**Experiment / Assignment / Tutorial No. 8**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Batch: A1 Roll No.: 1911004 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/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:**

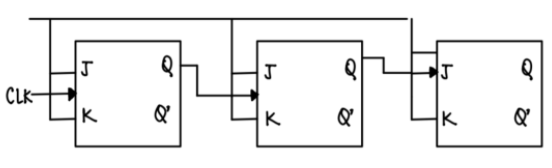
A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. A specified sequence of states appears as counter output. This is the main difference between a register and a counter. There are two types of counter, 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:**

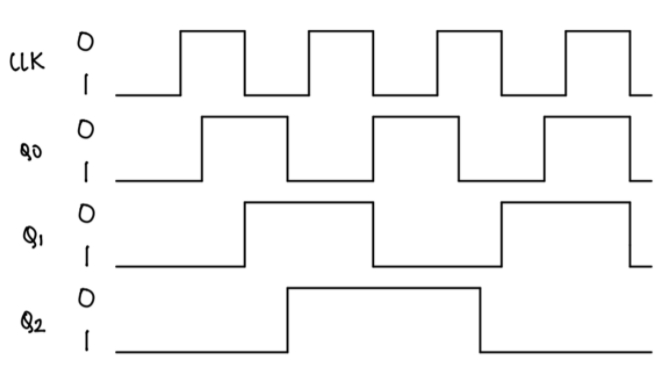
**Truth Table for 3 bit Asynchronous UP counter**

|  |  |  |  |
| --- | --- | --- | --- |
| Clock Input | OUTPUT | | |
| Count | 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)**



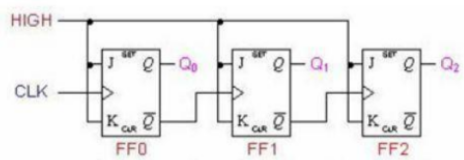
**Timing Diagram for 3 bit Asynchronous UP counter (Negative edge)**



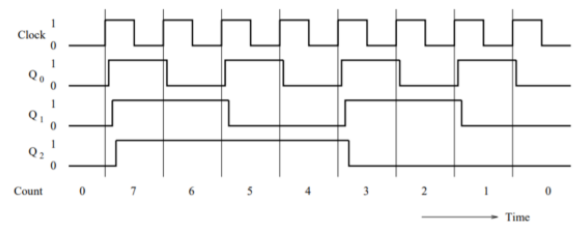
**Truth Table for 3 bit Asynchronous DOWN counter**

|  |  |  |  |
| --- | --- | --- | --- |
| Clock Input | OUTPUT | | |
| Count | QC | QB | QA |
| 7 | 1 | 1 | 1 |
| 6 | 1 | 1 | 0 |
| 5 | 1 | 0 | 1 |
| 4 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 |

**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 | 1 |
| 1 | 1 | x | 0 |

**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** |
| 0 0 0 | 0 0 1 | 1 X | 0 X | 0 X |
| 0 0 1 | 0 1 0 | X 1 | 1 X | 0 X |
| 0 1 0 | 0 1 1 | 1 X | X 0 | 0 X |
| 0 1 1 | 1 0 0 | X 1 | X 1 | 1 X |
| 1 0 0 | 1 0 1 | 1 X | 0 X | X 0 |
| 1 0 1 | 1 1 0 | X 1 | 1 X | X 0 |
| 1 1 0 | 1 1 1 | 1 X | X 0 | X 0 |
| 1 1 1 | 0 0 0 | X 1 | X 1 | 1 X |

**K Map**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Q0 | |
| Q2 | Q1 | 0 | 1 |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | X | X |
| 1 | 1 | X | X |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Q0 | |
| Q2 | Q1 | 0 | 1 |
| 0 | 0 | 0 | 1 |
| 0 | 1 | X | X |
| 1 | 0 | X | X |
| 1 | 1 | 0 | 1 |

**J2 J1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Q0 | |
| Q2 | Q1 | 0 | 1 |
| 0 | 0 | 1 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | 1 | X |
| 1 | 1 | 1 | X |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Q0 | |
| Q2 | Q1 | 0 | 1 |
| 0 | 0 | X | X |
| 0 | 1 | X | X |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

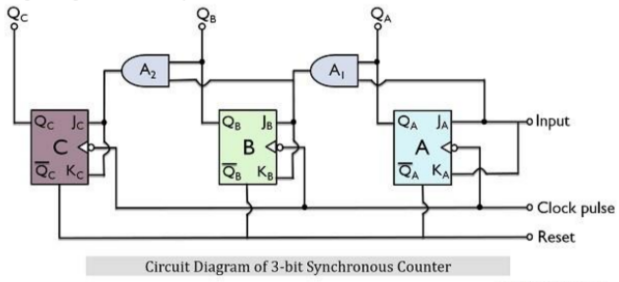
**J0 K2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Q0 | |
| Q2 | Q1 | 0 | 1 |
| 0 | 0 | X | X |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | X |

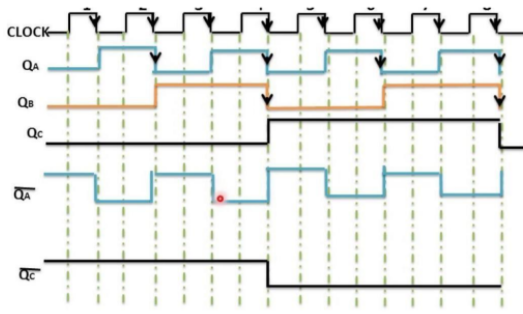
|  |  |  |  |
| --- | --- | --- | --- |
|  | | Q0 | |
| Q2 | Q1 | 0 | 1 |
| 0 | 0 | X | 1 |
| 0 | 1 | X | 1 |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 1 |

**K1 K0**

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



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



**Code:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity jkc is

Port ( clock : in std\_logic;

reset : in std\_logic;

count : out std\_logic\_vector(2 downto 0)

);

end jkc;

architecture rtl of jkc is

COMPONENT jkff

PORT(

clock : in std\_logic;

reset : in std\_logic;

j : in std\_logic;

k : in std\_logic;

q : out std\_logic

);

END COMPONENT;

signal temp : std\_logic\_vector(2 downto 0) := "000";

begin

d0 : jkff

port map (

reset => reset,

clock => clock,

j => '1',

k => '1',

q => temp(2) );

d1 : jkff

port map (

reset => reset,

clock => temp(2),

j => '1',

k => '1',

q => temp(1) );

d2 : jkff

port map (

reset => reset,

clock => temp(1),

j => '1',

k => '1',

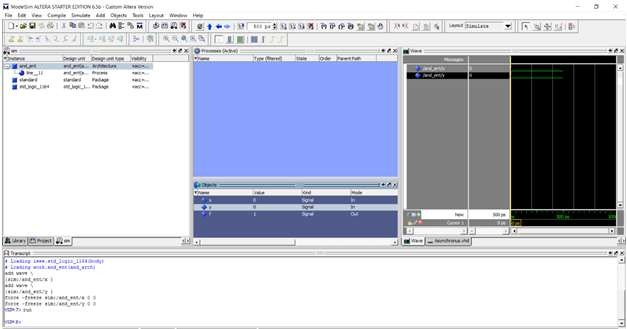
q => temp(0) );

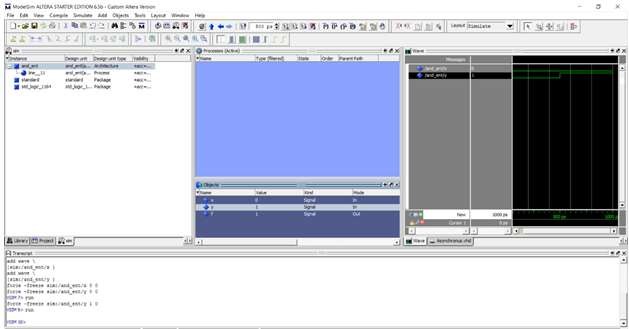
count(2) <= temp(0);

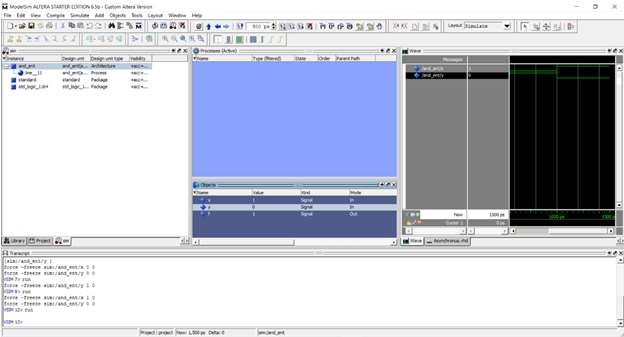
count(1) <= temp(1);

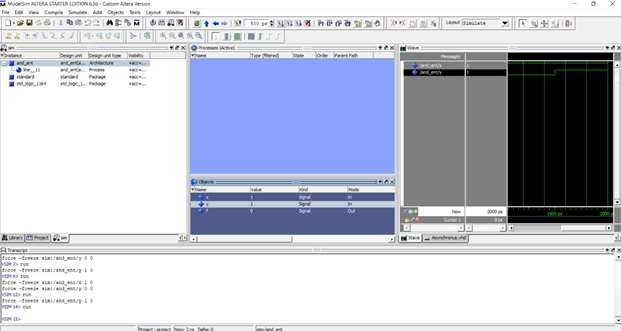
count(0) <= temp(2);

end rtl;







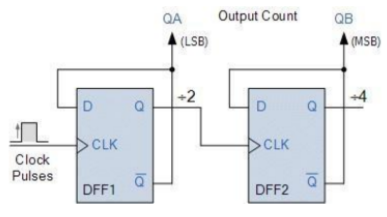


**Conclusion:**

We understood the concept of synchronous – asynchronous binary counters with its logic diagram & truth table; implemented it using VHLD programming.

**Post Lab Descriptive Questions**

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



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

