LAB - 8 Date: 1st April,2020

Lab Title: Ring Counters

Recorded and Submitted by: Bhesaniya Abhishy

Objective: To write VHDL Program for ring counters, Overbacks counter and Twisted back counters.

Implimentation

We will use two switch buttons to make counters one for count up and one for reset the counter.

we will use 4 led for ring counter and 4 led for JHON counter to show output.

~~Hex~~ Seven segment display will display the states ~~counts~~. Hex0 for ring counter and Hex2 for JHON.

Procedure:

- 1st step is to make table for output for correspondin input.
- then define inputs and outputs in entity section
- we will use signal for both input and output.

```
Signal RING : std-logic-vector(3 downto 0);
signal JHON : std-logic-vector(3 downto0);
```

- whene evere there is a reset occurs Load both the counters with default Values.
- We will use case statment to display the state on seven segment LED.

- After writting entire code we will compile the code and will do pin assignment.

- After doing pin assignment we will again compile this Project.

- Then we will load the program in to FBGA board and will test for different case.

Table for Straight ring / overbeck Counter

State	Q_0	Q_1	Q_2	Q_3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
0	1	0	0	0

Table for Twisted ring / Johnson Counter

State	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
0	0	0	0	0

Pin Assignment

Input / Output Variables	Signal Name	FBGA Pin no.
EN	SW [9]	PIN - AE12
CLEAR	KEY [0]	PIN - AA14
Clock	KEY [1]	PIN - AA15
J[3]	LEDR [9]	PIN - Y21
J[2]	LEDR [8]	PIN - W21
J[1]	LEDR [7]	PIN - W20
J [0]	LEDR [6]	PIN - Y19
R[3]	LEDR [3]	PIN - V18
R[2]	LEDR [2]	PIN - V17
R[1]	LEDR[1]	PIN - W16
R [0]	LEDR[0]	PIN - V16
HEX0[6]	HEX0 [0]	PIN - AE26
HEX0[5]	HEX0 [1]	PIN - AE27
HEX0 [4]	HEX0 [2]	PIN - AE28
HEX0 [3]	HEX0 [3]	PIN - AG27
HEX0 [2]	HEX0 [4]	PIN - AF28
HEX0 [1]	HEX0 [5]	PIN - AG28
HEX0 [0]	HEX0 [6]	PIN - AH28
HEX1 [6]	HEX2 [0]	PIN - AB28
HEX1 [5]	HEX2 [1]	PIN - AE29
HEX1 [4]	HEX2 [2]	PIN - AD29
HEX1 [3]	HEX2 [3]	PIN - AC28
HEX1 [2]	HEX2 [4]	PIN - AD30
HEX1 [1]	HEX2 [5]	PIN - AC29
HEX 1 [0]	HEX2 [6]	PIN - AC30

VHDL Program:-

```
library ieee;
use ieee.std-logic-1164.all;
use ieee.numeric-std.all;

entity abhesaniyelab8 is
Port (
EN : in std-logic;
CLEAR : in std-logic;
Clock : in std-logic;
R : out std-logic-vector(3 downto 0);
J : out std-logic-vector(3 downto 0);
HEX0 : out std-logic-vector(6 downto 0);
HEX1 : out std-logic-vector(6 downto 0);
);
end abhesaniyelab8;

architecture COUNTER of abhesaniyelab8 is

Signal RING : std-logic-vector(3 downto 0);
signal JHON : std-logic-vector(3 downto 0);

begin

        Process (clock, EN, CLEAR)
        begin

        if (EN = '1')
        then
                if CLEAR = '0' then
                        RING <= "0001";
                        JHON <= "0000";
```

```
else if Rising - edge (clock ) then
        RING(1) <= RING (0) ;
        RING(2) <= RING (1) ;
        RING(3) <= RING (2) ;
        RING (0) <= RING (3) ;
        JHON (1) <= ~~JHON~~ JHON (0);
        JHON (2) <= ~~RING~~ JHON (1);
        JHON (3) <= R JHON (2);
        JHON (0) <= NOT JHON (3);

          R <= RING;
          J <= JHON;

    Case RING is
        when "0001" => HEX0 <= "0000001";
        when "0010" => HEX0 <= "1001111";
        when "0100" => HEX0 <= "0010010";
        when "1000" => HEX0 <= "0000110";
        when others => HEX0 <= "1111111";
    end case;

    Case JHON is

          when "0000" => HEX1 <= "0000001";
          when "0001" => HEX1 <= "1001111";
          when "0011" => HEX1 <= "0010010";
          when "0111" => HEX1 <= "0000110";
          when "1111" => HEX1 <= "1001100";
          when "1110" => HEX1 <= "0100100";
          when "1100" => HEX1 <= "0100000";
          when "1000" => HEX1 <= "0001111";
          when Others => HEX1 <= "1111111";
          when Others

     End case;

     End if;
```

```
        else
            RING <= "0001";
            JHON <= "0000";
            HEX0 <= "1111111";
            HEX1 <= "1111111";
        end if
        end process;

    End COUNTER;
```

Discussion :