INSTITUTE OF ENGINEERING AND TECHNOLOGY, INDORE

EIR7E1: Circuit Design Using VHDL

“Design of Synchronous 4-bit Counter “

Submitted To:

Dr. Vaibhav Neema

Submitted By:

Shivanshu Mishra

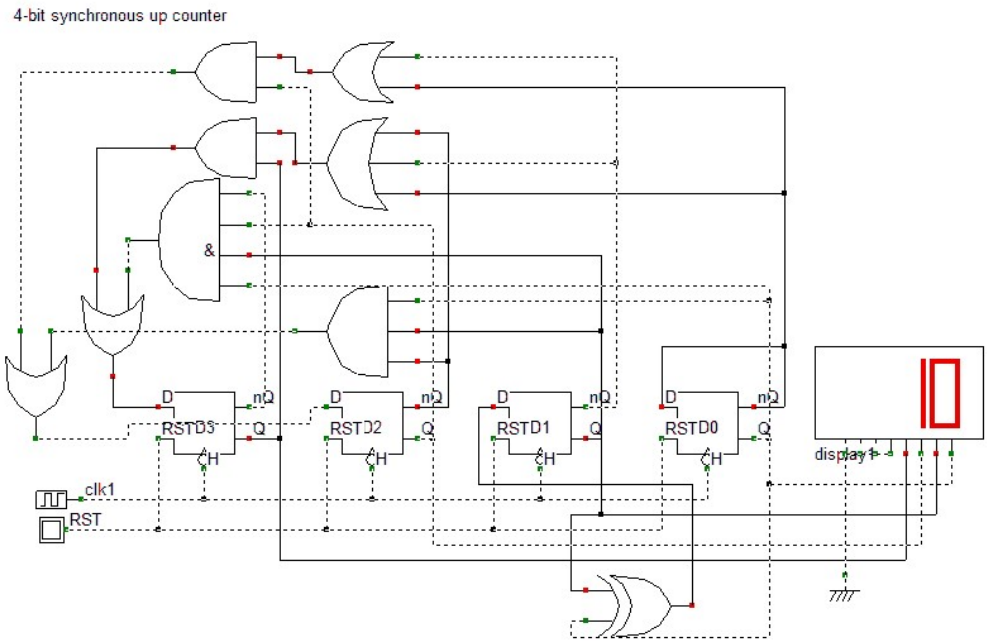
17E7046

Synchronous Counters:

Counters are sequential circuits that employ a cascade of flip-flops that are used to count something. There are three types of synchronous counters: up, down, and up-down.

Up-counter-

Synchronous means to be driven by the same clock. The flip-flops in the synchronous counters are all driven by a single clock input. You can see the logic circuit of the 4-bit synchronous up-counter. It has two inputs of STD_LOGIC, Clock and Reset. And four outputs since its a 4-bit counter.



VHDL Code for Synchronous 4-bit Counter

DESIGN CODE

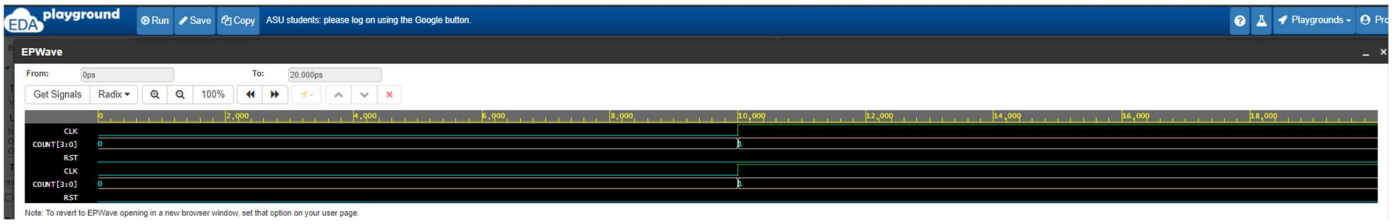
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity SOURCE is
   Port (CLK, RST: in STD_LOGIC;
         COUNT: inout STD_LOGIC_VECTOR (3 downto 0)
        );
end SOURCE;

architecture Behavioral of SOURCE is

begin
        process (CLK, RST)
                begin

                        if (RST = '1') then
                        COUNT <= "0000";
                        Elsif (rising_edge (CLK)) then
                        COUNT <= COUNT+1;

                end if;
        end process;
end Behavioral;
```

TESTBENCH CODE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity sync_upcounter_tb is
end sync_upcounter_tb;

architecture tb of sync_upcounter_tb is
component sync_upcounter is
        Port (CLK, RST: in STD_LOGIC;
        COUNT: inout STD_LOGIC_VECTOR (3 downto 0)
        );
end component;

signal CLK, RST: STD_LOGIC: = '1';
signal COUNT: STD_LOGIC_VECTOR (3 downto 0);

begin

        uut: sync_upcounter port map (
                CLK => CLK,
                RST => RST,
                COUNT => COUNT
        );

clock: process
begin
```

```
        RST <= '0';

        CLK <= '0';
        wait for 10 ns;
        CLK <= '1';
        wait for 10 ns;
        wait;
end process;
end tb;
```

SIMULATION WAVEFORM



Website to visit the code:-

https://www.edaplayground.com/x/QBaC