Experiment No. 6

Title: Design of 4 bit ALU to do the following operations:

ADD / SUB, Bitwise AND, OR, EXOR, 1 bit rotate left / right.

Batch: B1 **Roll No.: 16010420133** **Experiment No.: 6**

Aim: Design of 4 bit ALU to do the following operations:

ADD / SUB, Bitwise AND, OR, EXOR, 1 bit rotate left / right.

Resources needed: Simulation Software, (Circuitverse).

Theory:

Arithmetic Logic Unit:

In computing, an arithmetic logic unit (ALU) is a digital circuit that performs arithmetic and logical operations. The ALU is a basic building block of the central processing unit (CPU) of a computer.

Arithmetic and Logic Units (or ALUs) are found at the core of microprocessors, where they implement the arithmetic and logic functions offered by the processor (e.g., addition, subtraction, AND'ing two values, etc.). An ALU is a combinational circuit that combines many common logic circuits in one block. Typically, ALU inputs are comprised of two N-bit busses, a carry-in, and M select lines that select between the 2^M ALU operations. ALU outputs include an N-bit bus for function output and a carry out.

K. J. SOMAIYA COLLEGE OF ENGG.

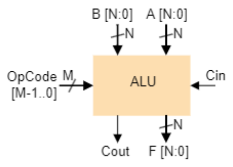


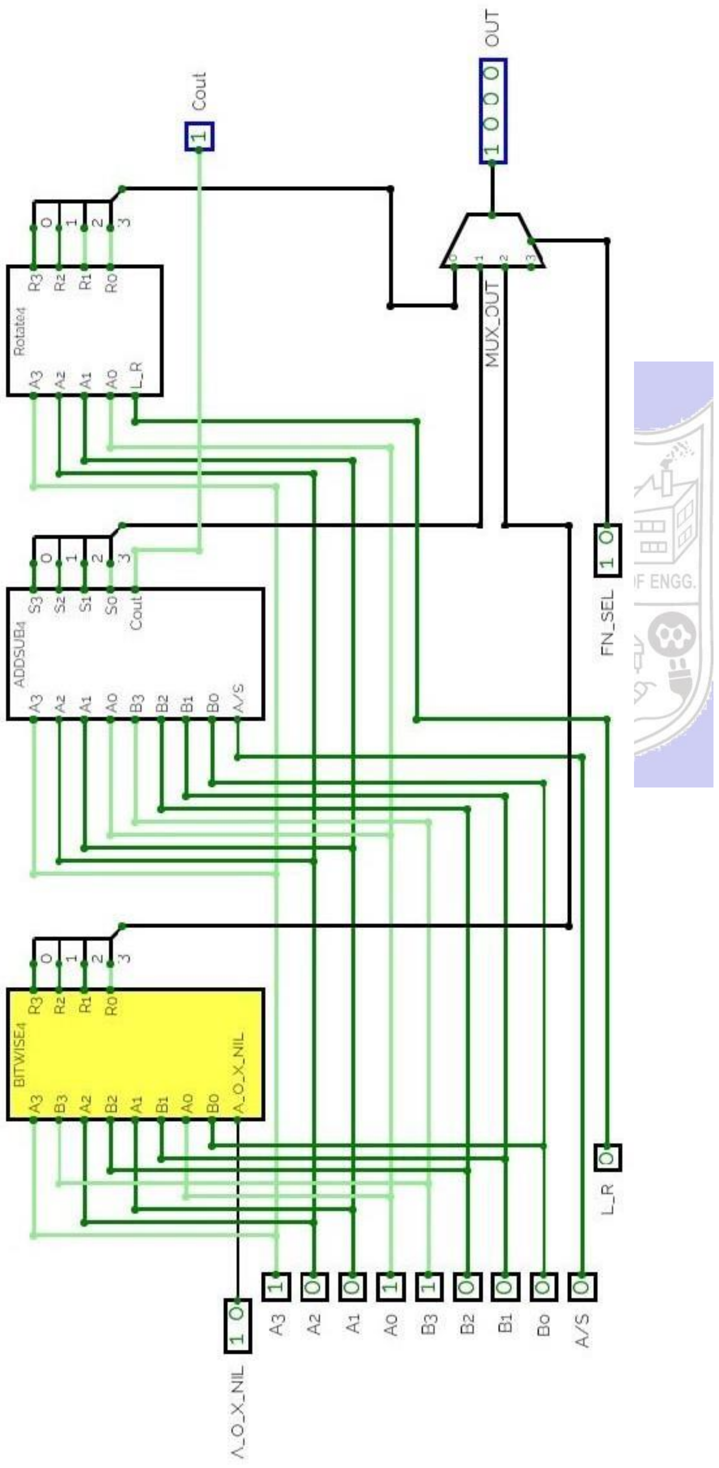
Figure 1. Arithmetic and logic units

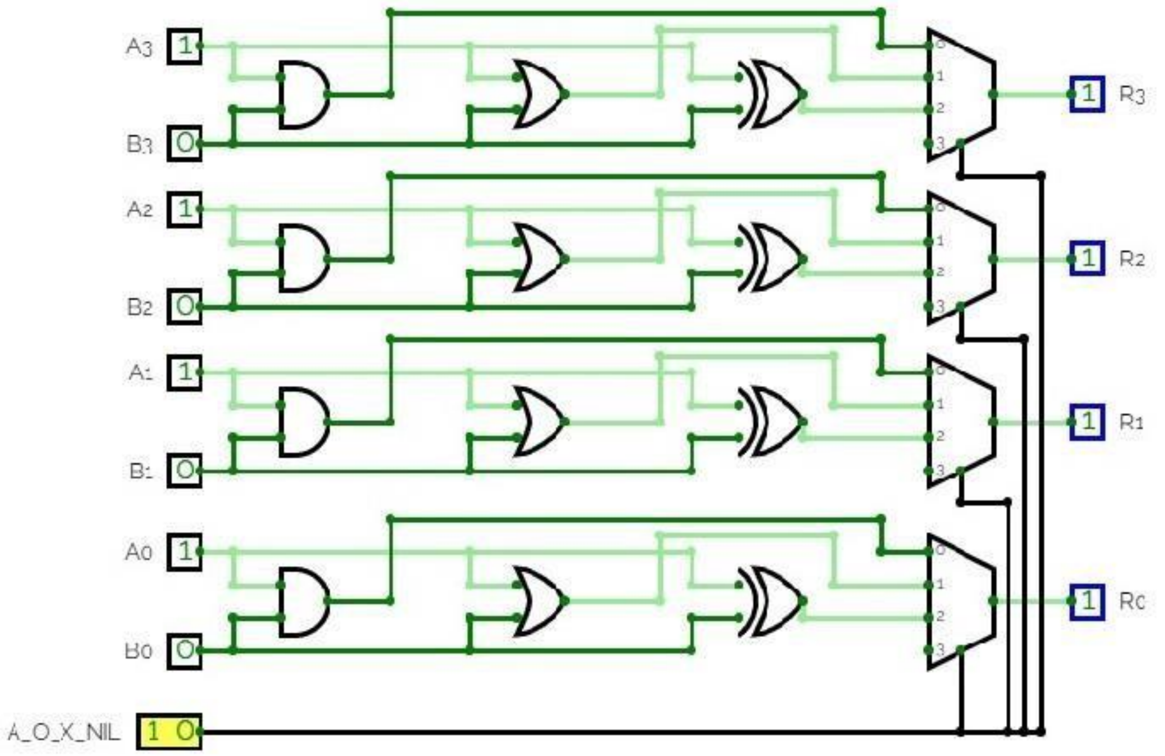
ALUs can be designed to perform a variety of different arithmetic and logic functions. Possible arithmetic functions include addition, subtraction, multiplication, comparison, increment, decrement, shift, and rotate; possible logic functions include AND, OR, XOR, XNOR, INV, CLR (for clear), and PASS (for passing a value unchanged). All of these functions find use in computing systems, although a complete description of their use is beyond the scope of this document. An ALU could be designed to include all of these functions, or a subset could be chosen to meet the specific needs of a given application. Either way, the design process is similar (but simpler for an ALU with fewer functions).

