EX 06:SYNCHORNOUS COUNTERS:UP COUNTER AND DOWN COUNTER

AIM:

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

HARDWARE REQUIRED:

1. PC
2. Cyclone II
3. 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

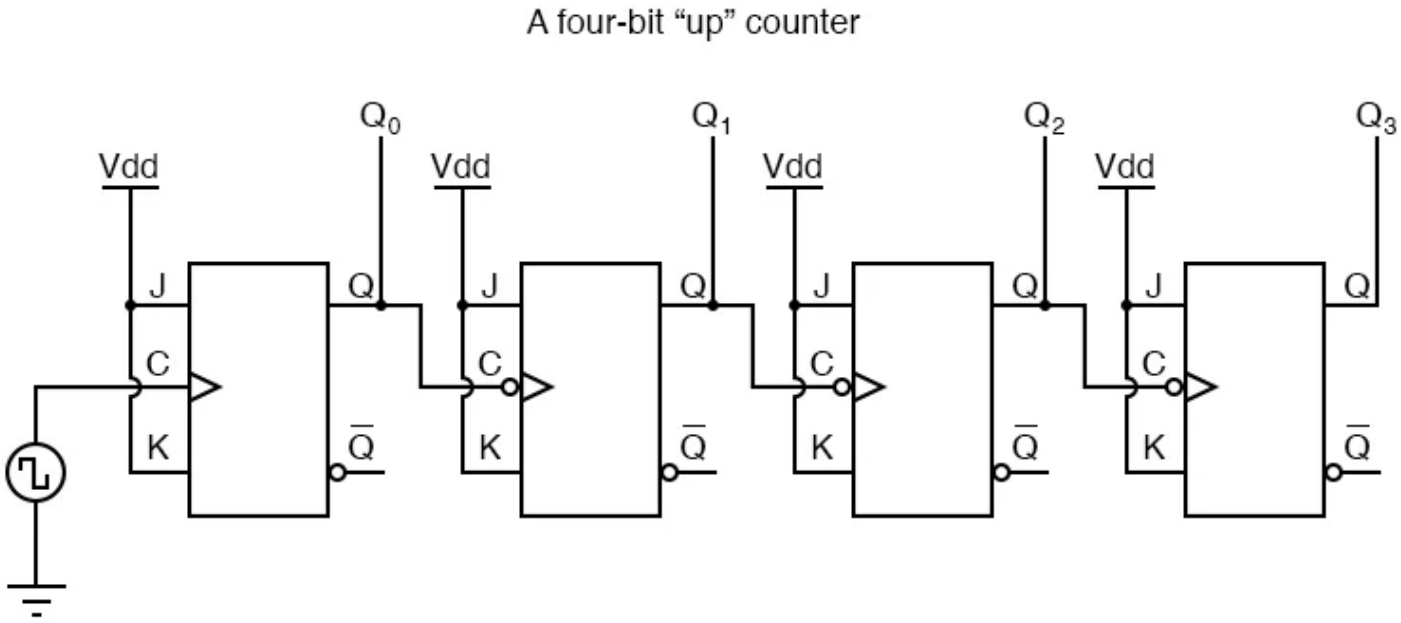
Since we know that binary count sequences follow a pattern of octave (factor of 2) frequency division, and that J-K flip-flop multivibrators set up for the “toggle” mode are capable of performing this type of frequency division, we can envision a circuit made up of several J-K flip-flops, cascaded to 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:

