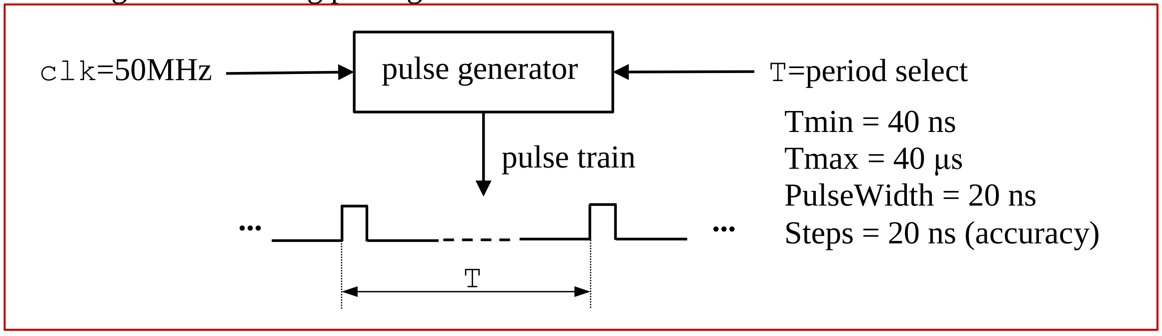
No : Answers **Name :** Solutions
Eskişehir Osmangazi University Faculty of Engineering and Architecture
Department of Electrical Engineering & Electronics 25.11.2021
"*Introduction To VHDL-FPGA*" Midterm
Note : Books, notes, computers are allowed, communication of all kind is prohibited. 90 minutes.

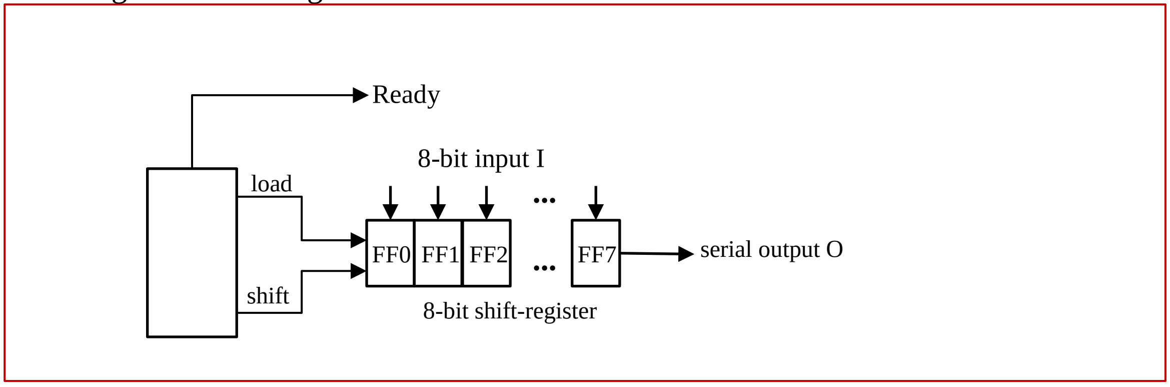
1. Design the following pulse generator in VHDL.



```
entity PGen is port (
  clk : in  STD_LOGIC;
  T   : in  integer;
  P   : out STD_LOGIC);
end PGen;
architecture PGen of PGen is
  constant Tmax : integer := 1999;
  signal Px : STD_LOGIC := '0';
  signal cntr : integer range 0 to Tmax;

begin
  P <= Px;
  PG:process(clk,T,cntr,Px) is begin
    if(rising_edge(clk)) then
      if((cntr>=T)or(cntr>=Tmax)) then
        cntr <= 0;
        Px <= not(Px);
      else
        cntr <= cntr+1;
        Px <= '0';
      end if;
    end if;
  end process;
end PGen;
```

2. Design the following circuit in VHDL.

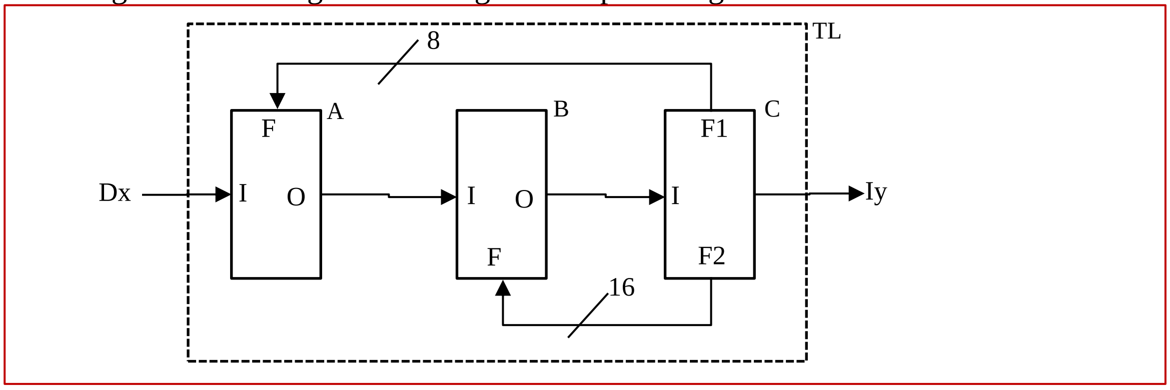


The circuit serially shifts-out data in the FFs. It sends a Ready (active High) signal to the output just when the data in the FF0 (the last bit) is being shifted out. At the end of the appearance of this bit at the output, the circuit loads new data from input I and the first bit of the new data (FF7) will be seen at the output. This is just a parallel-in serial-out shift register with a Ready signal getting active during the last bit.

```vhdl
entity PiSo is port (
  clk   : in  STD_LOGIC;
  I     : in  STD_LOGIC_VECTOR(0 to 7);
  O     : out STD_LOGIC;
  Ready : out STD_LOGIC);
end PiSo;
architecture PiSo of PiSo is
  signal SR : STD_LOGIC_VECTOR(0 to 7);
  signal cntr : integer range 0 to 7 :=0;

begin
  O <= SR(7);
  process(clk,I,cntr,SR) is begin
    if(rising_edge(clk)) then
      if(cntr=7) then
        cntr <= 0;
        SR <= I;
      else
        cntr <= cntr+1;
        SR <= '0' & SR(0 to 6);
      end if;
      if(cntr=6) then
        Ready <= '1';
      else
        Ready <= '0';
      end if;
    end if;
  end process;
end PiSo;
```

3. Design the following circuit using the components given.



```
entity TL is port (
  clk    : in  STD_LOGIC;
  Dx     : in  STD_LOGIC_VECTOR(7 downto 0);
  Iy     : out STD_LOGIC_VECTOR(7 downto 0));
end TL;
architecture TL of TL is
  component A is port(
    clk : in  STD_LOGIC;
    I   : in  STD_LOGIC_VECTOR(7 downto 0);
    F   : in  STD_LOGIC_VECTOR(7 downto 0);
    O   : out STD_LOGIC);
  end component;
  component B is port(
    I : in  STD_LOGIC;
    F : in  STD_LOGIC_VECTOR(15 downto 0);
    O : out STD_LOGIC_VECTOR(3 downto 0));
  end component;
  component C is port(
    clk : in  STD_LOGIC;
    I   : in  STD_LOGIC_VECTOR(3 downto 0);
    F1  : out STD_LOGIC_VECTOR(7 downto 0);
    F2  : out STD_LOGIC_VECTOR(15 downto 0);
    O   : out STD_LOGIC_VECTOR(7 downto 0));
  end component;

  signal F1: STD_LOGIC_VECTOR(7 downto 0);
  signal OA: STD_LOGIC;
  signal OB: STD_LOGIC_VECTOR(3 downto 0);
  signal F2: STD_LOGIC_VECTOR(15 downto 0);

begin
  CA:A port map( clk => clk, I => Dx, F => F1, O => OA);
  CB:B port map( I => OA, F => F2, O => OB);
  CC:C port map(
    clk => clk, I => OB, F1 => F1, F2 => F2, O => Iy);
end TL;
```