Name: Deep Patel **Roll No.: CS21B1008**

Introduction to HDL Assignment

Q.3 Design a system which receives 16-bit data sequentially and output even and odd sequenced data from the fourth data point onwards. Verify the design functionally by writing a test-bench at least for two sets of 16-bit data. You need to simulate the entire design using the test bench.

Design Code:

File No. 3: prob3.vhd

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

entity prob3 is
port (d: in std_logic_vector(3 downto 0);
clk,nreset,nready: in std_logic;
q_odd: out std_logic_vector(3 downto 0);
q_even:out std_logic_vector(3 downto 0));
end prob3;

architecture probarch of prob3 is

component d_latch_3
port (d: in std_logic_vector (3 downto 0);
enable: in std_logic_vector (1 downto 0);
q_odd: out std_logic_vector(3 downto 0);
clk,nreset : in std_logic;
q_even: out std_logic_vector(3 downto 0));
end component;

component counter16 is
port(nreset : in std_logic;
nready : in std_logic;
clk : in std_logic;
count : out std_logic_vector (3 downto 0));
end component;

component en_gen3 is
port(clk : in std_logic;
```

```
nreset: in std_logic;
count_in: in std_logic_vector(3 downto 0);
enable : out std_logic_vector (1 downto 0));
end component;

signal int_clk : std_logic;
signal counter : std_logic_vector(3 downto 0);
signal enable_out : std_logic_vector (1 downto 0);
begin
DFF_3 : d_latch_3 port map(clk => clk,nreset => nreset,enable => enable_out,d => d,q_odd =>
q_odd,q_even => q_even);
unit_enable: en_gen3 port map(clk => clk,nreset => nreset,count_in => counter,enable =>
enable_out);
unit_counter: counter16 port map(clk => clk,nreset => nreset,nready => nready,count =>
counter);
end probarch;
```

Sub Files:

1. **d_latch.vhd**

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

entity d_latch_3 is
port(
clk: in std_logic;
nreset: in std_logic;
enable: in std_logic_vector (1 downto 0);
d : in std_logic_vector (3 downto 0);
q_odd: out std_logic_vector(3 downto 0);
q_even: out std_logic_vector(3 downto 0)
);
end entity d_latch_3;

architecture d_arch3 of d_latch_3 is
begin
process (nreset, clk) is
begin
  if (nreset = '0') then
    q_odd <= (others=>'0');
    q_even <= (others=>'0');
```

```
  elsif (rising_edge(clk)) then
    if (enable = "01") then
      q_odd <= d;
    elsif (enable = "10") then
      q_even <= d;
    end if;
  end if;
end process;
end d_arch3;
```

2. engen3.vhd

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

entity en_gen3 is
port( clk : in std_logic;
nreset: in std_logic;
count_in: in std_logic_vector(3 downto 0);
enable : out std_logic_vector(1 downto 0)
);
end entity en_gen3;
architecture en_gen_arch3 of en_gen3 is
begin
process (nreset, clk) is
begin
  if (nreset = '0') then
    enable <= "00";
  elsif (rising_edge(clk)) then
     if (count_in > "010") and (count_in(0) = '0') then
       enable <= "01";
     elsif (count_in > "010") and (count_in(0) = '1') then
       enable <= "10";
     end if;
  end if;
end process;
end en_gen_arch3;
```

3. **count_8.vhd**

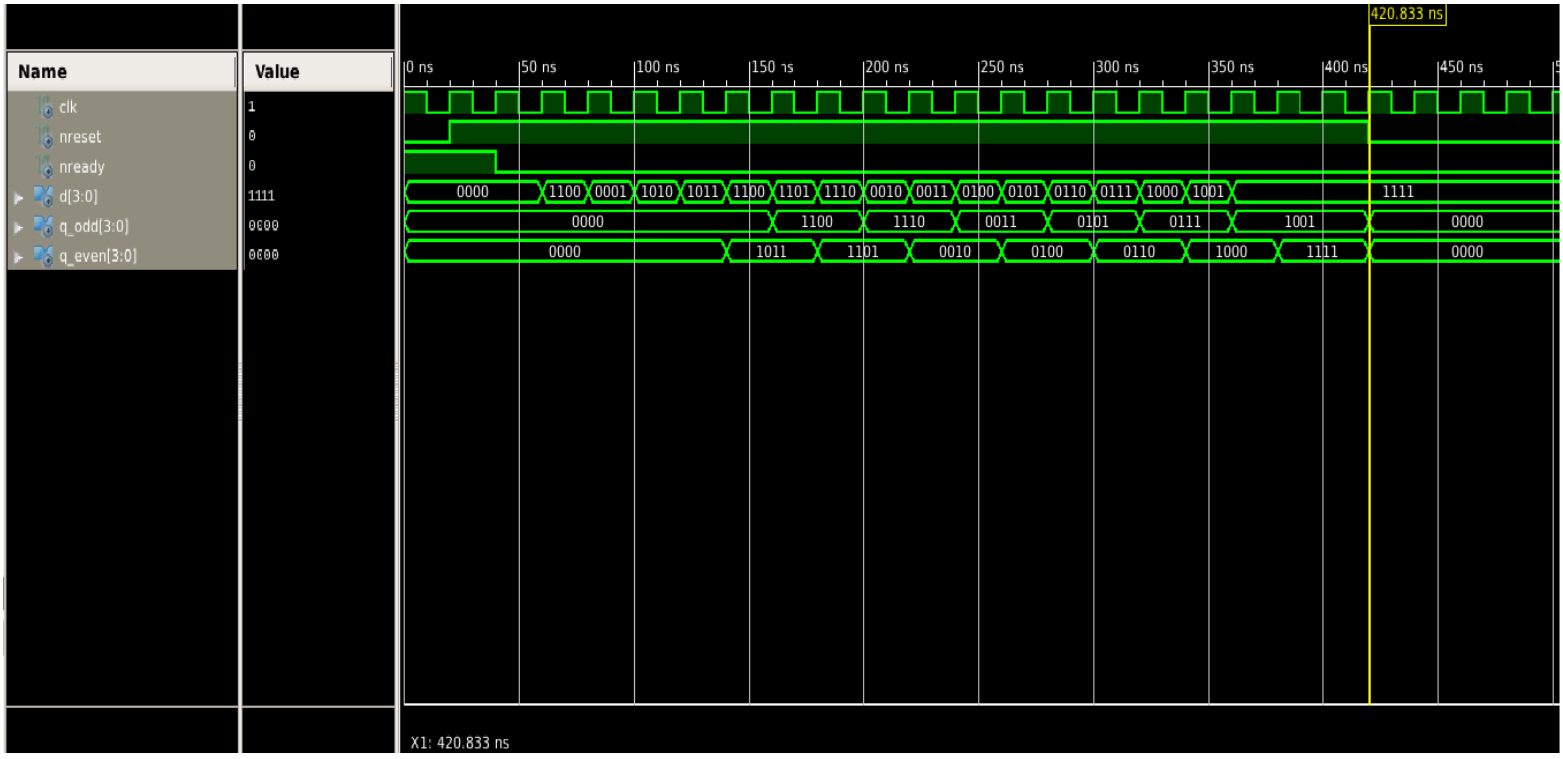
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_unsigned.all;
entity counter16 is

port(
clk : in std_logic;
nreset : in std_logic;
nready : in std_logic;
count : out std_logic_vector (3 downto 0)
);
end counter16;
architecture counter16_arch of counter16 is
signal temp : unsigned (3 downto 0);
constant MAXCOUNT : unsigned (3 downto 0) := "1111";
begin
p0: process (nreset, clk) is

begin
if (nreset = '0') then
temp <= (others => '0');
elsif (rising_edge(clk)) then
  if (nready = '1') then
  temp <= (others => '0');
  elsif (temp = MAXCOUNT) then
  temp <= (others => '0');
  else
  temp <= temp + 1;
  end if;
end if;
end process p0;
count <= std_logic_vector(temp);
end counter16_arch;
```

Test Bench - 1:

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

entity tprob3 is end tprob3;

architecture tb1p3 of tprob3 is
component prob3
port(d: in std_logic_vector(3 downto 0);
clk,nreset,nready: in std_logic;
q_odd:out std_logic_vector(3 downto 0);
q_even:out std_logic_vector(3 downto 0));
end component;

signal clk : std_logic := '1';
signal nreset : std_logic := '1';
signal nready : std_logic;
signal d: std_logic_vector(3 downto 0) := (others=>'0');
signal q_odd: std_logic_vector(3 downto 0);
signal q_even: std_logic_vector(3 downto 0);

begin
dut:prob3 port map(d=>d, nready=>nready, nreset=>nreset, clk=>clk, q_odd=>q_odd,
q_even=>q_even);
clock_gen: process(clk)
begin
clk <= not clk after 10 ns;
end process clock_gen;
wavegen_proc: process
 begin
nreset <= '0';
nready <= '1';
wait for 20 ns;
nreset <= '1';
wait for 20 ns;
nready <= '0';
 wait for 20 ns; d <= "1100";
 wait for 20 ns; d <= "0001";
  wait for 20 ns; d <= "1010";
 wait for 20 ns; d <= "1011";
 wait for 20 ns; d <= "1100";
 wait for 20 ns; d <= "1101";
```

```
wait for 20 ns; d <= "1110";
wait for 20 ns; d <= "0010";
wait for 20 ns; d <= "0011";
wait for 20 ns; d <= "0100";
wait for 20 ns; d <= "0101";
wait for 20 ns; d <= "0110";
wait for 20 ns; d <= "0111";
wait for 20 ns; d <= "1000";
wait for 20 ns; d <= "1001";
wait for 20 ns; d <= "1111";
wait for 20 ns;
wait for 40 ns;
nreset <= '0';
wait;
end process;
end architecture tb1p3;
```

Simulation of Test Bench - 1:



Test Bench - 2:

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

entity t2prob3 is end t2prob3;

architecture tb2p3 of t2prob3 is
component prob3
port(d: in std_logic_vector(3 downto 0);
clk,nreset,nready: in std_logic;
q_odd:out std_logic_vector(3 downto 0);
q_even:out std_logic_vector(3 downto 0));
end component;

signal clk : std_logic := '1';
signal nreset : std_logic := '1';
signal nready : std_logic;
signal d: std_logic_vector(3 downto 0) := (others=>'0');
signal q_odd: std_logic_vector(3 downto 0);
signal q_even: std_logic_vector(3 downto 0);

begin
dut:prob3 port map(d=>d, nready=>nready, nreset=>nreset, clk=>clk, q_odd=>q_odd,
q_even=>q_even);
clock_gen: process(clk)
begin
clk <= not clk after 10 ns;
end process clock_gen;
wavegen_proc: process
 begin
nreset <= '0';
nready <= '1';
wait for 20 ns;
nreset <= '1';
wait for 20 ns;
nready <= '0';
wait for 20 ns; d <= "1000";
 wait for 20 ns; d <= "1001";
 wait for 20 ns; d <= "1111";
 wait for 20 ns; d <= "1100";
 wait for 20 ns; d <= "0001";
  wait for 20 ns; d <= "1010";
 wait for 20 ns; d <= "1011";
```

```
wait for 20 ns; d <= "1100";
wait for 20 ns; d <= "1101";
wait for 20 ns; d <= "1110";
wait for 20 ns; d <= "0001";
 wait for 20 ns; d <= "1010";
wait for 20 ns; d <= "1011";
wait for 20 ns; d <= "0010";
wait for 20 ns; d <= "0011";
wait for 20 ns; d <= "1000";
wait for 20 ns; d <= "1001";
wait for 20 ns; d <= "1111";
wait for 20 ns;
wait for 40 ns;
nreset <= '0';
wait;
end process;
end architecture tb2p3;
```

Simulation of Test Bench - 2: