equal and less or greater thus we can deduce the equations for 5-bit Numbers. 5-bit numbers may contain A_0, A_1, A_2, A_3, A_4 and similarly for B, B_0, B_1, B_2, B_3, B_4. Thus generalizing the expressions; **For Equal A==B:**

For equality each bit be equal as for one bit we have ***(A'B+AB')'*** thus for each bit making changes and equality expression for 5-bit number will be:

A==B: $(A_4'B_4 + A_4B_4')'.(A_3'B_3 + A_3B_3')'.(A_2'B_2 + A_2B_2')'.(A_1'B_1 + A_1B_1')'.(A_0'B_0 + A_0B_0')'$

From above equation making some substation so to write easily in upcoming next expressions as it is used.

$$X_4 = (A_4'B_4 + A_4B_4')' \qquad X_3 = (A_3'B_3 + A_3B_3')'$$

$$X_2 = (A_2'B_2 + A_2B_2')' \qquad X_1 = (A_1'B_1 + A_1B_1')'$$

$$X_0 = (A_0'B_0 + A_0B_0')'$$

For A<B:

Since as we mentioned earlier we see the starting bits so will compare the starting bits if starting are equal then go to next and so on till the last one. Since for single bit having A<B is given as A'B, thus we will have:

A<B: $(A_4'B_4) + X_4(A_3'B_3) + X_4X_3(A_2'B_2) + X_4X_3X_2(A_1'B_1) + X_4X_3X_2X_1(A_0'B_0)$

For A>B:

Since as we mentioned earlier we see the starting bits so will compare the starting bits if starting are equal then go to next and so on till the last one. Since for single bit having A>B is given as AB', thus we will have:

$$A<B: (A_4B_4') + X_4(A_3B_3') + X_4\,X_3(A_2B_2') + X_4\,X_3X_2(A_1B_1') + X_4\,X_3X_2\,X_1(A_0B_0')$$

Display:

We have made a unique display that is unique than the other displays at the time of display when numbers are Equal it prints E in equal circuit output and two other less and greater are filled with – while when they are not equal the equal out pout is n (intersection symbol) to show not equal and upper represents the smaller number and lower the greater number.

For Equality:

The truth table was as:

Output y	a		b	c	d	e	f	g	Expressions
0 (not equal)	1		1	1	0	a, e, f			1
1 (equal)	1		0	0	1		b, c		Y'
						d, g			Y

Inputting these in the 7-segment display the common cathode that is ground. To display E for equality and n (intersection type sign) for inequality.

For Lesser (upper):

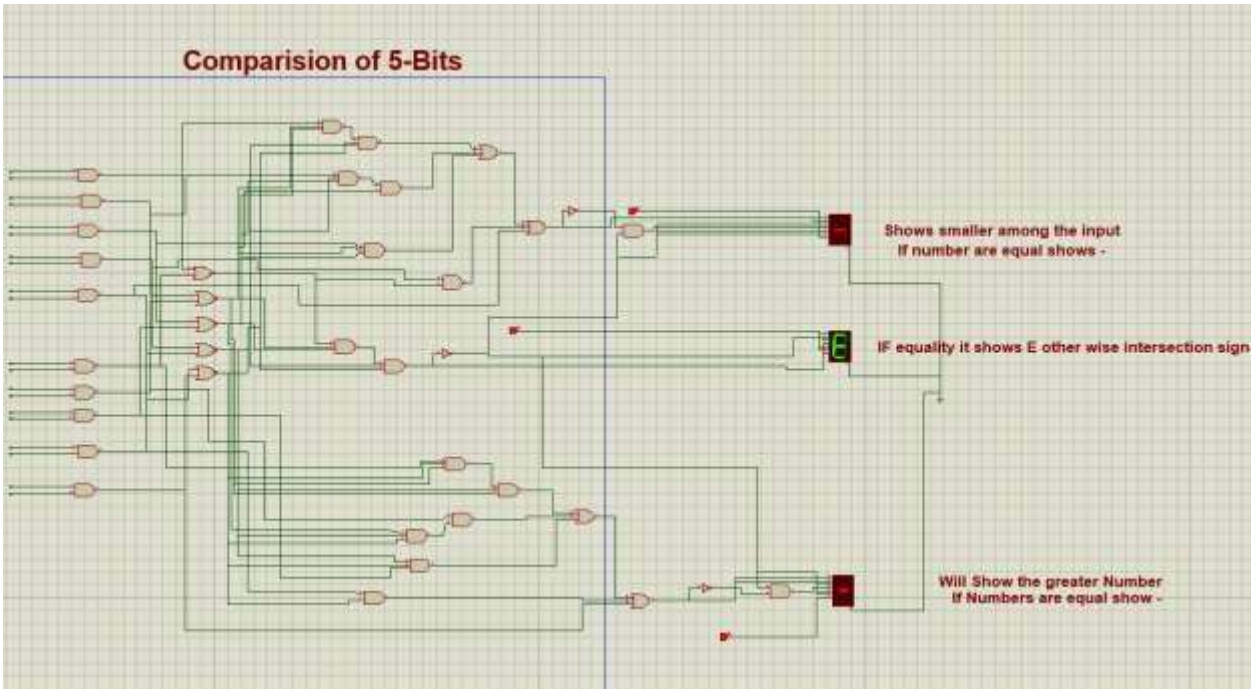
The truth table was as, Y2 is the current output and y is of equality acting as the enable if it 0 then works display otherwise display -, if A<B gives 0 mean B is smaller thus printing b but when gives 1 means A is lesser printing A simplified expression are given below:

Output y equality	Y2	a	b	c	d	e	f	g	Expressions
					a, b,				Y' Y2
0 (not equal)	0	0	0	1	1	1	1	1	1
0 (not equal)	1	1	1	1	c, e, f				Y'
1 (equal)	X	0	0	0	0	0	0	1	Y'Y2'

For greater (lower):

The truth table was as, Y2 is the current output and y is of equality acting as the enable if it 0 then works display otherwise display -, if A>B gives 0 mean B is greater thus printing b but when gives 1 means A is greater printing A, simplified expression are given below:

Output y equality	Y2	a	b	c	d	e	f	g	Expressions
					a, b,				Y'Y2'
0 (not equal)	0	1	1	1	0	1	1	1	1
0 (not equal)	1	0	0	1	c, e, f				Y'
1 (equal)	X	0	0	0	0	0	0	1	Y'Y2


Comparision of 5-Bits
Shows smaller among the input
If number are equal shows -
IF equality it shows E other wise intersection sign
Will Show the greater Number
If Numbers are equal show -