EXP11: SYNCHRONOUS UP COUNTER

SYNCHRONOUS UP COUNTER

AIM:

To implement 4 bit synchronous up counter and validate functionality.

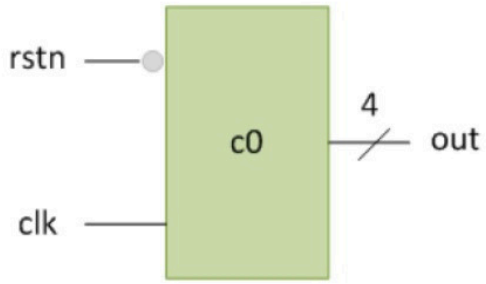
SOFTWARE REQUIRED:

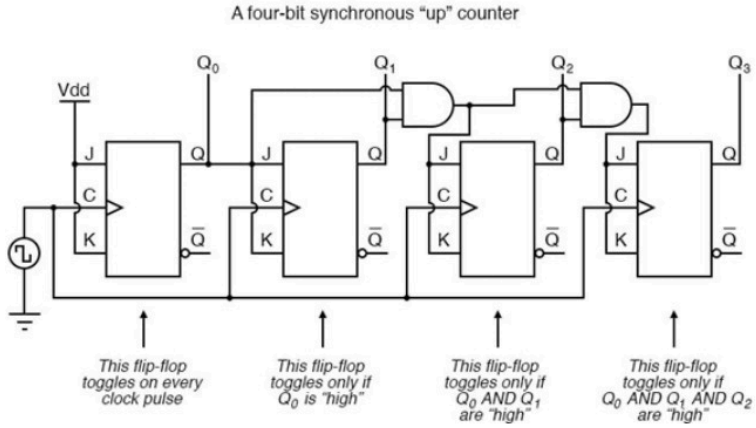
Quartus prime

THEORY:

4 bit synchronous UP Counter

If we enable each J-K flip-flop to toggle based on whether or not all preceding flip-flop outputs (Q) are “high,” we can obtain the same counting sequence as the asynchronous circuit without the ripple effect, since each flip-flop in this circuit will be clocked at exactly the same time:





Each flip-flop in this circuit will be clocked at exactly the same time. The result is a four-bit synchronous “up” counter. Each of the higher-order flip-flops are made ready to toggle (both J and K inputs “high”) if the Q outputs of all previous flip-flops are “high.” Otherwise, the J and K inputs for that flip-flop will both be “low,” placing it into the “latch” mode where it will maintain its present output state at the next clock pulse. Since the first (LSB) flip-flop needs to toggle at every clock pulse, its J and K inputs

are connected to Vcc or Vdd, where they will be “high” all the time. The next flip-flop need only “recognize” that the first flip-flop’s Q output is high to be made ready to toggle, so no AND gate is needed. However, the remaining flip-flops should be made ready to toggle only when all lower-order output bits are “high,” thus the need for AND gates.

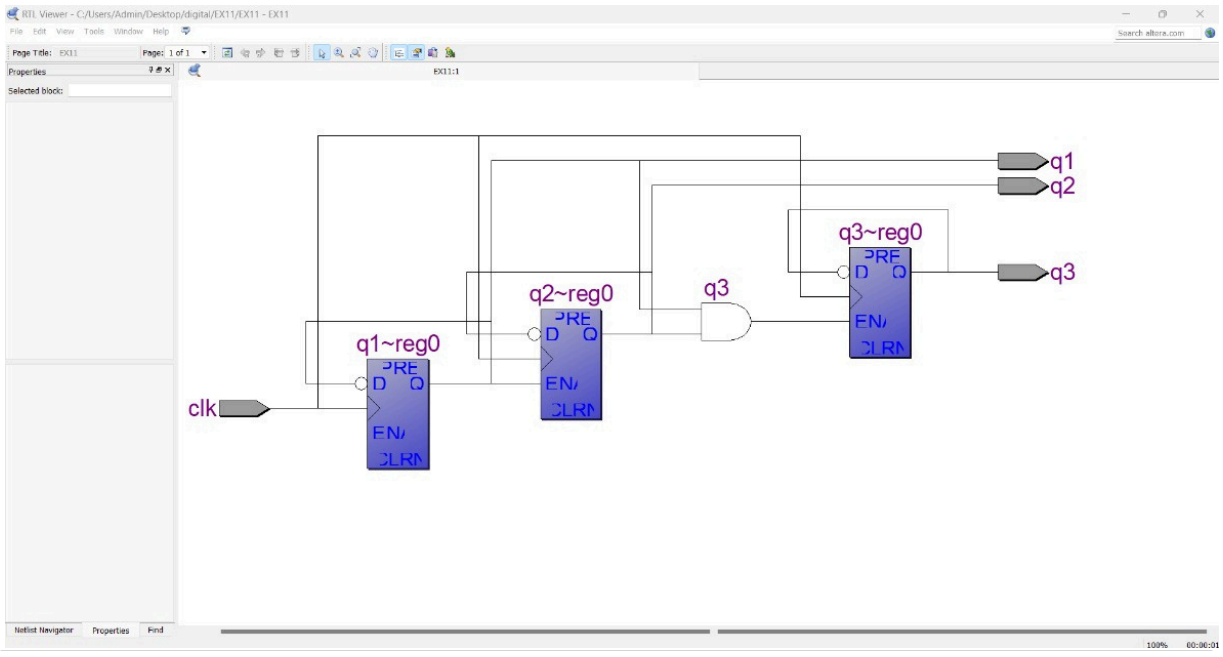
Procedure:

1. Type the program in Quartus software.
2. Compile and run the program.
3. Generate the RTL schematic and save the logic diagram.
4. Create nodes for inputs and outputs to generate the timing diagram.
5. For different input combinations generate the timing diagram.

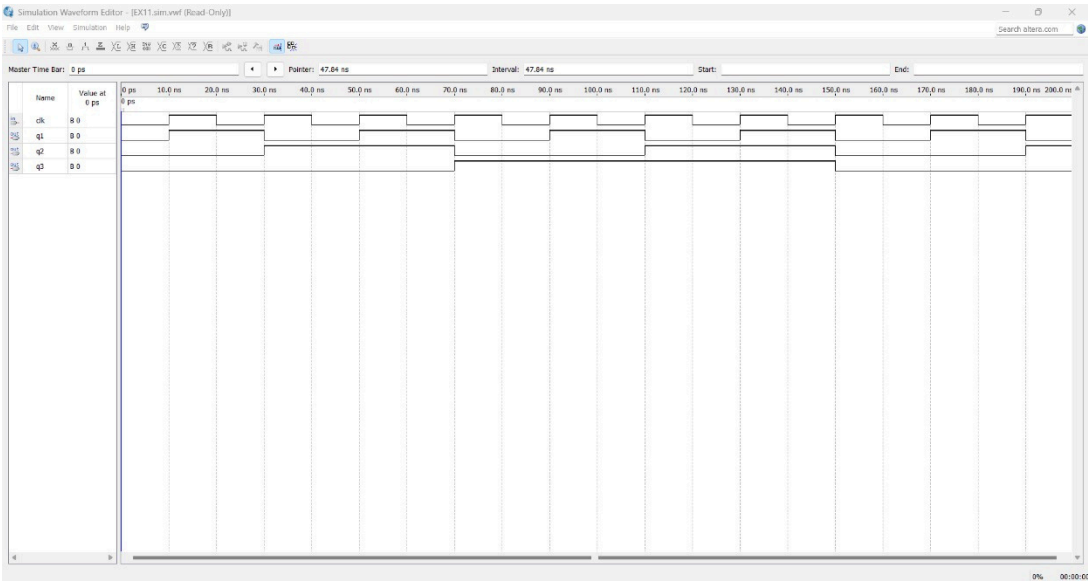
PROGRAM:

```
1     module EX11(clk,q1,q2,q3);
2     input clk;
3     output reg q1,q2,q3;
4     always@(posedge clk)
5     begin
6     q3 = (q1&q2)^q3;
7     q2 = q1^q2;
8     q1 = 1^q1;
9     end
10    endmodule
```

RTL LOGIC UP COUNTER:



TIMING DIAGRAM FOR UP COUNTER:



TRUTH TABLE:

CK	Q_3	Q_2	Q_1	Q_0	$\overline{Q_3}$	$\overline{Q_2}$	$\overline{Q_1}$	$\overline{Q_0}$
0	0	0	0	0	1	1	1	1
1	0	0	0	1	1	1	1	0
2	0	0	1	0	1	1	0	1
3	0	0	1	1	1	1	0	0
4	0	1	0	0	1	0	1	1
5	0	1	0	1	1	0	1	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	0	0
8	1	0	0	0	0	1	1	1
9	1	0	0	1	0	1	1	0
10	1	0	1	0	0	1	0	1
11	1	0	1	1	0	1	0	0
12	1	1	0	0	0	0	1	1
13	1	1	0	1	0	0	1	0
14	1	1	1	0	0	0	0	1
15	1	1	1	1	0	0	0	0

RESULTS:

The 4-bit synchronous up counter implemented in Verilog has been successfully completed. Its functionality has been validated through simulation, confirming that the counter increments by one on each clock cycle and correctly transitions through the count sequence from 0000 to 1111, as expected for an up counter. The counter design adheres to synchronous principles, ensuring reliable and accurate counting behavior.

Developed by: Rhudhresh

Register Number: 212223050039