Counter 4-bit

Name: dhruv subandh Roll No.:7

Code:

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

entity Counter_4 is
    port(
        rst : in  STD_LOGIC;
        pr :  in STD_LOGIC;
        clk : in  STD_LOGIC;
        clk_div : inout STD_LOGIC;
        dir : in  STD_LOGIC;
        Q : out  STD_LOGIC_VECTOR (3 downto 0)
    );
end Counter_4;

architecture Behavioral of Counter_4 is
    signal Qtemp : STD_LOGIC_VECTOR (3 downto 0) := "0000";
    signal Counter : STD_LOGIC_VECTOR (26 downto 0) := (others => '0');
begin
    process(rst,pr,clk_div,dir)
    begin
        if rst ='1' then
            Qtemp <= (OTHERS =>'0');
        elsif pr='1' then
            Qtemp <= (OTHERS =>'1');
        elsif falling_edge(clk_div) then
            if dir = '1' then
```

```
            if Qtemp < 15 then
               Qtemp <= Qtemp + 1;
            else
               Qtemp <= "0000";
            end if;
        else
           if Qtemp > 0 then
               Qtemp <= Qtemp - 1;
           else
               Qtemp <= "1111";
           end if;
        end if;
    end if;
    end process;
    Q <= Qtemp;

    process(clk)
       begin
          if clk'event and clk = '1' then
             counter <= counter + 1;
          end if;
    end process;
       clk_div <= counter(0);
end Behavioral;
```

TestBench:

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

```
entity Counter_4_TB is
end Counter_4_TB;

architecture Behavioral of Counter_4_TB is
component Counter_4 is
    port(
        rst : in  STD_LOGIC;
        pr :  in STD_LOGIC;
        clk : in  STD_LOGIC;
        clk_div : inout std_logic;
        dir : in  STD_LOGIC;
        Q : out  STD_LOGIC_VECTOR (3 downto 0)
    );
end component;

    signal rst : std_logic := '0';
    signal pr  : std_logic := '0';
    signal clk : std_logic := '0';
    signal clk_div : std_logic := '0';
    signal dir : std_logic := '0';
    signal Q : std_logic_vector(3 downto 0);

begin
    uut: Counter_4 port map(rst, pr, clk, clk_div, dir, Q);

    clk_process : process
    begin
        clk <= '0';
        wait for 10 ns;
        clk <= '1';
```

```
        wait for 10 ns;
    end process;

    stim_proc : process
    begin
        rst <= '1';
        wait for 20 ns;
        rst <= '0';

        --Up Counter
        dir <= '1';
        wait for 500 ns;

        pr <= '1';
        wait for 20 ns;
        pr <= '0';

        --Down Counter
        dir <= '0';
        wait for 300 ns;

        wait;
    end process;
end Behavioral;
```

Constraints:

```
set_property PACKAGE_PIN L1 [get_ports {Q[3]}]
set_property PACKAGE_PIN P1 [get_ports {Q[2]}]
set_property PACKAGE_PIN N3 [get_ports {Q[1]}]
set_property PACKAGE_PIN P3 [get_ports {Q[0]}]
```

set_property IOSTANDARD LVCMOS33 [get_ports {Q[3]}]

set_property IOSTANDARD LVCMOS33 [get_ports {Q[2]}]

set_property IOSTANDARD LVCMOS33 [get_ports {Q[1]}]

set_property IOSTANDARD LVCMOS33 [get_ports {Q[0]}]

set_property PACKAGE_PIN W5 [get_ports clk]

set_property PACKAGE_PIN U16 [get_ports clk_div]

set_property PACKAGE_PIN R2 [get_ports dir]

set_property PACKAGE_PIN T1 [get_ports pr]

set_property PACKAGE_PIN U1 [get_ports rst]

set_property IOSTANDARD LVCMOS33 [get_ports clk]

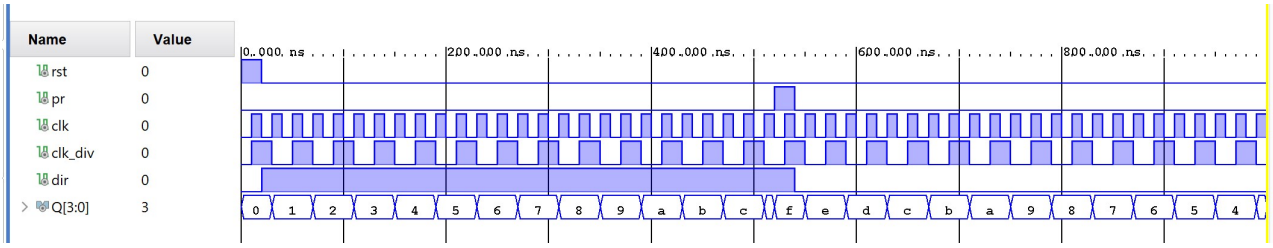
set_property IOSTANDARD LVCMOS33 [get_ports clk_div]

set_property IOSTANDARD LVCMOS33 [get_ports dir]

set_property IOSTANDARD LVCMOS33 [get_ports pr]

set_property IOSTANDARD LVCMOS33 [get_ports rst]

set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets pr_IBUF]



MOD 25 Counter

Name: dhruv subandh Roll No.:7

Code:

```
library IEEE;
 use IEEE.STD_LOGIC_1164.ALL;
 use IEEE.STD_LOGIC_UNSIGNED.ALL;
 entity mod25 is
   Port (
      rst : in STD_LOGIC;
      pr  : in STD_LOGIC;
      clk : in STD_LOGIC;
      clk_div : inout STD_LOGIC;
      dir : in STD_LOGIC;
      Q   : out STD_LOGIC_VECTOR(4 downto 0)
   );
 end mod25;
 architecture mod25_arch of mod25 is
   signal Qtemp : STD_LOGIC_VECTOR(4 downto 0) := "00000";
   signal Counter : STD_LOGIC_VECTOR (27 downto 0) := (others => '0');
 begin
   process(rst, pr, clk, dir)
   begin
      if rst = '1' then
         Qtemp <= (others => '0');

      elsif pr = '1' then
         Qtemp <= (others => '1');

      elsif falling_edge(clk) then
```

```
        if dir = '1' then  --UP counting
            if Qtemp < 24 then
                Qtemp <= Qtemp + 1;
            else
                Qtemp <= "00000";
            end if;
        else              --DOWN counting
            if Qtemp > 7 then
                Qtemp <= Qtemp -1;
            else
                Qtemp <= "11111";
            end if;
        end if;
    end if;
  end process;
   Q <= Qtemp;

  process(clk)
    begin
          if clk'event and clk = '1' then
              counter <= counter + 1;
          end if;
  end process;
  clk_div <= counter(0);
end mod25_arch;
```

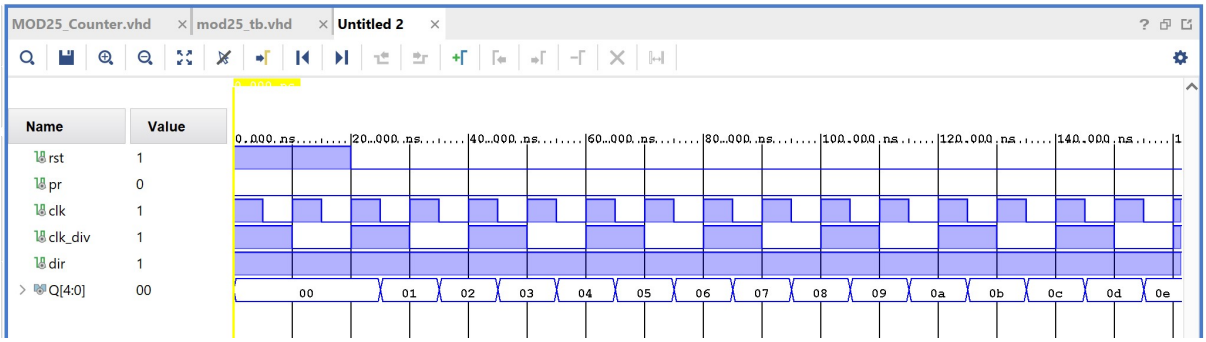
Test Bench:

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

```
entity mod25_tb is
end mod25_tb;
architecture Behavioral of mod25_tb is
  --Component Declaration for the Unit Under Test (UUT)
  component mod25
    Port (
      rst : in STD_LOGIC;
      pr  : in STD_LOGIC;
      clk : in STD_LOGIC;
      clk_div : inout STD_LOGIC;
      dir : in STD_LOGIC;
      Q   : out STD_LOGIC_VECTOR(4 downto 0)
    );
  end component;
  --Signals to connect to the UUT
  signal rst : STD_LOGIC := '0';
  signal pr  : STD_LOGIC := '0';
  signal clk : STD_LOGIC := '0';
  signal clk_div : STD_LOGIC := '0';
  signal dir : STD_LOGIC := '1'; --default: up count
  signal Q   : STD_LOGIC_VECTOR(4 downto 0);
begin
  --Instantiate the Unit Under Test (UUT)
  uut: mod25
    port map (
      rst => rst,
      pr  => pr,
      clk => clk,
      clk_div => clk_div,
      dir => dir,
```

```
        Q   => Q
    );
--Clock generation: 10 ns period (falling edge sensitive)
clk_process : process
begin
    while true loop
        clk <= '1';
        wait for 5 ns;
        clk <= '0';
        wait for 5 ns;
    end loop;
end process;
--Stimulus process
stim_proc: process
begin
    --Initial reset
    rst <= '1';
    wait for 20 ns;
    rst <= '0';
    wait for 20 ns;
    --Up count for a few cycles
    dir <= '1';  --count up
    wait for 300 ns;
    --Now preset
    pr <= '1';
    wait for 20 ns;
    pr <= '0';
    wait for 20 ns;
    --Down count
    dir <= '0';
```

```
        wait for 300 ns;
        --Apply reset again
        rst <= '1';
        wait for 20 ns;
        rst <= '0';
        --End simulation
        wait;
    end process;
end Behavioral;
```



Universal Shit Register

Name: dhruv subandh Roll No.:7

Code:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity USR_Code is
port (
        clk : in std_logic;
        clk_div : inout std_logic;
        rst : in std_logic;
        load : in std_logic;
        mode : in std_logic_vector (1 downto 0);
        si : in std_logic;
        pi : in std_logic_vector (3 downto 0);
        so : out std_logic;
        po : out std_logic (3 downto 0)
);
end USR_Code;

architecture beh of USR_Code is
        signal shift_reg : std_logic_vector(3 downto 0);
        signal counter : std_logic_vector (26 downto 0) := (others => '0');
begin
        process(clk)
                begin
                        if clk'event and clk = '1' then
                                if rst = '1' then
                                        counter <= (others => '0');
                                else
```

```
                counter <= counter + 1;
            end if;
        end if;
end process;
clk_div <= counter(0);

process(clk_div, rst, load)
    begin
        if rst = '1' then
            shift_reg <= (others => '0');
        elsif
            clk_div'event and clk_div = '1' then
                case mode is
                    when "00" =>
                        shift_reg(3 downto 1) <= shift_reg(2 downto 0);
                        shift_reg(0) <= si;
                        so <= shift_reg(3);
                    when "01" =>
                        shift_reg(3 downto 1) <= shift_reg(2 downto 0);
                        shift_reg(0) <= si;
                        po <= shift_reg;
                    when "10" =>
                        if load ='0' then
                            shift_reg <= pi;
                        else
                        shift_reg(3 downto 1) <= shift_reg(2 downto 0);
                        end if;
                        so <= shift_reg(3);
                    when "11" =>
                        po <= pi;
                    when others => null;
                end case;
        end if;
end process;
```

```
end beh;
```

TestBench:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity USR_TB is
end USR_TB;

architecture beh of USR_TB is
	component USR_Code is
	port (
		clk : in std_logic;
		clk_div : inout std_logic;
		rst : in std_logic;
		load : in std_logic;
		mode : in std_logic_vector (1 downto 0);
		si : in std_logic;
		pi : in std_logic_vector (3 downto 0);
		so : out std_logic;
		po : out std_logic_vector (3 downto 0));
	end com ponent;

	signal clk, clk_div, rst, load, si, so : std_logic := '0';
	signal mode : std_logic_vector(1 downto 0);
	signal pi, po : std_logic_vector (3 downto 0);

begin
	uut: USR_Code PORT MAP (clk, clk_div, rst, load, mode, si, pi, so, po);

	clk_process : process
```

```
        begin
                clk <= '0';
                wait for 10 ns;
                clk <= '1';
                wait for 10 ns;
end process;

stim_process : process
        begin
                rst <= '1';
                wait for 20 ns;
                rst <= '0';

        mode <= "00";
                si <= '1';
                wait for 200 ns;
                si <= '0';
                wait for 200 ns;

        mode <= "01";
                si <= '1';
                wait for 100 ns;
                si <= '0';
                wait for 200 ns;

        mode <= "10";
                load <= '0';
                pi <= "1110";
                wait for 100 ns;
                load <= '1';
                wait for 300 ns;

        mode <= "11";
                pi <= "1010";
```

```
                wait for 50 ns;
                pi <= “1010”;
                wait for 50 ns;
        wait;
        end process;
end beh;
```