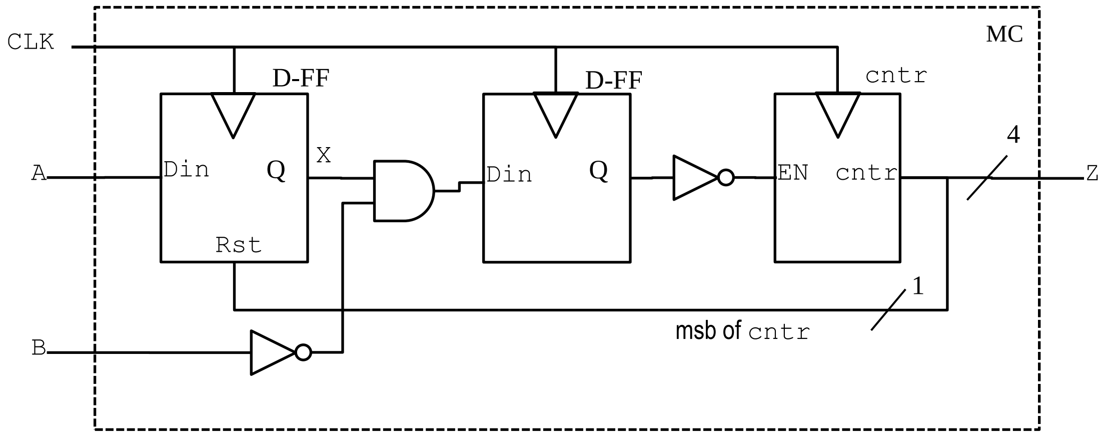
No : Answers **Name :** Solutions
Eskişehir Osmangazi University Faculty of Engineering and Architecture
Department of Electrical Engineering & Electronics 07.01.2025
"*Introduction To VHDL-FPGA*" Final
Note : Books, notes, computers are allowed, communication of all kind is prohibited. 90 minutes.

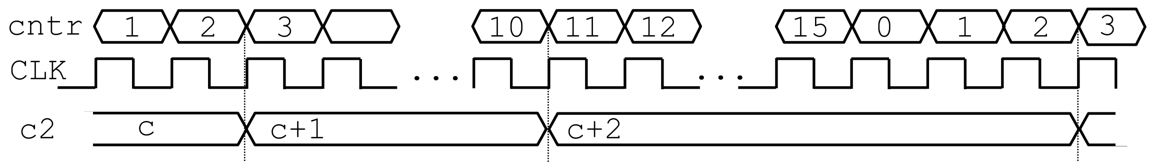
1. Design the following circuit in VHDL within one process. `Rst` of D-FF is synchronous.



```
entity MC is port (
  clk, A, B : in STD_LOGIC;
  Z : out STD_LOGIC_VECTOR(3 downto 0));
end MC;
architecture MC of MC is
  signal cntr : STD_LOGIC_VECTOR(3 downto 0);
  signal X, Y : STD_LOGIC;
begin
  process(CLK) is begin
    if(rising_edge(CLK)) then
      if(cntr(3)='1') then
        X <= '0';
      else
        X <= A;
      end if;
      Y <= not(X and not(B)); -- see note below
      if(Y='1') then
        cntr <= cntr +1;
      end if;
    end if;
    Z < cntr;
  end process;
end MC;

-- same thing can be achieved with
--     Y <= (X and not(B));
--     if(Y='0') then
```

2. A counter (`cntr`) counts from 0 to 15, incrementing 1 on every rising edge of `CLK` and going back to zero afterwards, continuously. Another 4-bits counter (`c2`), when enabled through EN input, increments when the first counter is "0010" or "1010". Complete this fully synchronous circuit in VHDL.

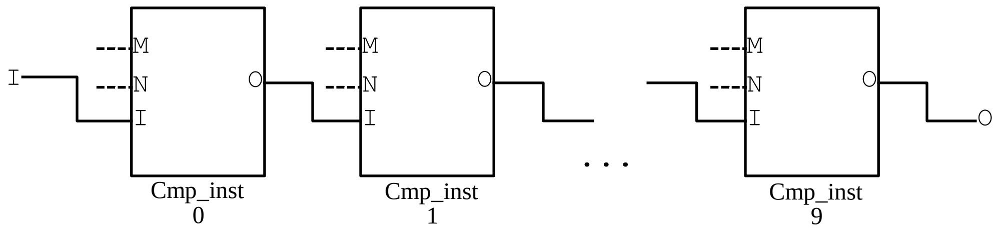


```
entity CCC is port (
  clk  : in  STD_LOGIC;
  EN   : in  STD_LOGIC;
  cnt2 : out STD_LOGIC_VECTOR(3 downto 0));
end CCC;
architecture CCC of CCC is
  signal cntr,c2 : STD_LOGIC_VECTOR(3 downto 0);
begin
  process(CLK) is begin
    if(rising_edge(CLK)) then
      cntr <= cntr +1; -- rollsover to 0000 at 1111
      if((EN='1')and((cntr="0010")or(cntr="1010"))) then
        c2 <= c2 +1;
      end if;
    end if;
    cnt2 <= c2;
  end process;
end CCC;
```

3. In package MyPack.vhd used in your code, an array of records is defined as;

```
type MT is record (M: integer; N: integer);
type MAT is array (integer range <>) of MT;
```

In the following code, an array of `MAT` type is created along with the component declaration of `Cmp`. Complete the code that creates a `Cmp` chain as shown, using `for-generate`. Each instantiation of `Cmp` uses corresponding values in the `MA` array.



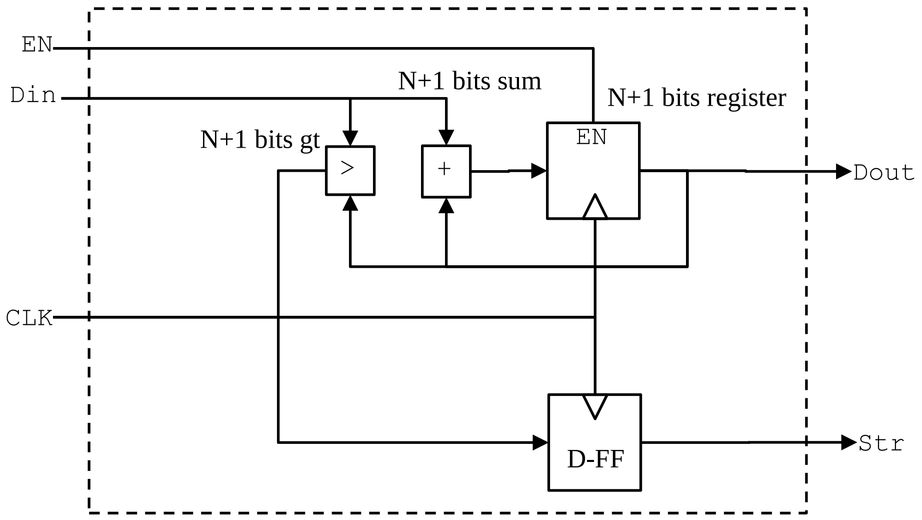
```
...
use work.MyPack.All;
entity Test is port (
  CLK : in  STD_LOGIC;
    I : in  STD_LOGIC;
    O : out STD_LOGIC);
end Test;
architecture Test of Test is
  constant MA: MAT(0 to 9):=((5,2),(5,3),(7,6),others=>(1,0));
  component Cmp generic (M: integer; N: integer);
    port ( CLK, I: in STD_LOGIC; O: out STD_LOGIC);
  end component;
  signal Ix : STD_LOGIC_VECTOR(0 to MA'HIGH+1);
begin
  Ix(0) <= I; O <= Ix(MA'HIGH+1)
  L1:for i in 0 to MA'HIGH generate
    Cmp_inst: Cmp
      generic map (
        M => MA(i).M,
        N => MA(i).N
      )
      port map (
        CLK => CLK,
        I   => Ix(i),
        O   => O(i+1)
      );
  end generate;

end Test;
```

4. Draw the circuit inferred by the following VHDL code, using primitive elements like flip-flops, counters, registers, gates, comparators and basic math-circuits like adders and multipliers.

```
entity Cir is generic ( N: integer:=4); port (
    CLK : in  STD_LOGIC;
    EN  : in  STD_LOGIC;
    Din : in  integer range 0 to 2**N -1;
    Dout: out integer range 0 to 2**(N+1) -1;
    Str : out STD_LOGIC);
end Cir;
architecture Cir of Cir is
  signal Dx : integer range 0 to 2**(N+1) -1;
begin
  process(CLK) is
    variable vStr : STD_LOGIC;
  begin
    if(rising_edge(CLK)) then
      vStr := '0';
      if(EN='1') then
        Dx <= Din + Dx;
      end if;
      if(Din>Dx) then
        vStr := '1';
      end if;
      Str <= vStr;
    end if;
    Dout <= Dx;
  end process;
end Cir;
```

```
--comparator and adder are combinatorial elements
```


EN
Din
N+1 bits sum
N+1 bits register
N+1 bits gt
>
+
EN
Dout
CLK
D-FF
Str