Experiment-06 Synchornous counters up counter:-

**AIM:**

To implement 4 bit up and down counters and validate functionality.

**HARDWARE REQUIRED: –** PC, Cyclone II , USB flasher **SOFTWARE REQUIRED:** Quartus prime

**THEORY:**

**UP COUNTER**

The counter is a digital sequential circuit and here it is a 4 bit counter, which simply means it can count from 0 to 15 and vice versa based upon the direction of counting (up/down).

The counter (“count“) value will be evaluated at every positive (rising) edge of the clock (“clk“) cycle. The Counter will be set to Zero when “reset” input is at logic high. The counter will be loaded with “data” input when the “load” signal is at logic high. Otherwise, it will count up or down. The counter will count up when the “up\_down” signal is logic

high, otherwise count down

Since we know that binary count sequences follow a pattern of octave (factor of 2) frequency division, and that J-K flip-flop multivibrators set up for the “toggle” mode are capable of performing this type of frequency division, we can envision a circuit made up of several J-K flip-flops, cascaded to produce four bits of output. The main problem facing us is to determine how to connect these flip-flops together so that they toggle at the right times to produce the proper binary sequence. Examine the following binary count sequence, paying attention to patterns preceding the “toggling” of a bit between 0 and 1: Binary count sequence, paying attention to patterns preceding the “toggling” of a bit between 0 and 1.

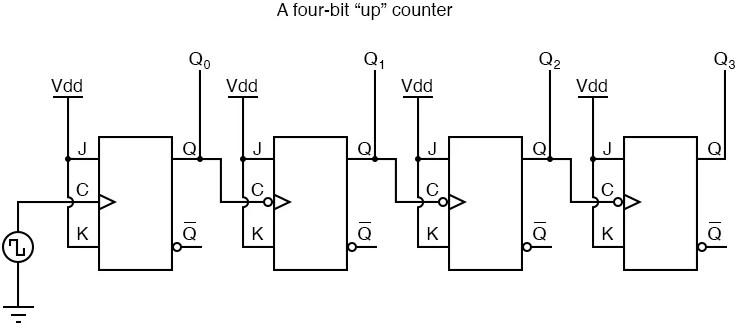
Note that each bit in this four-bit sequence toggles when the bit before it (the bit having a lesser significance, or place-weight), toggles in a particular direction: from 1 to 0.

Starting with four J-K flip-flops connected in such a way to always be in the “toggle” mode, we need to determine how to connect the clock inputs in such a way so that each succeeding bit toggles when the bit before it transitions from 1 to 0.

The Q outputs of each flip-flop will serve as the respective binary bits of the final, four-bit count: Four-bit “Up” Counter

**DOWN COUNTER**

A 4-bit down counter is a digital counter circuit, which provides a binary countdown from binary 1111 to 0000. This circuit uses four D-type flip-flops, which are positive edge triggered. At each stage, the flip-flop feeds its inverted output (/Q) back into its own data input (D). However, it feeds its non-inverted output (Q) to the clock input (CK) of the following stage. This type of circuit operates in an asynchronous (ripple) manner because the flip-flop stages do not rely on a common clock pulse for timing. The operational speed of the counter depends upon the signal propagation through successive stages, rather than a common clock pulse as in synchronous circuits.



# Procedure:

1.Create a New Project:

Open Quartus and create a new project by selecting "File" > "New Project Wizard."

Follow the wizard's instructions to set up your project, including specifying the project name, location, and target device (FPGA).

2.Create a New Design File:

Once the project is created, right-click on the project name in the Project Navigator and select "Add New File."

Choose "Verilog HDL File" or "VHDL File," depending on your chosen hardware description language.

1. Write the Combinational Logic Code:

Open the newly created Verilog or VHDL file and write the code for your combinational logic.

3.Compile the Project:

To compile the project, click on "Processing" > "Start Compilation" in the menu.

Quartus will analyze your code, synthesize it into a netlist, and perform optimizations based on your target FPGA device.

4.Analyze and Fix Errors:

If there are any errors or warnings during the compilation process, Quartus will display them in the Messages window.

Review and fix any issues in your code if necessary.

View the RTL diagram.

5.Verification:

Click on "File" > "New" > "Verification/Debugging Files" > "University Program VWF".

Once Waveform is created Right Click on the Input/Output Panel > " Insert Node or Bus" > Click on Node Finder > Click On "List" > Select All.

Give the Input Combinations according to the Truth Table and then simulate the Output Waveform

**Program :**

**UP counter:**

module upcounter(clk,q1,q2,q4);

input clk;

output reg q1,q2,q3,q4;

always@(posedge clk)

begin

q4=(q1&q2&q3)^q4

q3=(q1&q2)^q3;

q2=q1^q2;

q1=1^q1;

**program**

**Down counter:**

Module downcounter(clk,q1,q2,q3,q4);

Input clk;

Output regq1,q2,q3,q4;

always@(posedge clk)

begin

q4=((~q3)&(~q2)&(~q1))^q4);

q3=((~q2)&(~q1))^q3;

q2=(~q1)^q2;

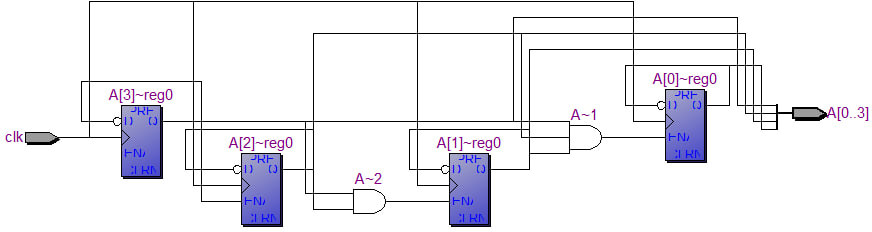
q1=1^q1;

end

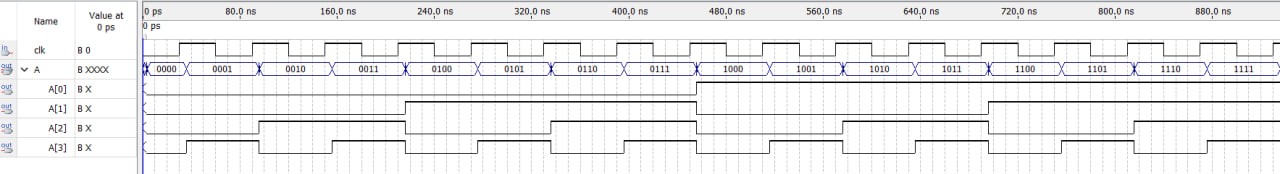
end module

Program for flipflops and verify its truth table in quartus using Verilog programme

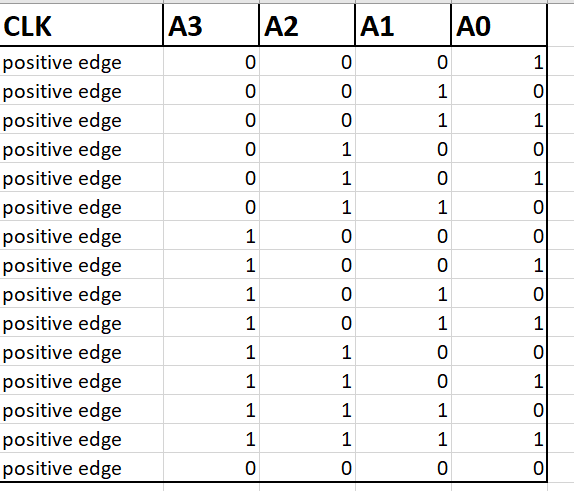
# RTL LOGIC UP COUNTER :



**TIMING DIGRAMS FOR UP COUNTER:**



**TRUTH TABLE FOR UP COUNTER:**



**RESULTS:**

By this we have verified the truth table of 4-bit up-counter using verilog.