CSBB251: Computer Architecture and Organisation

Submitted By:

Name: KARTIK MITTAL

Roll No: 221210056

Branch: CSE

Semester: 4

Group : 2

Submitted To: Dr. Amandeep Kaur

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

2023

Lab Assignment -1

Aim – Download And Install Pspice.

Software Used – Chrome

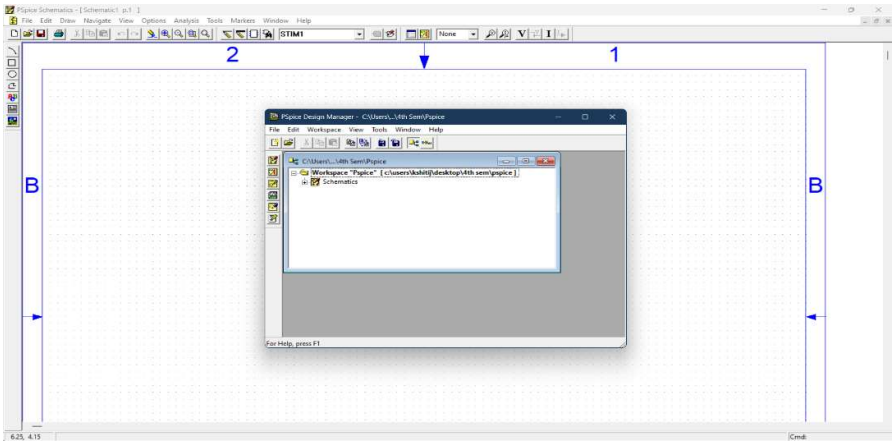
Theory –

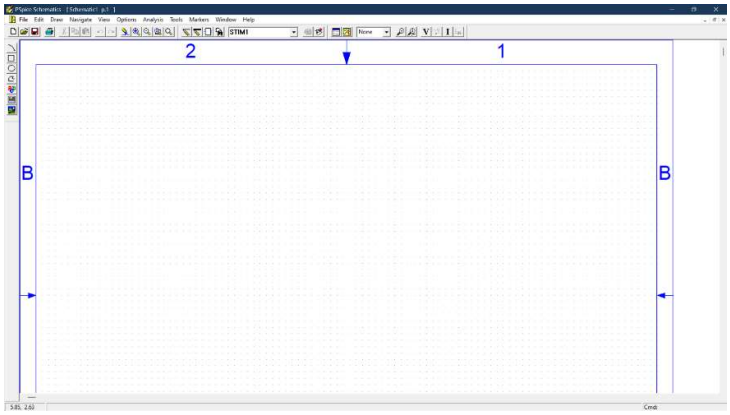
PSpice (Personal Simulation Program with Integrated Circuit Emphasis) is a widely used electronic circuit simulation software. Developed by MicroSim Corporation, PSpice is now a part of the Cadence Design Systems suite of electronic design automation (EDA) tools. PSpice allows engineers and designers to simulate the behaviour of electronic circuits before they are built, helping in the analysis and optimization of designs.

Here are some key points about PSpice:

1. Circuit Simulation: PSpice is primarily used for simulating electronic circuits. It enables designers to model and analyse the behaviour of circuits containing a variety of components such as resistors, capacitors, inductors, transistors, and integrated circuits.
2. Schematic Capture: PSpice provides a graphical user interface for designing circuits through schematic capture. Users can create circuit schematics by placing and connecting components using a drag-and-drop interface.
3. Simulation Types: PSpice supports different types of simulations, including DC analysis, AC analysis, transient analysis, and more. These simulations help in studying the circuit's behaviour under various conditions.

Output –




2
1
B
B

Lab Assignment -2

Aim – C Codes Of Various Codes

Software Used – VS Code

Theory –

```c
#include<stdio.h>
#include<math.h>
 int main(){

    int bit1,bit2;
    printf("Enter bit1 : ");
    scanf("%d",&bit1);
    printf("Enter bit2 : ");
    scanf("%d",&bit2);


    // AND GATE
    int and_gate = bit1 & bit2;
    printf("Output of AND gate is : %d ",and_gate);

    // NAND GATE
    printf("\nOutput of NAND gate is : %d ",abs(1-and_gate));

    // OR GATE
    int or_gate = bit1 | bit2;
    printf("\nOutput of OR gate is : %d ",or_gate);

    // NOR GATE
    printf("\nOutput of NOR gate is : %d ",abs(1-or_gate));

    // NOT GATE
    printf("\nOutput of NOT gate is : %d ",abs(1-bit1));

    // XOR gate
    int xor_gate=bit1^bit2;
    printf("\nOutput of XOR gate is : %d ",xor_gate);


    return 0;
}
```

Output –

```
Enter bit1 : 1
Enter bit2 : 1
Output of AND gate is : 1
Output of NAND gate is : 0
Output of OR gate is : 1
Output of NOR gate is : 0
Output of NOT gate is : 0
Output of XOR gate is : 0
PS C:\Users\HP\Desktop\college\semester4\Computer Architecture and organisation>
```