Procedure:

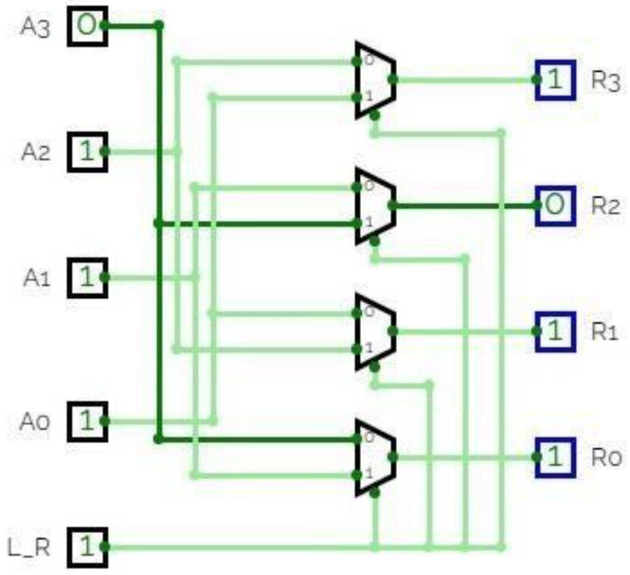
Design the circuits as shown below

DIS LAB6 – 4 BIT ALU

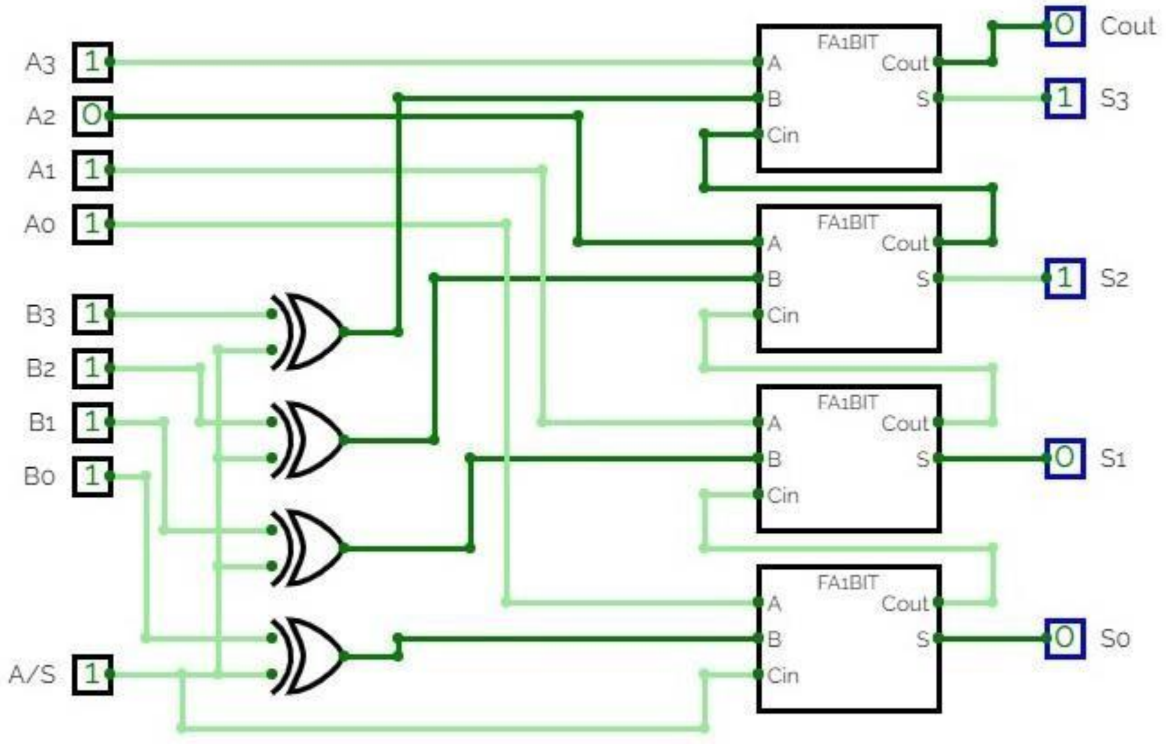




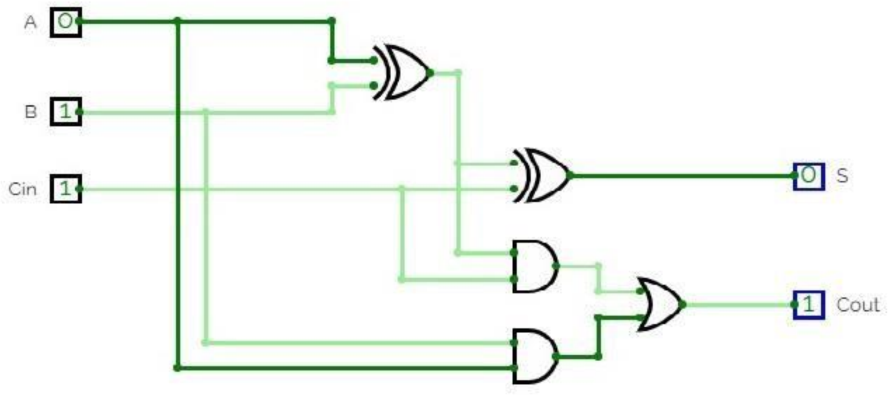
BITWISE AND, OR, EXOR



ROTATE LEFT/RIGHT



4 BIT ADD / SUBTRACT



1 BIT FULL ADDER

Observations and Results: Simulate some of the circuits as discussed in the Lab session and analyse the circuits and write the working of each of the circuit shown above.

4 – bit ALU:
The purpose of the ALU is to perform mathematical operations such as addition, subtraction, multiplication, and division. Additionally, the ALU processes basic logical operations like

AND/OR calculations. It serves as the computational hub of the Central Processing Unit (CPU) for a computer system.

There is total 8 inputs in the given circuit. Every time input is taken, and different operations are performed, like bitwise operations, addition/subtraction and rotating the bits. All the outputs of the operations are then provided as inputs to the multiplexer which in turn gives the output.

Controlled by the four function select inputs and the mode control input, ALU can perform all the 16 possible logic operations or 16 different arithmetic operations on active HIGH or active LOW operands.

When the mode control input is HIGH, all internal carries are inhibited, and the device performs logic operations on the individual bits. When control is LOW, the carries are enabled and the ALU performs arithmetic operations on the two 4-bit words. The ALU incorporates full internal carry look-ahead and provides for either ripple carry between devices using the Cn+4 output, or for carry look-ahead between packages using the carry propagation and carry generate signals.

