

|  |
| --- |
|  |
| **Experiment / Assignment / Tutorial No. 8** |
|  |
| **Grade: AA / AB / BB / BC / CC / CD /DD** |
|  |
| **Signature of the Staff In-charge with date** |



**Batch:B4 Roll No.:16010122221 Experiment / assignment / tutorial No.: 8**

**Title:** 3-bit Asynchronous and Synchronous Counter

**Objective:** Design of 3 bit asynchronous counter using JK flip flop in VHDL

# Expected Outcome of Experiment:

**CO2:** Use different minimization technique and solve combinational circuits, synchronous & asynchronous sequential circuits.

**CO4:** Implement digital networks using VHDL

# Books/ Journals/ Websites referred:

* R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill
* M .Morris Mano, “Digital Logic & computer Design”, PHI
* A.P.Godse, D.A.Godse, “Digital Logic Design” <http://www.fatih.edu.tr/~aliadam/EEE122A/EEE122Ch6COUNTERS.pdf>
* ModelSim Software Link:

https:/[/www.mentor.com/company/higher\_ed/m](http://www.mentor.com/company/higher_ed/modelsim-student-edition)o[delsim-student-edition](http://www.mentor.com/company/higher_ed/modelsim-student-edition)

* J. Bhasker, “VHDL Primer”, Pearson Education
* Douglas L. Perry, “VHDL Programming by Example”, Tata McGraw Hill
* <http://esd.cs.ucr.edu/labs/tutorial/>

# Pre Lab/ Prior Concepts:

A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. A specified sequence of states appears as counter output. This is the main difference between a register and a counter. There are two types of counter, synchronous and asynchronous. In synchronous common clock is given to all flip flop and in asynchronous first flip flop is clocked by external pulse and then each successive flip flop is clocked by Q or Q output of previous stage. A soon the clock of second stage is triggered



by output of first stage. Because of inherent propagation delay time all flip flops are not activated at same time which results in asynchronous operation.

# Implementation Details:

*Code:*

library IEEE;

use IEEE.std\_logic\_1164.all; entity up\_counter is

port ( CLK : in std\_logic; -- Clock input CLK active low

T : in std\_logic; -- Asynchronous clear input CLR active high Q: out std\_logic\_vector(2 downto 0)); end up\_counter;

architecture up\_counter\_arch of up\_counter is shared variable tmp: std\_logic\_vector(2 downto 0):=(others=>'0'); signal tmp\_out: std\_logic\_vector(2 downto 0):=(others=>'0');

begin process(CLK) is

begin

if(rising\_edge(clk)) and (T='1') then tmp(0):=not tmp(0); tmp\_out(0)<=tmp(0); end if; end process;

process(tmp\_out(0)) is begin

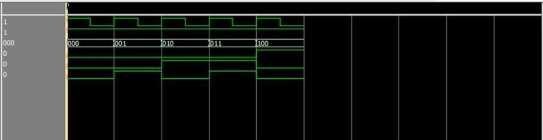
if(falling\_edge(tmp\_out(0)) and (T='1')) then tmp(1):=not tmp(1);

tmp\_out(1)<=tmp(1); end if; end process; process(tmp\_out(1)) is begin

if(falling\_edge(tmp\_out(1)) and (T='1')) then tmp(2):=not tmp(2);

tmp\_out(2)<=tmp(2); end if; end process; Q<=tmp\_out; end up\_counter\_arch;