Lab Assignment -3

Aim – Half and Full Adder

Software Used – PSpice

Theory –

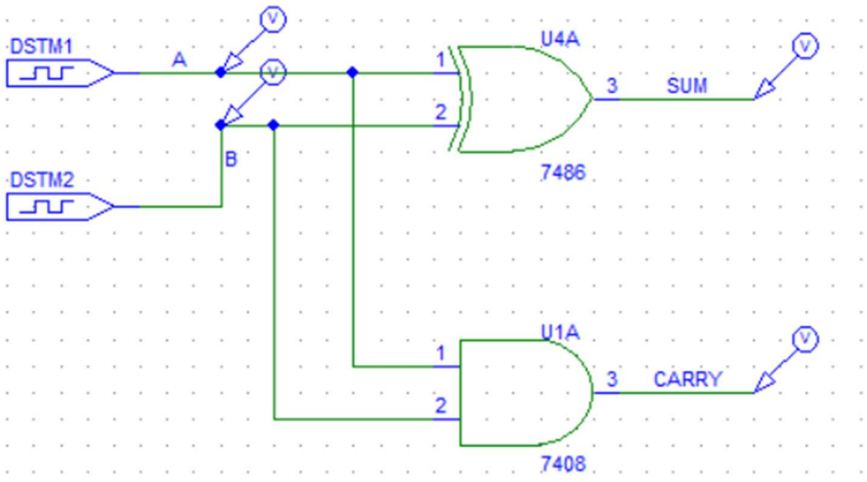
Half Adder

A half adder is a digital circuit that performs the addition of two single binary digits. It has two inputs, typically labelled A and B, and two outputs, the Sum (S) and the Carry (C). The Sum output represents the least significant bit of the sum of A and B, while the Carry output represents whether a 1 has been carried over to the next higher bit position (which would be needed if both A and B are 1).

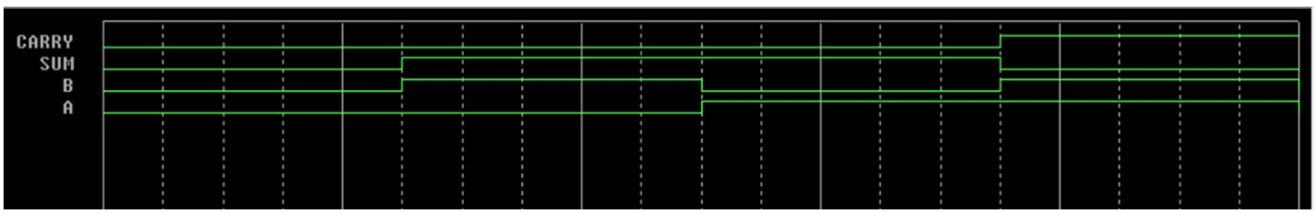
Full Adder

A full adder extends the concept of the half adder by including the ability to process a carry input (Cin). This makes it capable of adding together two binary digits and an incoming carry from the addition of previous digits. A full adder has three inputs: A, B, and Cin, and two outputs: Sum (S) and Carry (Cout).

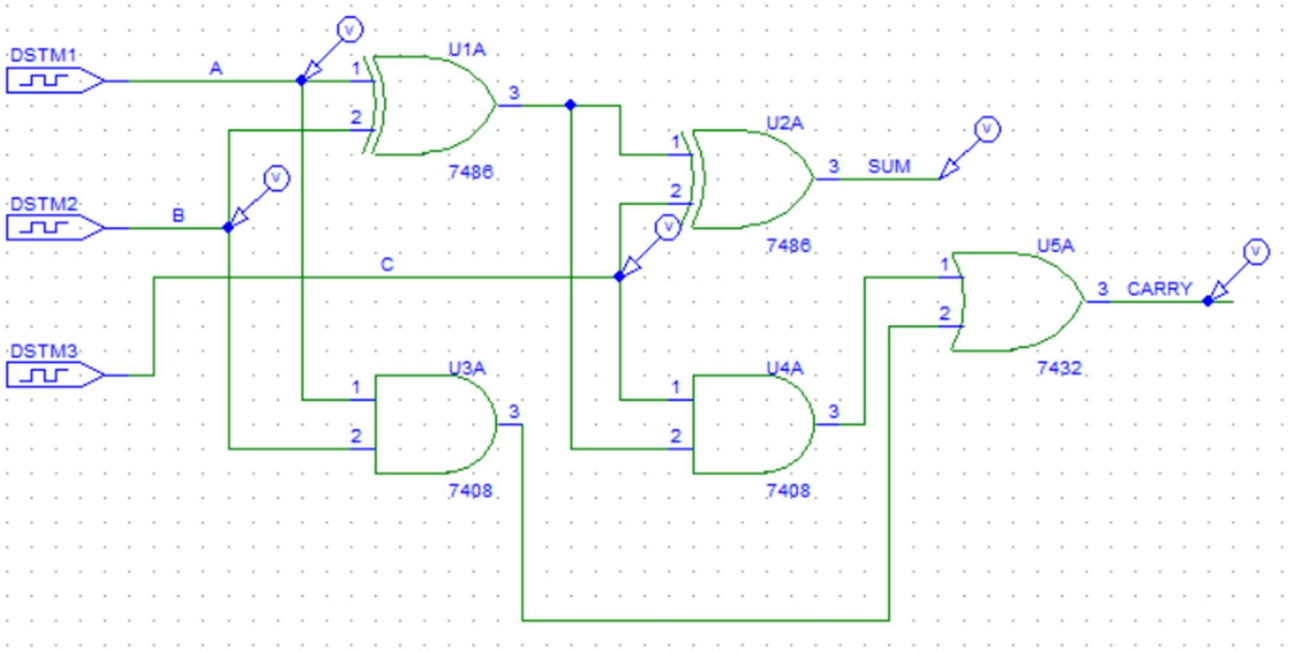
Half Adder Circuit –



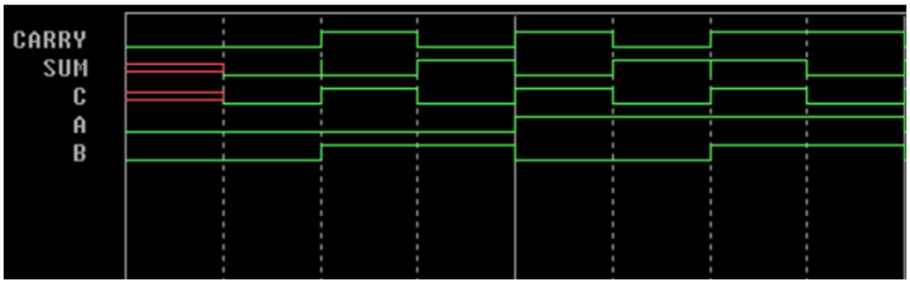
Output –



Full Adder Circuit –



Output –



Lab Assignment -4

Aim – Binary Adder Subtractor

Software Used – PSpice

Theory –

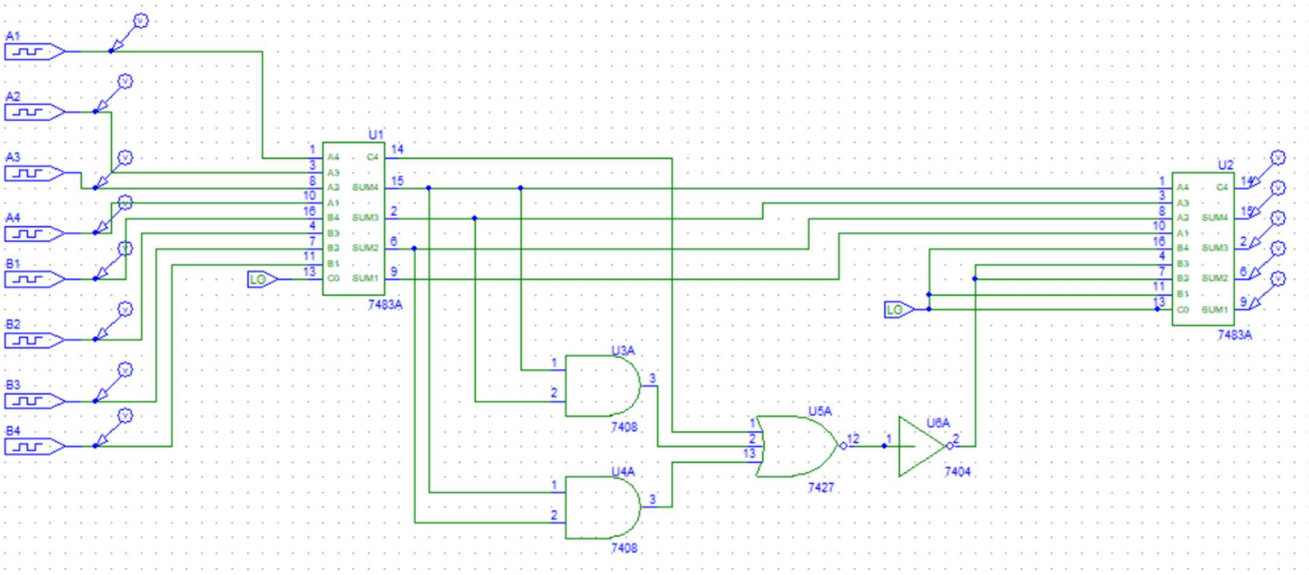
A binary adder-subtractor can switch between addition and subtraction operations by controlling how the second operand and the carry input are processed. It typically involves the following components:

- Full Adder(s): For adding bits of the operands. Multiple full adders are cascaded to handle multi-bit numbers.
- XOR Gates: To conditionally invert the bits of the second operand, effectively implementing the Two's Complement when subtracting.
- Control Input: A single bit that determines the operation mode (addition or subtraction). When the control input is 0, the circuit performs addition. When it is 1, the circuit performs subtraction by inverting the second operand and setting the initial carry input to 1 (to account for the +1 in Two's Complement).

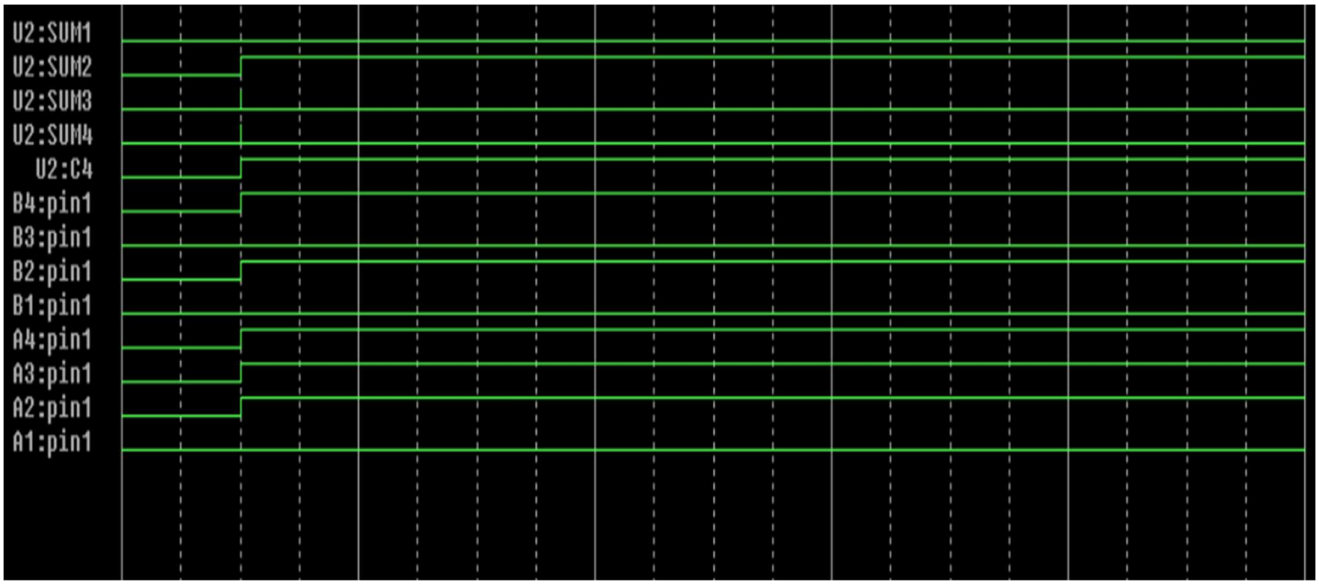
Operations

1. Addition: For addition, the control input is set to 0. The XOR gates pass the second operand directly (since XOR with 0 leaves the bit unchanged), and the carry in for the least significant bit is set to 0. The full adders then perform the addition as normal.
2. Subtraction: For subtraction, the control input is set to 1. The XOR gates invert the bits of the second operand (since XOR with 1 inverts the bit), and the carry in for the least significant bit is set to 1, effectively adding the Two's Complement of the second operand to the first. This operation yields the subtraction result.

Circuit –


A1
A2
A3
A4
B1
B2
B3
B4
LO
U1
7483A
U2
7483A
U3A
7408
U4A
7408
U5A
7427
U6A
7404

Output –


U2:SUM1
U2:SUM2
U2:SUM3
U2:SUM4
U2:C4
B4:pin1
B3:pin1
B2:pin1
B1:pin1
A4:pin1
A3:pin1
A2:pin1
A1:pin1

Lab Assignment - 5

Aim – 4bit Arithmetic Unit

Software Used – PSpice

Theory –

A 4-bit Arithmetic Logic Unit (ALU) is a digital circuit that performs arithmetic and logical operations on 4-bit binary numbers. It is a fundamental building block in the design of microprocessors, computers, and other digital systems. The ALU can execute a variety of operations, such as addition, subtraction, and bitwise operations like AND, OR, XOR, and NOT. The complexity and functionality of an ALU can vary, but a basic 4-bit ALU typically includes the following features:

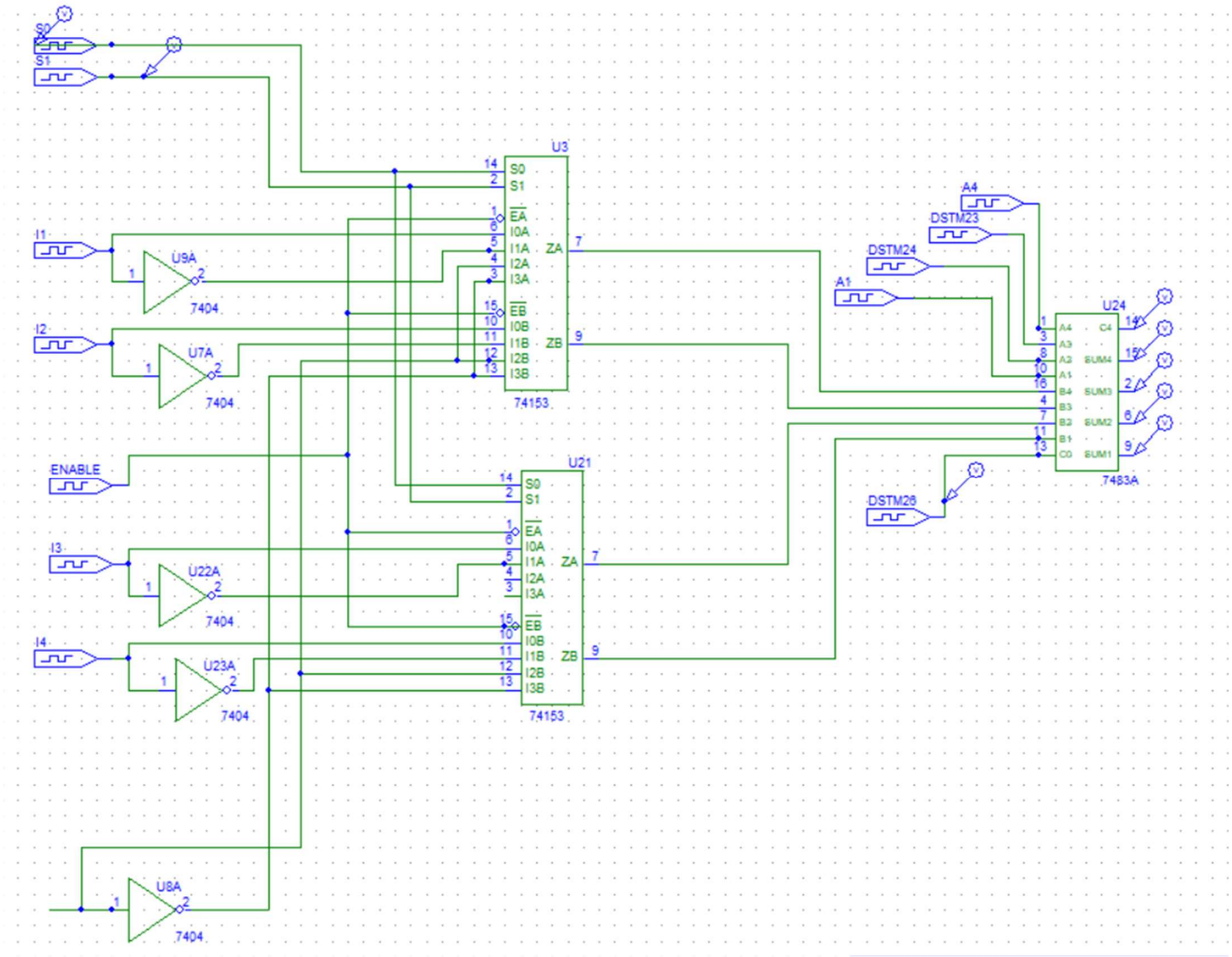
Core Functions

- **Arithmetic Operations**: The ALU can perform basic arithmetic operations on 4-bit operands, including addition and subtraction. Addition is typically implemented using a series of full adders, while subtraction can be performed by adding the Two's Complement of the second operand.
- **Logical Operations**: Logical operations include AND, OR, XOR, and NOT. These operations are performed bit by bit on the operands.

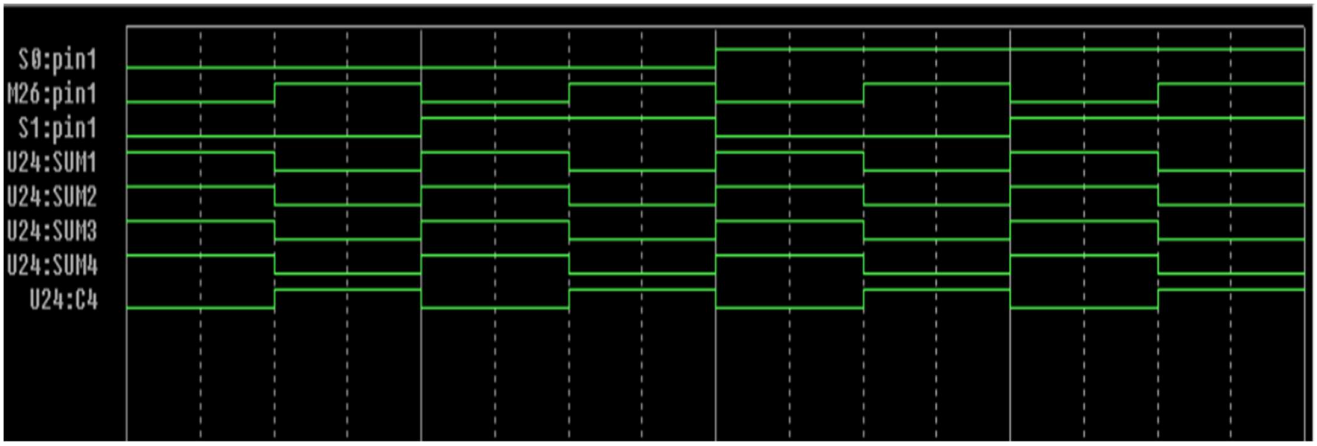
Components

- **Full Adders**: A 4-bit ALU uses four full adders to perform binary addition. Each full adder adds two input bits and a carry bit from the previous adder, producing a sum bit and a carry-out bit.
- **Multiplexers**: Multiplexers (MUX) are used to select the operation to be performed based on control inputs. They can choose between adding, subtracting, or performing logical operations.
- **Logic Gates**: A series of AND, OR, XOR, and NOT gates are used to perform bitwise logical operations.
- **Control Logic**: The control logic interprets the operation code (opcode) and generates signals to control the multiplexers and other components of the ALU, determining which operation is performed.

Circuit –


S0
S1
I1
I2
I3
I4
ENABLE
U9A
U7A
U22A
U23A
U8A
7404
U3
U21
74153
ZA
ZB
A4
DSTM23
DSTM24
A1
DSTM26
U24
7483A
SUM4
SUM3
SUM2
SUM1
C4

Output –


S0:pin1
M26:pin1
S1:pin1
U24:SUM1
U24:SUM2
U24:SUM3
U24:SUM4
U24:C4

Lab Assignment - 6

Aim – Circular Shift and Logic Shift

Software Used – PSpice

Theory –

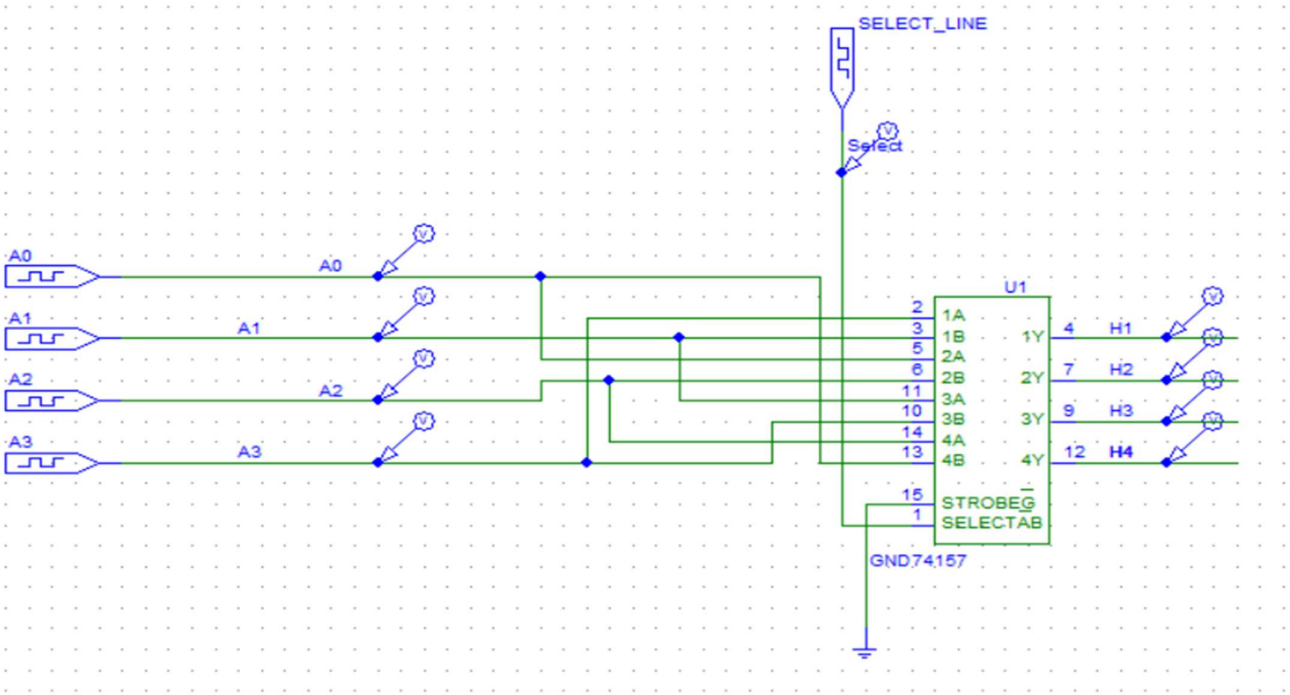
A circular shift, also known as a cyclic shift or a rotate operation, is a type of bit manipulation operation performed on binary data. In a circular shift, the bits of a binary number are shifted to the left or right by a specified number of positions, with the key characteristic that no bits are lost in the process. Instead, bits that are shifted out on one end are reintroduced on the opposite end. This operation is useful in various computing tasks, including cryptography, data compression, and the implementation of certain algorithms.

Types of Circular Shifts

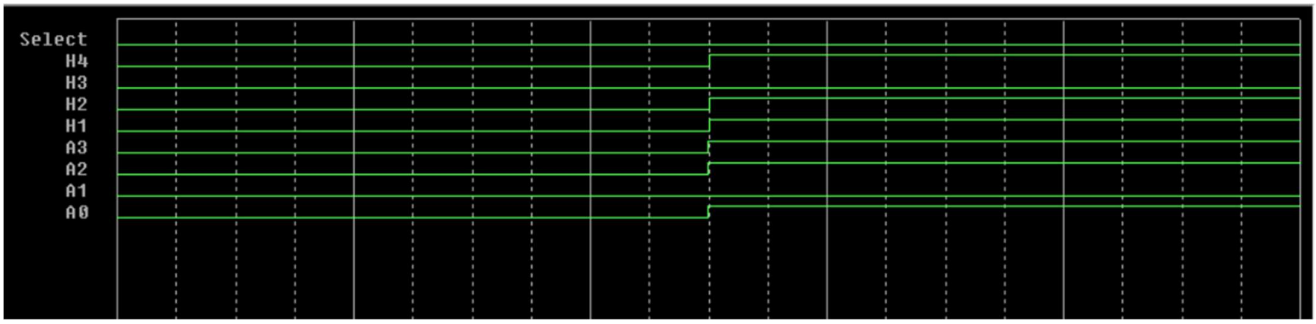
There are two primary types of circular shifts:

1. **Circular Left Shift (Rotate Left)**: In a circular left shift, all bits are shifted one position to the left, and the leftmost bit (most significant bit) is moved to the rightmost position (least significant bit).
2. **Circular Right Shift (Rotate Right)**: In a circular right shift, all bits are shifted one position to the right, and the rightmost bit (least significant bit) is moved to the leftmost position (most significant bit).

Circular Shift Circuit –



Circular Shift Output –



Logic Shift -

Theory –

Logic shifts are bitwise operations that involve shifting the bits of a binary number left or right. There are two main types of logic shifts: logical left shift and logical right shift.

Types of Logical Shifts

There are two primary types of logical shifts:

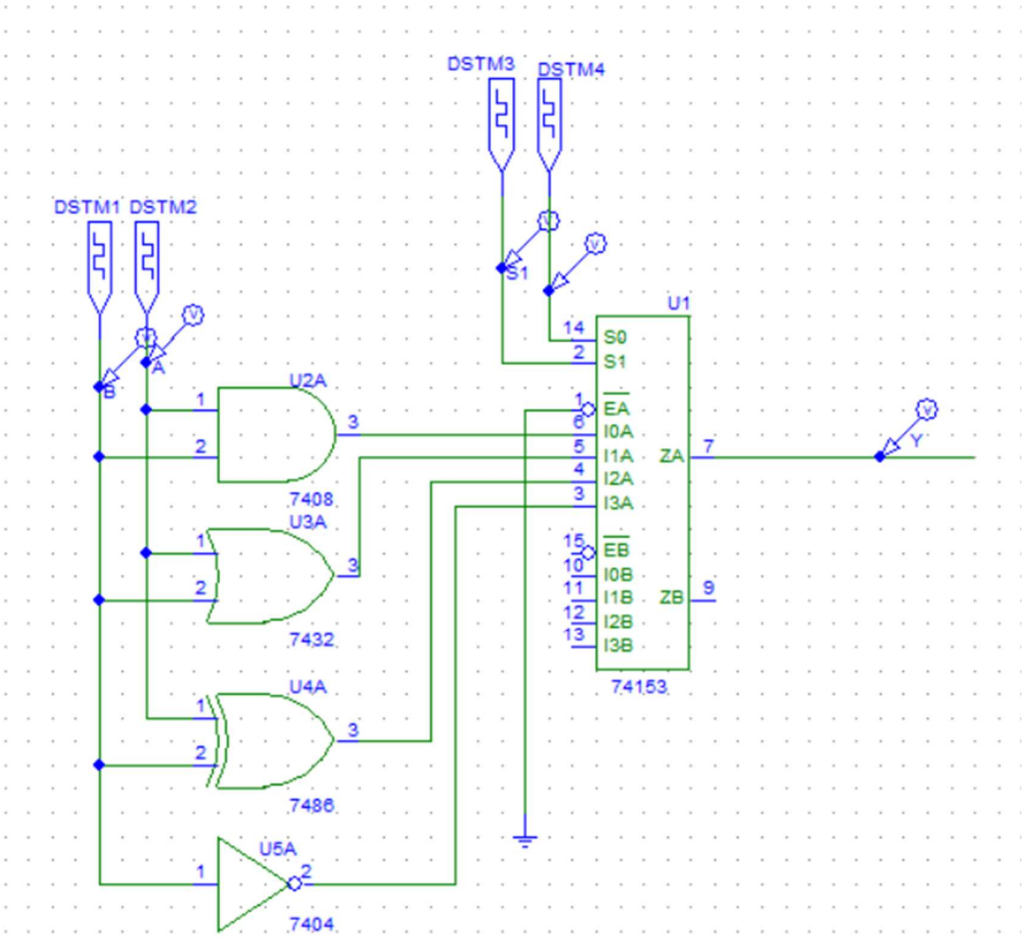
Logical Left Shift

Each bit in the binary representation is shifted to the left by a specified number of positions. The leftmost bits are shifted out, and zeros are shifted in on the right side. A logical left shift effectively multiplies the number by 2 raised to the power of the shift count. Often denoted by the **<<** operator in programming languages.

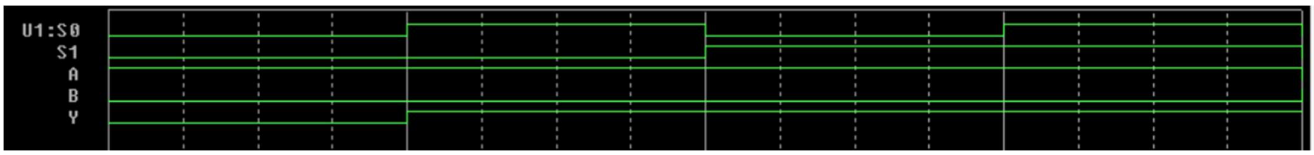
Logical Right Shift

Each bit in the binary representation is shifted to the right by a specified number of positions. The rightmost bits are shifted out, and zeros are shifted in on the left side. A logical right shift effectively divides the number by 2 raised to the power of the shift count. Often denoted by the **>>** operator in programming languages.

Logic Shift Circuit –



Logic Shift Output –



Lab Assignment - 7

Aim – Implement ALU

Software Used – PSpice

Theory –

An Arithmetic Logic Unit (ALU) is a critical component of a CPU (Central Processing Unit) in a computer. It is designed to carry out arithmetic and logic operations on the operands received from the processor. The ALU performs a wide range of operations, such as addition, subtraction, multiplication, division, as well as bitwise operations like AND, OR, XOR, and NOT. The specific set of operations that an ALU can perform depends on its design and the requirements of the CPU it is part of.

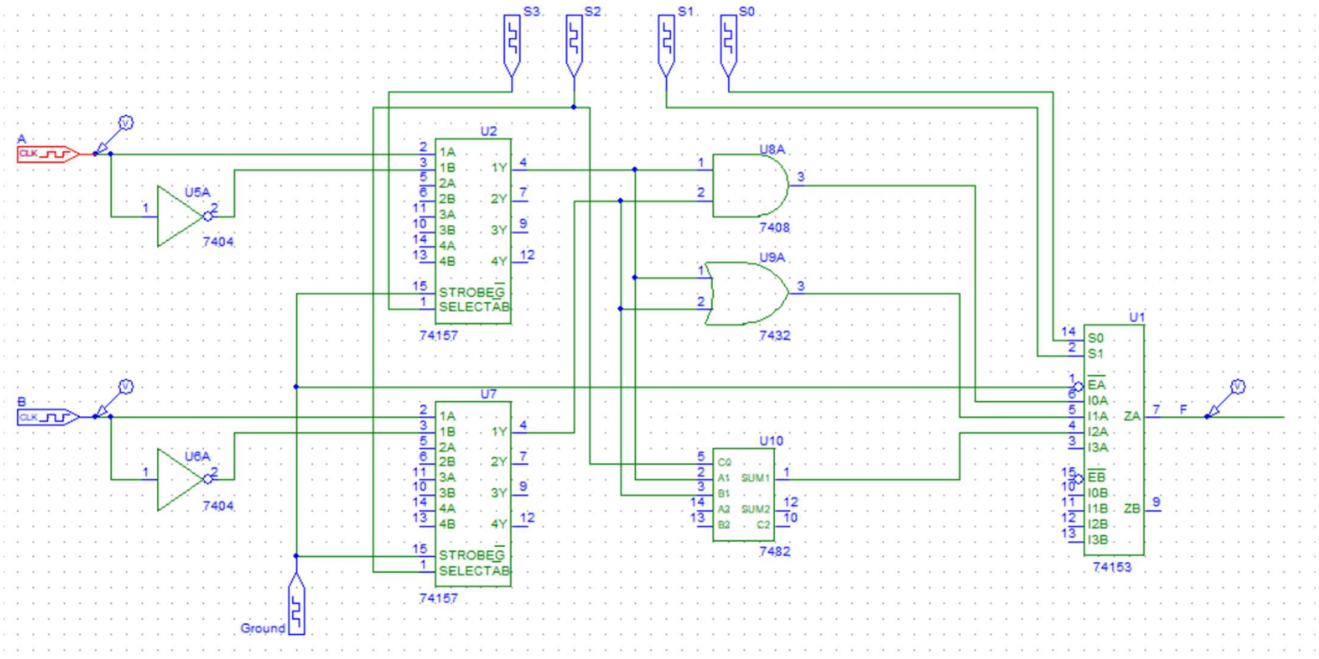
Core Functions

- **Arithmetic Operations**: These include basic calculations such as addition, subtraction, and in more complex ALUs, multiplication and division.
- **Logical Operations**: Operations involving bitwise manipulation of data, including AND, OR, XOR (exclusive OR), and NOT operations.

