No : Answers **Name : Solutions**

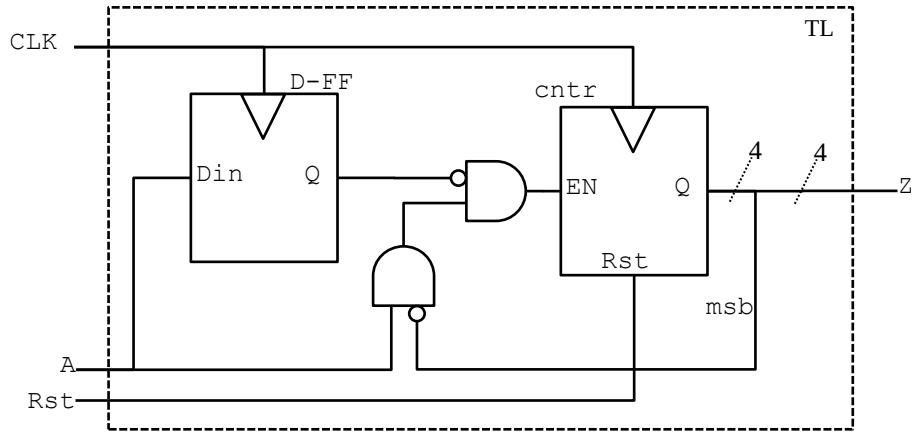
Eskişehir Osmangazi University Faculty of Engineering and Architecture

Department of Electrical Engineering & Electronics 13.11.2025

"*Introduction To VHDL-FPGA*" Midterm

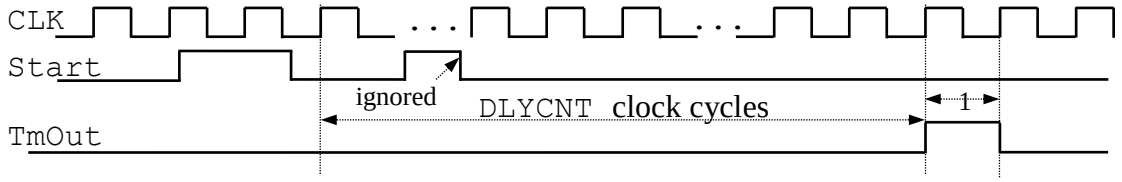
Note : Books, notes, computers are allowed, communication of all kind is prohibited. 90 minutes.

1. Design the following circuit in VHDL within one process. `Rst` of `D-FF` is synchronous, `cntr` is 4 bits up-counter.




```
entity TL is port (
  clk, Rst, A : in STD_LOGIC;
             Z : out STD_LOGIC_VECTOR(0 to 3));
end TL;
architecture TL of TL is
  signal pA : STD_LOGIC;
  signal cntr : STD_LOGIC_VECTOR(0 to 3);
begin
  process(CLK) is begin
    if(rising_edge(CLK)) then
      pA <= A;
      if(Rst='1') then
        cntr <= "0000";
      elsif((cntr(0)='0')and(pA='0')and(A='1')) then
        cntr <= cntr+1;
      end if;
    end if;
    Z < cntr;
  end process;
end TL;
```

2. A timer-entity accepts a `Start` input and outputs a delayed `TmOut` output. Timer detects the falling-edge of the `Start` input and starts counting at the rising-edge of the clock `CLK`. When the count reaches `generic` integer input `DLYCNT`, the output `TmOut` is set to `'1'` for one clock cycle. `Start` input is ignored when timer is actively counting. Complete timer circuit in VHDL.




```
entity Timer is
  generic ( DLYCNT : integer := 1000 );
  port (
    clk   : in  STD_LOGIC;
    Start : in  STD_LOGIC;
    TmOut : out STD_LOGIC);
end Timer;
architecture Timer of Timer is
  signal cntr : integer range 0 to DLYCNT := 0;
  signal pS : STD_LOGIC;
  signal counting : STD_LOGIC := '0';
begin
  process(CLK) is begin
    if(rising_edge(CLK)) then
      pS <= Start;
      if(counting='1') then
        if(cntr=DLYCNT-1) then
          counting <= '0';
          TmOut <= '1';
        else
          cntr <= cntr +1;
        end if;
      else
        TmOut <= '0';
        cntr <= 0;
        if((pS='1')and(Start='0'))
          counting <= '1';
        end if;
      end if;
    end if;
  end process;
end Timer;
```

3. Two 4-bits counters up-count in a ping-pong fashion; `cntrA` counts up, while `cntrB` waits at 0000. When `cntrA` reaches 1111, it is set to 0000 and `cntrB` starts counting while `cntrA` waits at 0000. When `cntrB` reaches 1111, it is set to 0000 and `cntrA` starts counting again while `cntrB` waits at 0000. This behaviour continues indefinitely. Design the circuit using VHDL.

```
entity TwoCounters is port (
  clk  : in  STD_LOGIC;
  A, B : out STD_LOGIC_VECTOR(0 to 3));
end TwoCounters;
architecture TwoCounters of TwoCounters is
  signal cntrA, cntrB: STD_LOGIC_VECTOR(0 to 3);
  signal AB : STD_LOGIC := '0'; -- which cntr?
begin
  A <= cntrA; B <= cntrB;
  process(CLK) is begin
    if(rising_edge(CLK)) then
      if(AB='0') then
        if(cntrA="1111") then
          cntrA <= "0000";
          AB <= '1';
        else
          cntrA <= cntrA +1;
        end if;
      else
        if(cntrB="1111") then
          cntrB <= "0000";
          AB <= '0';
        else
          cntrB <= cntrB +1;
        end if;
      end if;
    end if;
  end process;
end TwoCounters;
```