A=B is an open collector output and can be wired-AND with other A=B outputs to give a comparison for more than 4 bits. The open drain output A=B should be used with an external pull-up resistor to establish a logic HIGH level. The A=B signal can also be used with the Cn+4 signal to indicate A > B and A < B.

The function table lists the arithmetic operations that are performed without a carry in. An incoming carry adds a one to each operation.

Because subtraction is performed by complementary addition (1s complement), a carry out means borrow; thus, a carry is generated when there is no under-flow, and no carry is generated when there is underflow.

BITWISE AND, OR, EXOR

The given circuit either performs AND or OR or XOR. Each Case is shown below -
a_o_x = 0 -> AND
1 -> OR
2 -> XOR

The 4th output is NIL. Multiplexer causes the functions to differ.

```
       a3 a2 a1 a0 b3 b2 b1 b0
 AND  1   1   1   1   0   0   0   0 -> 0000
OR     1   1   1   1   0   0   0   0 -> 1111
 XOR  1   1   1   1   0   0   0   0 -> 1111
```

ROTATE LEFT/RIGHT

A3 & A1 are both connected to multiplexers with outputs R2 & R0 on alternate input lines. (A3 to 1 on R2 & 0 on R0, vice versa for A1) Similarly, A2 & A0 are both connected to multiplexers with outputs R3 & R1.

The selector for all the multiplexers is common, and is set to 1 in this case, which corresponds to the input line 1 in the multiplexers. Thus the inputs on those lines will be the output of the multiplexers.

THUS:

A3 == R2

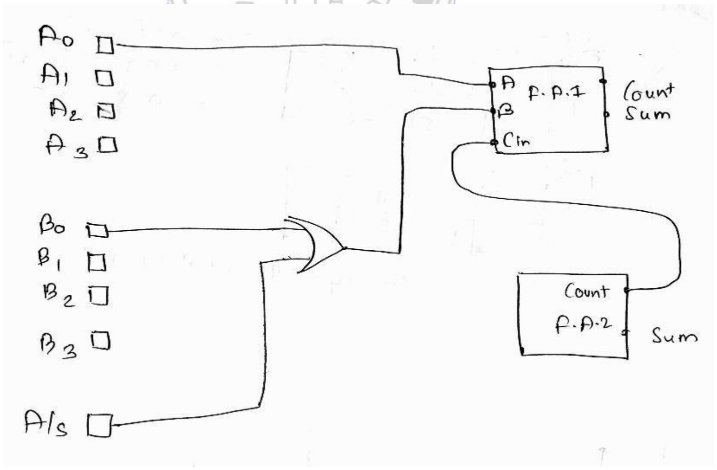
A2 == R1

A1 == R0

A0 == R3

4 BIT ADD / SUBTRACT

1) The circuit has 4 full adders parallelly connected where Cin of 1st 3 full adders are the count of the next full adder and the last /4th full adders' Cin is a control input that changes the circuit from a adder to a subtractor
2) This control input 'A/s' is converted to XOR gate with each of the beats of B.
3) Each full adder for example the first full adder has 1st I/p as A's 1st bit and 2nd I/p B's 1st bit XORed with control bit A/s as given below



Similarly for the next 3 full adder A's 1 bit as I/p and B's one bit XORed with control I/p "A/s" as 2nd I/p

4) When the Control i/p is'o' i.e The Cin For the 4th adder then the circuit acts as a adder i.e

Sum= A,s 4 bit +B's 4 bit XOR'O' +Cin

=A+B+Cin =A+B

As B+Cin +B , Cin = Control bit=0

5) When the control I/p is'1' then the circuit will perform subtraction I.e

Sum = A's 4 bit + B's Complemented 4 bits + Cin

Sum = A + (1st complement of B) + Cin

Where Cin = Control sigma=1

Sum = A + (1st complement of B) + 1

Sum= A + (2nd complement of B)

Since 1st Complement of B +1 =2's

Compliment of B

And also, we know adding any number with 2's compliment of another number is equal to their subtraction

Therefore, when control bit 'A/s' =1

Sum = A+2nd complement of B

= A- B

Therefor this ALU acts as a 4-bit adder and subtractor .

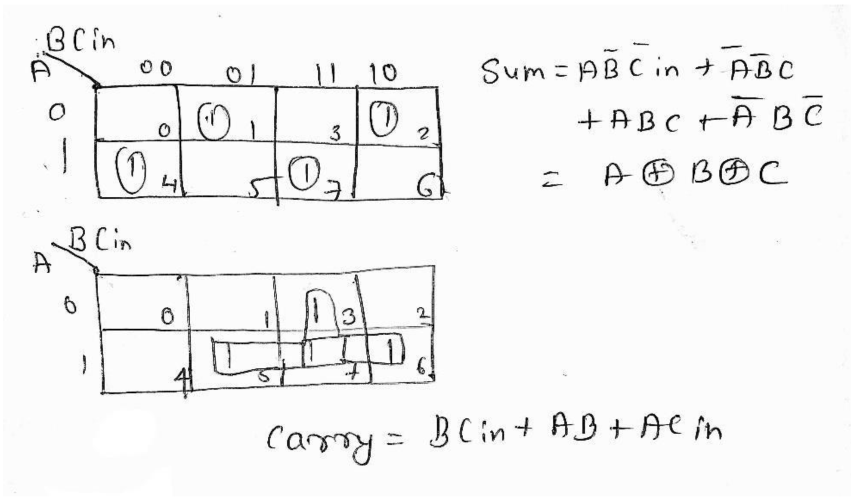
1 BIT FULL ADDER

This ALU is used to add 2 single bits with carry to generate its relevant sum and carry .

Truth Table

A	B	Cin	Sum	Count
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

K map : to find expression for sum and carry



BCin
A
00 01 11 10
0
1
Sum = $A\bar{B}\bar{C}in + \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C}$
$= A \oplus B \oplus C$
Carry = BCin + AB + ACin

A full adder adds binary numbers and accounts for values carried in as well as out. A one-bit full-adder adds three one-bit numbers, often written as A, B, and C_{in}; A and B are the operands, and C_{in} is a bit carried in from the previous less-significant stage.[2] The full adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. bit binary numbers. The circuit produces a two-bit output. Output carry and sum typically represented by the signals C_{in} and S, where the sum equals 2Cout + S.

Outcomes: Design of 4-bit ALU to do the following operations - Add, sub, and, or, exor, rotate left and rotate right.

Conclusion: Understood the working of 4-bit ALU to do the following operations:

ADD / SUB, Bitwise AND, OR, EXOR, 1 bit rotate left / right.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:
Books/ Journals/ Websites: