No : **Name :**

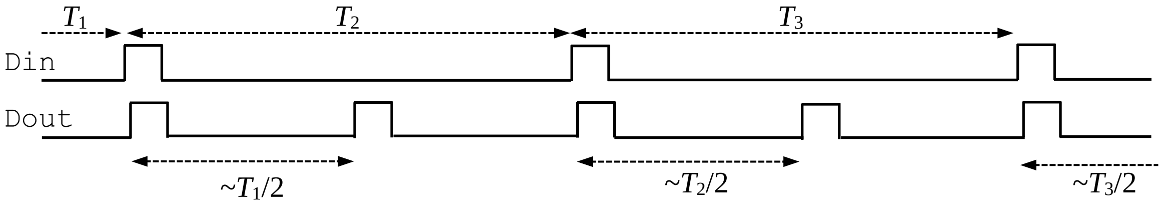
Eskişehir Osmangazi University Faculty of Engineering and Architecture

Department of Electrical Engineering & Electronics 17.01.2024

"*Introduction To VHDL-FPGA*" Final Exam

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

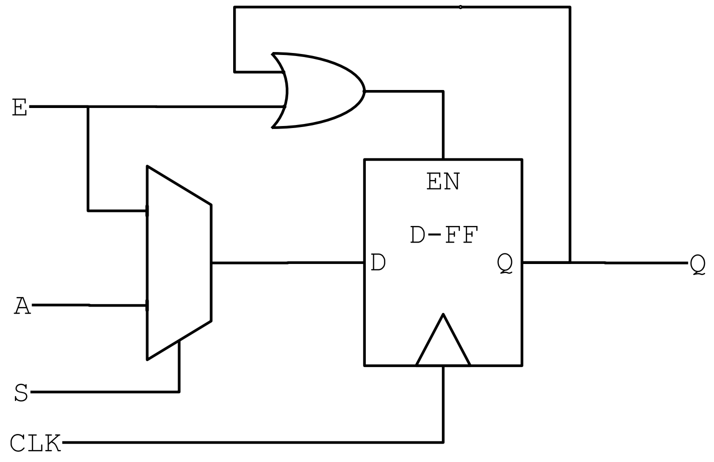
1. A circuit measures the period of incoming Din signal in number of clock cycles and inserts additional pulses with approximately half the measured period, continuously. The anticipated timing diagram is shown below.



Assume that clock frequency is much higher than the frequency of `Din` and the period of `Din` changes slowly. Design this circuit in VHDL.

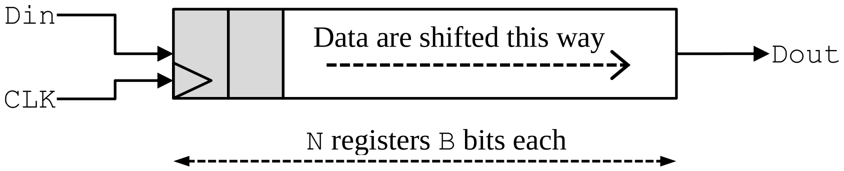
```
entity Ptm is port (
  CLK  : in  STD_LOGIC;
  Din  : in  STD_LOGIC;
  Dout : out STD_LOGIC);
end Ptm;

architecture Ptm of Ptm is
  signal Dinp, Doutx : STD_LOGIC;
  signal cntr, Hlf: integer;
begin
  process(CLK) is begin
    if(rising_edge(CLK)) then
      Dinp <= Din;
      -- on input 0->1 transition restart counting and
      -- store half of previous
      if((Dinp='0')and(Din='1')) then
        Hlf <= cntr/2; cntr <= 0;
      else
        cntr <= cntr +1;
      end if;
      if(cntr=Hlf) then -- generate 1 at half
        Doutx <= '1';
      else
        Doutx <= '0';
      end if;
      Dout <= Doutx or Din; -- combine input and generated
    end if;
  end process;
end;
```

2. Write a VHDL module that corresponds to the following circuit.

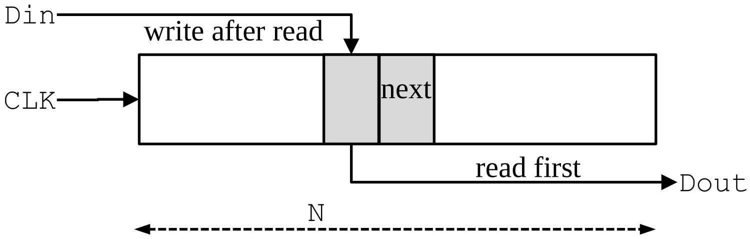


```
entity M is port (
  CLK, A E, S : in STD_LOGIC;
  Q : out STD_LOGIC);
end M;

architecture M of M is
  signal Qx : STD_LOGIC;
begin
  process(CLK) is begin
    if(rising_edge(CLK)) then
      if((E or Qx)='1') then
        if(S='0') then
          Qx <= E;
        else
          Qx <= A;
        end if;
      end if;
    end if;
  end process;
  Q <= Qx;
end M;

-- combinatorial circuits could be realized outside the process
```

3. A FIFO buffer is a sample delay device that the values appear at the output after N clock cycles they were read from input.

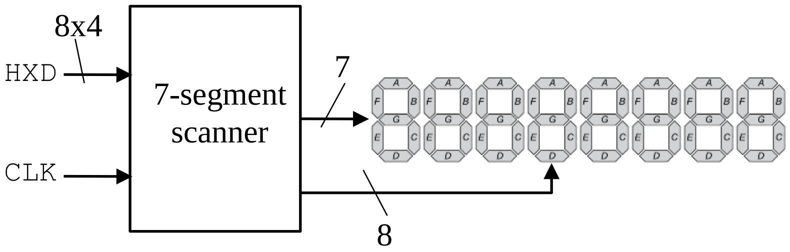


Since the circuit consumes high number of slices for large NxB, designers opt to use block memories as FIFO by configuring a circular buffer. After each read, a write is done at the same address (READ_FIRST mode). The address is incremented/decremented by 1 so that a value is read N clocks later it was written. Design the circuit in VHDL.



```
entity Fifo is generic(N:integer:=500); port (
  CLK  : in  STD_LOGIC;
  Din  : in  integer;
  Dout : out integer);
end Fifo;

architecture Fifo of Fifo is
  type Ft is array(0 to N-1) of integer;
  shared variable FB: Ft;
  signal ADR : integer :=0;
begin
  process(CLK) is begin
    if(rising_edge(CLK)) then
      Dout <= FB(ADR); -- these are done
      FB(ADR) := Din;  -- simultaneously
      if(ADR=N-1) then
        ADR <= 0;
      else
        ADR <= ADR +1;
      end if;
    end if;
  end process;
end Fifo;
```

4. A seven-segment LED scanner circuit receives an array of 4-bit numbers, decodes them to 7-segment visuals using a given decoder and sends them one by one to LED anodes activating related LED cathode for each numeric location.



Complete the code that does the job using the given 4-to-7 decoder.

```
type HXAT is array (0 to7) of STD_LOGIC_VECTOR(3 downto 0);
-- HXAT global type is defined in a library
entity SSeg is port (
  CLK : in STD_LOGIC; -- 100 Hz
  HXD : in HXAT;
  SEG : out STD_LOGIC_VECTOR(6 downto 0); -- seg. data to anodes
  DIG : out STD_LOGIC_VECTOR(7 downto 0); -- to cathodes
); end SSeg;
architecture SSeg of SSeg is
  component Dec4to7 is port(
    HX : in  STD_LOGIC_VECTOR(3 downto 0); -- ex: HX=0001
    SD : out STD_LOGIC_VECTOR(6 downto 0)  -- SD=abcdefg=0110000
  ); end component;
  signal HX : STD_LOGIC_VECTOR(3 downto 0);
  signal dign: integer range 0 to 7;
  signal DIGx: STD_LOGIC_VECTOR(7 downto 0);
begin
  HX <= HXD(dign); -- get the hex number from array
  CNV: Dec4to7 port map (HX=>HX,SD=>SEG); -- convert it to 7-Seg
  DIG <= DIGx; -- select digit

  -- move to next digit
  SCAN:process(CLK) is begin
    if(rising_edge(CLK)) then
      if(dign=7) then
        dign <= 0; DIGx <= x"01";
      else
        dign <= dign+1;
        DIGx <= DIGx(6 downto 0) & '0';
      end if;
    end if;
  end process;
end SSeg;
-- 3-to-8 decoder could've been used instead of shifting
```