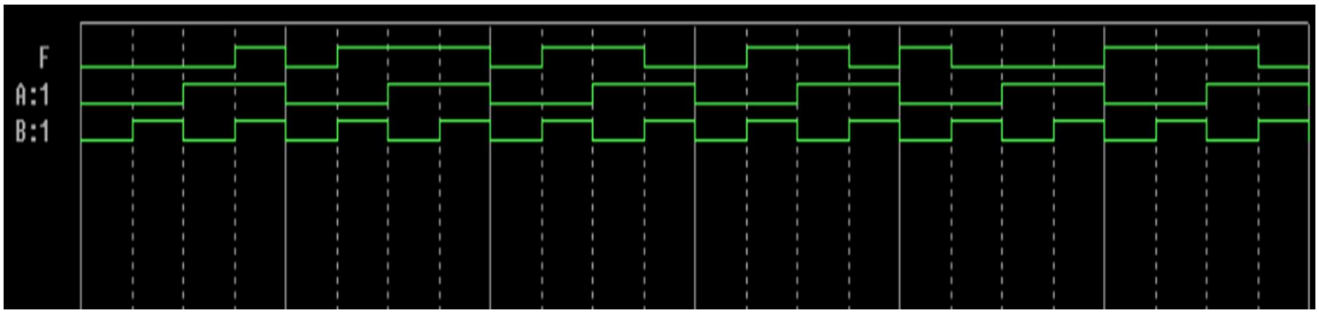
Components

- **Operands**: The data on which operations are performed. The ALU receives operands from the processor registers.
- **Operation Code (Opcode)**: A code that specifies which operation the ALU should

Circuit –



Output –



Lab Assignment - 8

Aim – Ripple Counter

Software Used – PSpice

Theory –

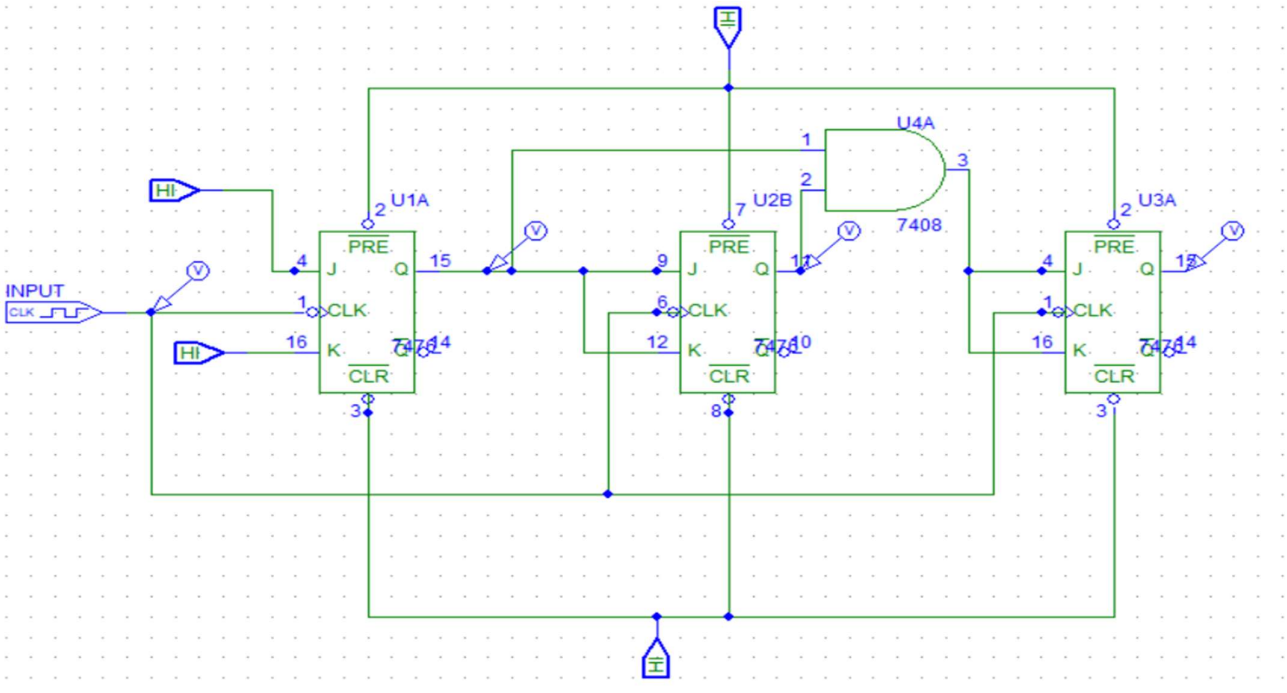
A ripple counter, also known as an asynchronous counter, is a type of digital counter used in electronics and computing for counting purposes. It consists of a series of flip-flops connected in a sequence where the output of one flip-flop serves as the clock input for the next flip-flop in the chain. This configuration causes the flip-flops to toggle states successively in a ripple effect, hence the name "ripple counter."

Basic Operation

The operation of a ripple counter starts with the first flip-flop, which is typically driven by an external clock pulse. Each time the clock input of this first flip-flop receives a pulse, it changes state from 0 to 1 or from 1 to 0. The transition of this flip-flop then acts as a trigger for the next flip-flop in the sequence, and so on, causing a ripple effect through the series of flip-flops.

Because the triggering of subsequent flip-flops depends on the state change of the previous flip-flops, there is an inherent propagation delay associated with ripple counters. This delay is the cumulative effect of the time it takes for each flip-flop in the series to respond to the clock pulse. As a result, the output of each flip-flop in a ripple counter does not change simultaneously but follows a sequential pattern that creates the characteristic ripple effect.

Circuit –



Output –