No : Answers **Name : Solutions**

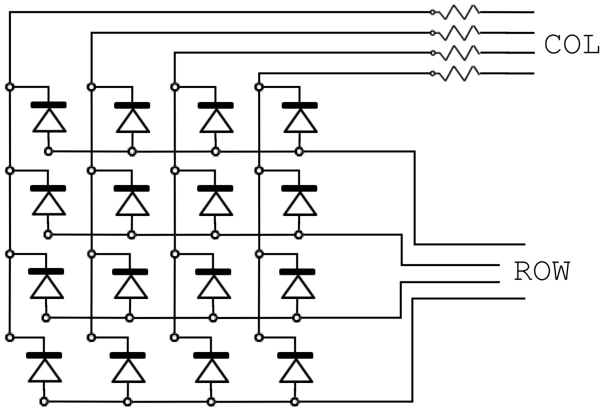
Eskişehir Osmangazi University Faculty of Engineering and Architecture

Department of Computer Engineering 02.01.2019

"*Introduction To VHDL-FPGA*" Final

Note : Books, notes, computers are allowed, communication of all kind is prohibited. 75 minutes.

1. A LED matrix is given as shown. You are asked to drive these LEDS to display row patterns given in a 4x4 array where value at each index is a row pattern. Since the row and column lines are shared/common, you need to light-up a row of LEDs for 10-50 ms and move on to next row for doing the same thing. This repeated scan of rows will trick human eye/brain into seeing the pattern.

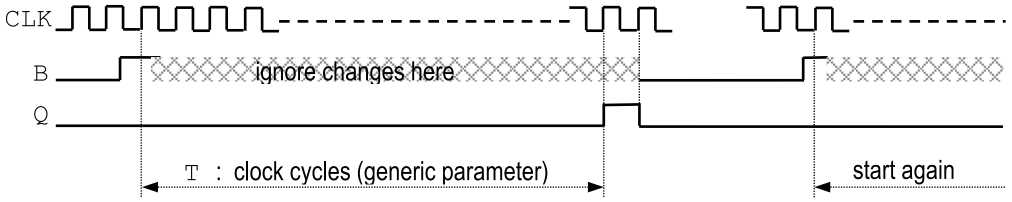


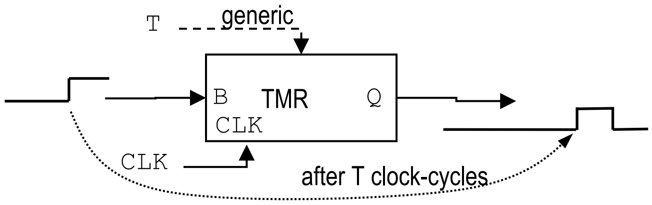
```
entity SCAN is Port (
   CLK  : in  STD_LOGIC; -- 2 MHz clock signal
   ROW  : out STD_LOGIC_VECTOR(3 downto 0);   -- one hot row selection
   COL  : out STD_LOGIC_VECTOR(3 downto 0)); -- inverted col pattern
end SCAN;

architecture SCAN of SCAN is
  type A4x4 is array (0 to 3) of STD_LOGIC_VECTOR(3 downto 0);
  constant ARR : A4x4 := ("0110","1001","1001","0110"); --displayed as O
  signal cntr : STD_LOGIC_VECTOR(17 downto 0);
begin
  process(CLK, cntr) is begin
    if(RISING_EDGE(CLK)) then cntr <= cntr + 1; end if;
  end process;
  with cntr(17 downto 16) select
    ROW <= "0001" when "00",
           "0010" when "01",
           "0100" when "10",
           "1000" when others;
  with cntr(17 downto 16) select
    COL <= not ARR(0) when "00",
           not ARR(1) when "01",
           not ARR(2) when "10",
           not ARR(3) when others;

end SCAN;
```

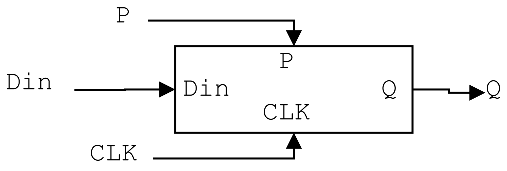
2. Design the generic timer component described in the timing diagram. Basically, it outputs a 1 clock cycle length pulse T clock cycles after the input goes from 0 to 1. It ignores the changes on input before the output pulse ends. T is a nonzero positive integer.





```
entity TMR is Generic ( T : in integer );
  Port (
    CLK, B : in  STD_LOGIC;
    Q      : out STD_LOGIC);
end TMR;
architecture TMR of TMR is
  signal cntr : integer := 0;
  signal Bp : STD_LOGIC;
begin
  process(CLK,B,Bp,cntr) is begin
    if(RISING_EDGE(CLK)) then
      if(cntr=0) then
        if(Bp&B="01") then cntr <= 1; end if;
        Q <= '0';
      else
        if(cntr=T) then
          cntr <= 0;
          Q <= '1';
        else
          cntr <= cnt+1;
        end if;
      end if;
      Bp <= B;
    end if;
  end process;
end TMR;
```

3. You are to detect a 4-bit pattern P in a signal serially received from an input Din. When the pattern is detected the output Q is set to '1' where otherwise it is '0'. Design the entity in VHDL.



```
entity DETECTOR is Port (
  CLK, Din : in  STD_LOGIC;
  P : in  STD_LOGIC_VECTOR(3 downto 0);
  Q : out STD_LOGIC);
end DETECTOR;
architecture DETECTOR of DETECTOR is
  signal D: STD_LOGIC_VECTOR(3 downto 0);
begin
  process(CLK,Din,D,P) is begin
    if(RISING_EDGE(CLK)) then
      D(0) <= D(1);
      D(1) <= D(2);
      D(2) <= D(3);
      D(3) <= Din;
      if(D=P) then
        Q <= '1';
      else
        Q <= '0';
      end if;
    end if;
  end process;
end DETECTOR;
```