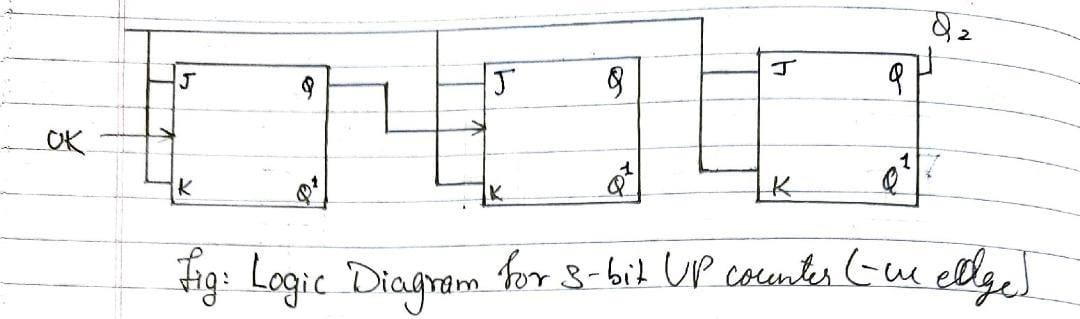




# Truth Table for 3 bit Asynchronous UP counter

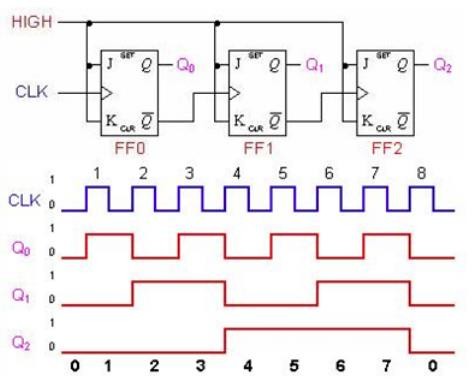
|  |  |  |  |
| --- | --- | --- | --- |
| **states** | **QC** | **QB** | **QA** |
| **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** |
| **2** | **0** | **1** | **0** |
| **3** | **0** | **1** | **1** |
| **4** | **1** | **0** | **0** |
| **5** | **1** | **0** | **1** |
| **6** | **1** | **1** | **0** |
| **7** | **1** | **1** | **1** |

**Logic Diagram for 3 bit UP counter (Negative edge)**





# Timing Diagram for 3 bit Asynchronous UP counter (Negative edge)

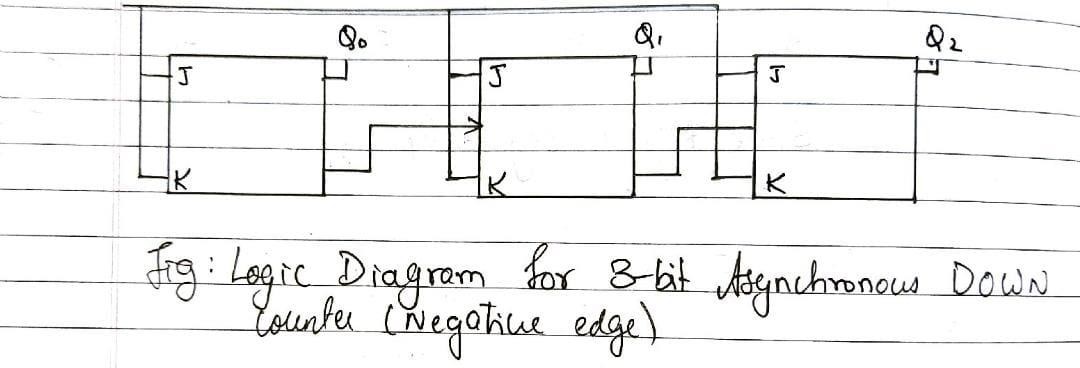


**Truth Table for 3 bit Asynchronous DOWN counter**

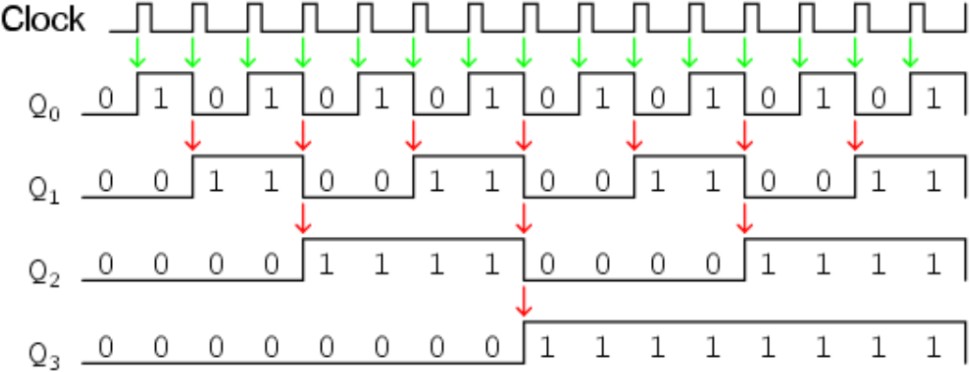
|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Q2** | **Q1** | **Q0** |
| **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** |
| **2** | **0** | **1** | **0** |
| **3** | **0** | **1** | **1** |
| **4** | **1** | **0** | **0** |
| **5** | **1** | **0** | **1** |
| **6** | **1** | **1** | **0** |
| **7** | **1** | **1** | **1** |