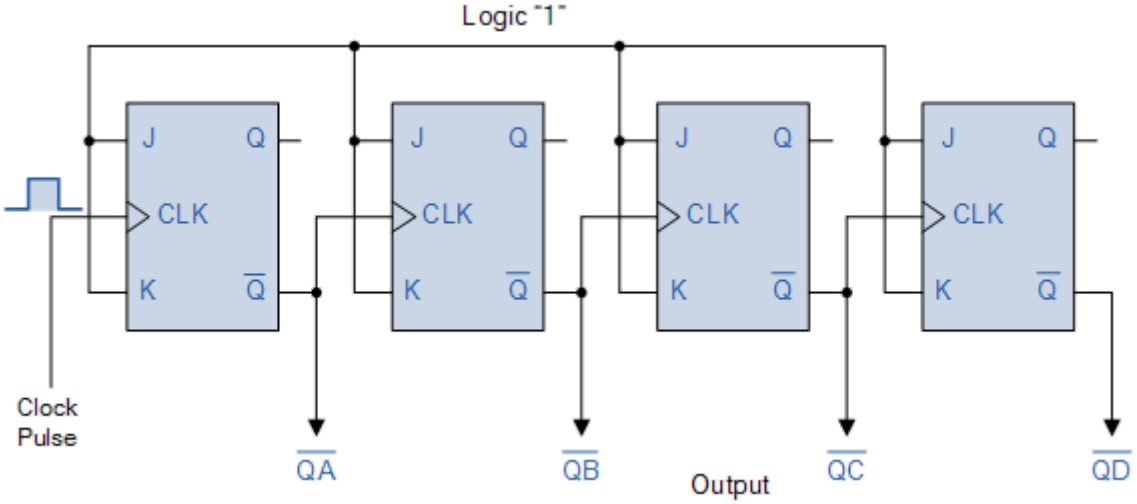
Four-bit “Up” Counter



DOWN COUNTER :

As well as counting “up” from zero and increasing or incrementing to some preset value, it is sometimes necessary to count “down” from a predetermined value to zero allowing us to produce an output that activates when the zero count or some other pre-set value is reached.

This type of counter is normally referred to as a Down Counter, (CTD). In a binary or BCD down counter, the count decreases by one for each external clock pulse from some preset value. Special dual purpose IC’s such as the TTL 74LS193 or CMOS CD4510 are 4-bit binary Up or Down counters which have an additional input pin to select either the up or down count mode.



4-bit Count Down Counter

PROCEDURE:

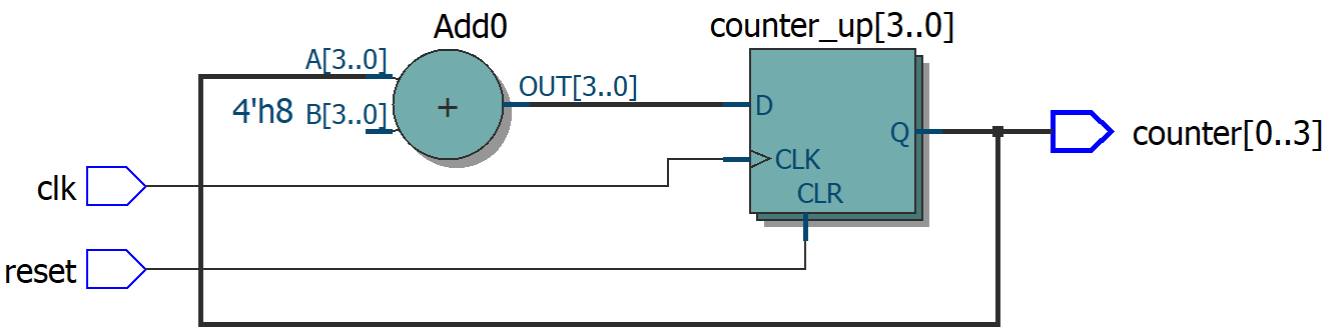
1. Open a new project using Quartus II.
2. Declare the inputs and outputs inside module projname().
3. Set the reset value using register.
4. Use commands like begin and end to stimulate the counter.
5. For Up counter increment the count and for Down counter decrement the count.
6. End the verilog programming.

UP COUNTER:

PROGRAM :

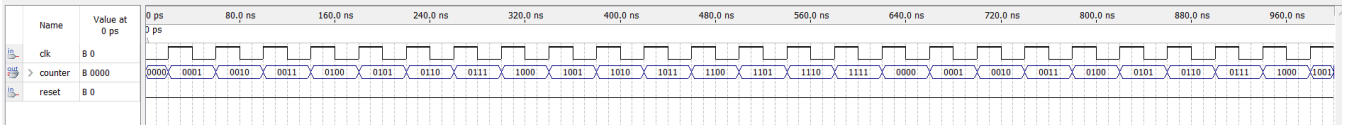
```
Program for Up Counter  and verify its truth table in quartus using Verilog programming.
Developed by: Rithiga Sri.B
RegisterNumber:  212221230083

module UC(input clk,input reset,output[0:3]counter);
reg[0:3]counter_up;
always@(posedge clk or posedge reset)
begin
if (reset)
counter_up<=4'd0;
else
counter_up<=counter_up+4'd1;
end
assign counter=counter_up;
endmodule
```

RTL LOGIC REALIZATION:

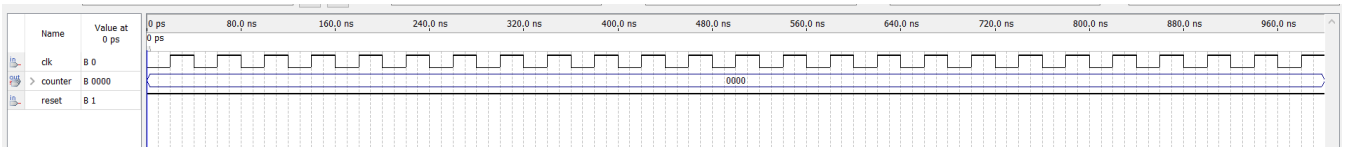


TIMING DIGRAMS:

- With Reset = 0



- With Reset = 1



TRUTH TABLE:

A	B	C	D	COUNT NUMBER
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

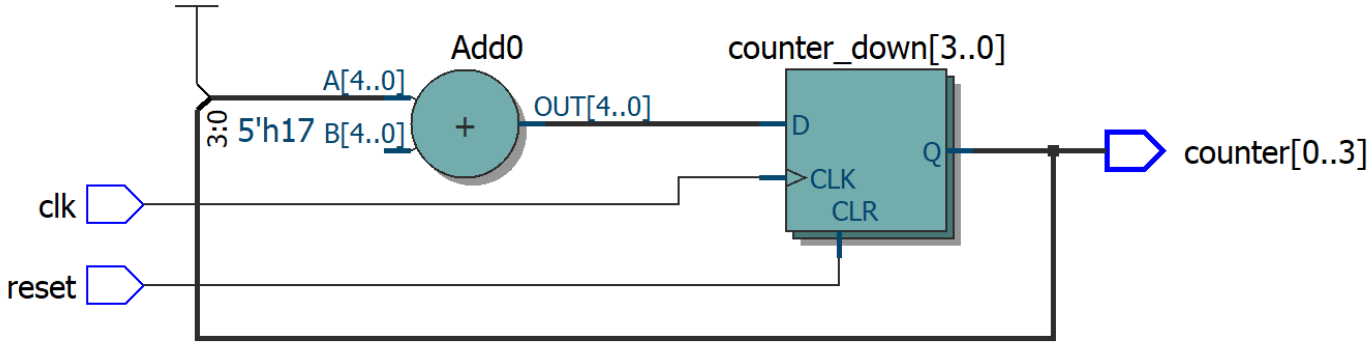
DOWN COUNTER:

PROGRAM :

```
Program for Down Counter  and verify its truth table in quartus using Verilog
programming.
Developed by: Rithiga Sri.B
RegisterNumber:  212221230083

module DC(input clk,input reset,output[0:3]counter);
reg[0:3]counter_down;
always@(posedge clk or posedge reset)
begin
if (reset)
counter_down<=4'd0;
else
```

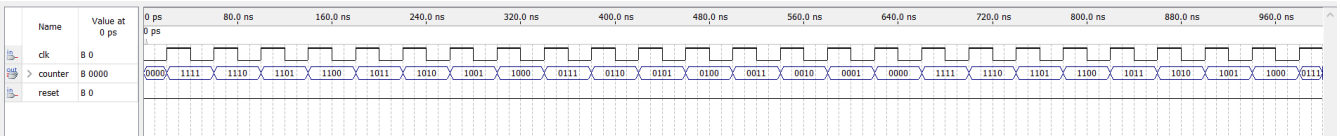
```
counter_down<=counter_down-4'd1;
end
assign counter=counter_down;
endmodule
```

RTL LOGIC REALIZATION:

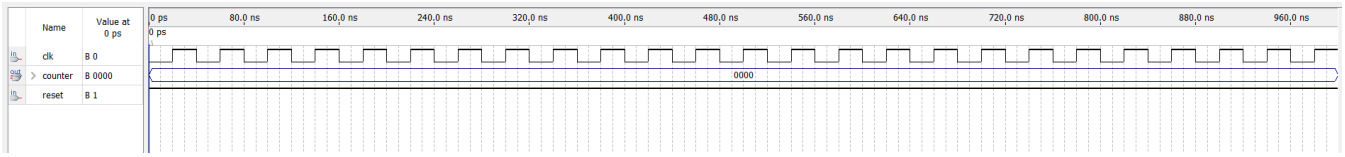


TIMING DIGRAMS:

- With Reset = 0



- With Reset = 1



TRUTH TABLE:

A	B	C	D	COUNT NUMBER
1	1	1	1	15
1	1	1	0	14
1	1	0	1	13
1	1	0	0	12
1	0	1	1	11
1	0	1	0	10
1	0	0	1	9
1	0	0	0	8
0	1	1	1	7
0	1	1	0	6
0	1	0	1	5
0	1	0	0	4
0	0	1	1	3
0	0	1	0	2
0	0	0	1	1
0	0	0	0	0

RESULT:

Hence the Four bit Up counter and Down counter is implemented successfully and its functionality is validated.