# Logic Diagram for 3 bit Asynchronous DOWN counter (Negative edge)



**Timing Diagram for 3 bit Asynchronous DOWN counter**





# Characteristic Table for 3 bit Synchronous UP counter

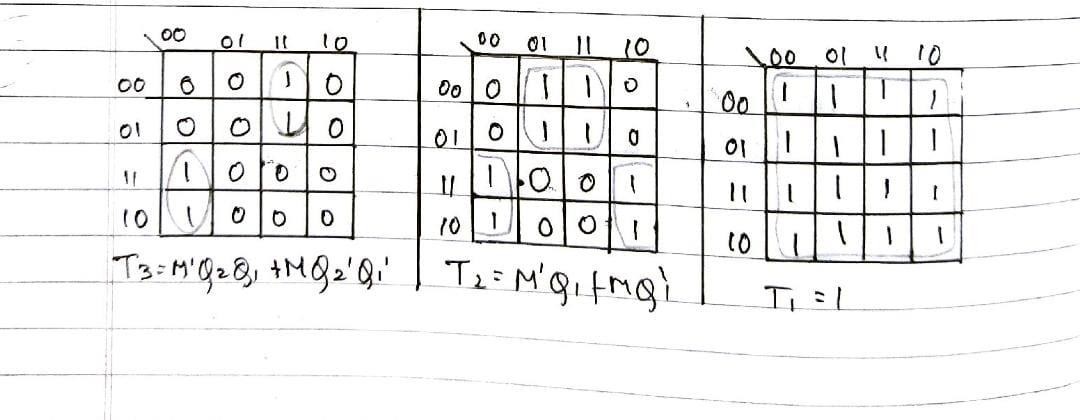
|  |  |  |  |
| --- | --- | --- | --- |
| **Q** | **Qt+1** | **J** | **K** |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 0 |
| 1 | 1 | X | 1 |

**Truth Table for 3 bit Synchronous UP Counter**

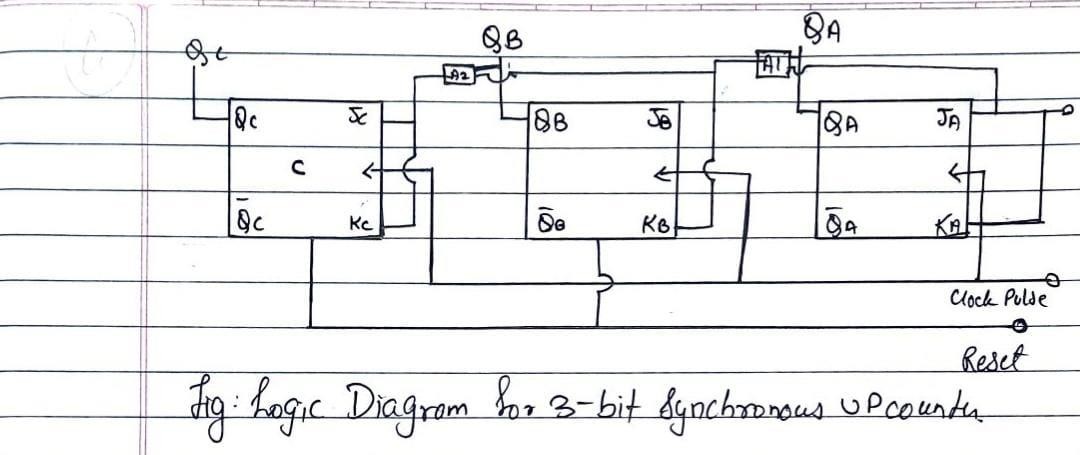
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Present State QA QB QC** | **Next State QA+1 Q B+1 QC+1** | **A**  **JA KA** | **B**  **JB KB** | **C**  **JC KC** |
| 000 | 001 | 0X | 0X | 1X |
| 001 | 011 | 0X | 1X | X0 |
| 010 | 110 | 1X | X0 | 0X |
| 011 | 010 | 0X | X0 | X1 |
| 100 | 000 | X1 | 0X | 0X |
| 101 | 100 | X0 | 0X | X1 |
| 110 | 111 | X0 | X0 | 1X |
| 111 | 101 | X0 | X1 | X0 |



# K Map

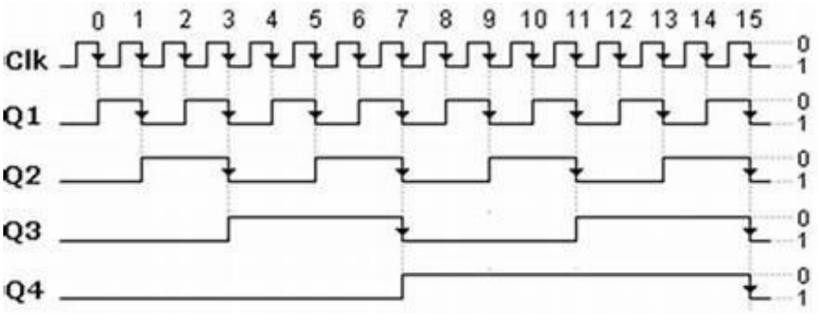


**Logic Diagram for 3 bit Synchronous UP counter**





# Timing Diagram for 3 bit Synchronous UP counter

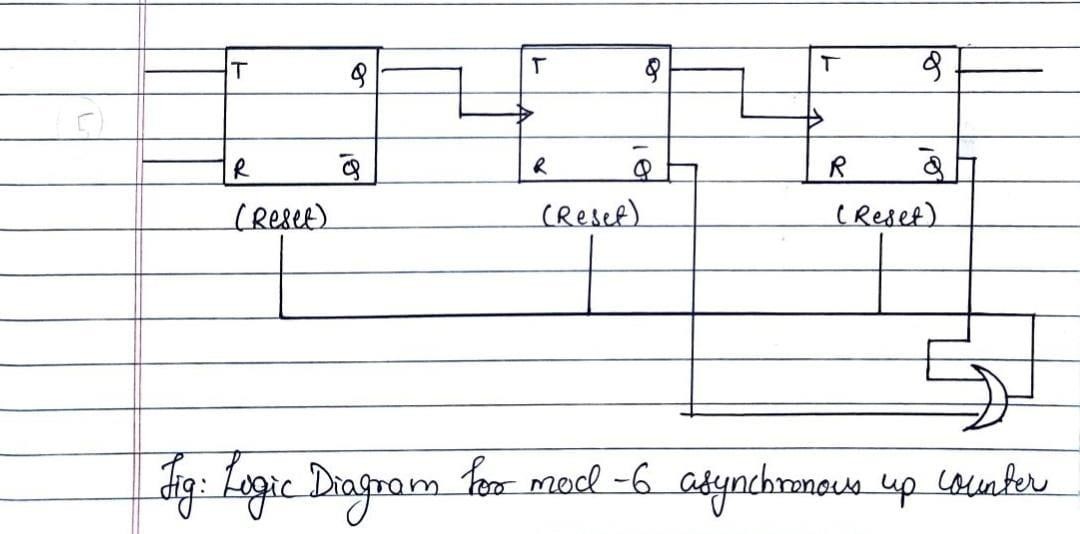


**Conclusion:**

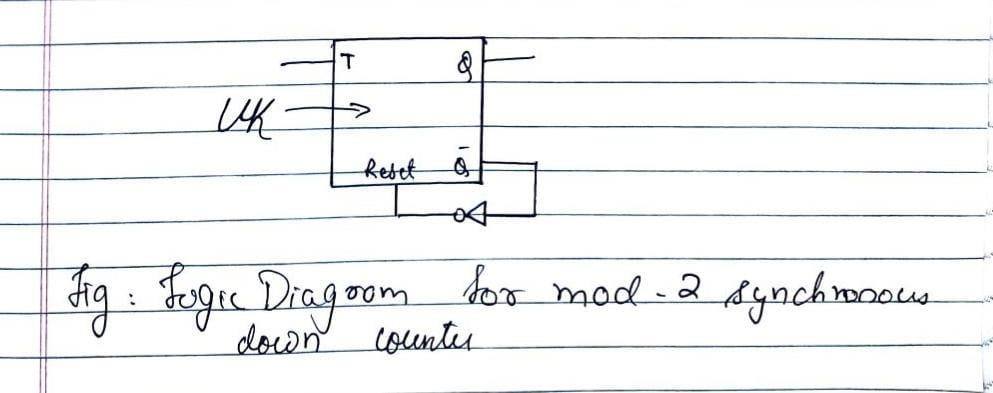
In this experiment, we have designed 3 bit synchronous and asynchronous counter using VHDL Code.

# Post Lab Descriptive Questions

1. Draw logic diagram for mod – 6 asynchronous up counter.





1. Draw logic diagram for mod-2 synchronous down counter.