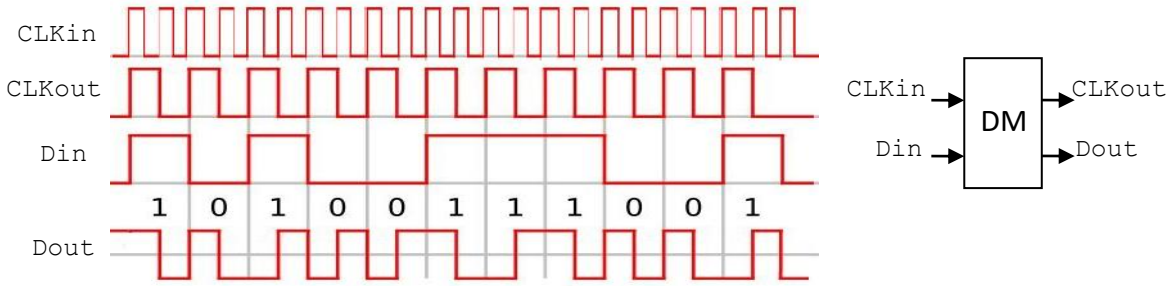
**No : Answers** **Name : Solutions**
**Eskişehir Osmangazi University Faculty of Engineering and Architecture**
**Department of Computer Engineering** 02.01.2018
"*Introduction To VHDL-FPGA*" Final Exam.
**Note : Books, notes, computers are allowed, communication of all kind is prohibited. 90 (ninety) minutes.**

1. Differential Manchester encoding is simply defined as
   - No transition at the beginning of bit period indicates a binary 1.
   - Transition at the beginning of bit period indicates a binary 0.
   - There is a transition in the middle of every bit period.

Timing diagram shows an example Differential Manchester signal generated using a double-rated input clock.




Design a Differential Manchester Encoder circuit using VHDL.

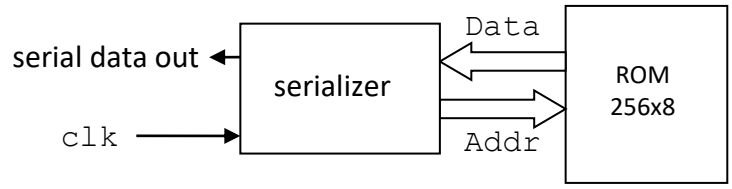
```
entity DME is Port (
   CLKin  : in  STD_LOGIC;
   Din    : in  STD_LOGIC;
   CLKout : inout STD_LOGIC;
   Dout   : inout STD_LOGIC);
end DME;

architecture DME of DME is

begin

  process(CLKin) is begin
    if(falling_edge(CLKin)) then
      CLKout <= not CLKout;
      if(CLKout='1') then
        Dout <= not Dout;
      elsif(Din='0') then
        Dout <= not Dout;
      end if;
    end if;
  end process;
end DME;
```

2. An 8-bits asynchronous (no clock) ROM with 8 address bits (256-bytes) is serialized out one by one, restarting from the x00 address after xFF.



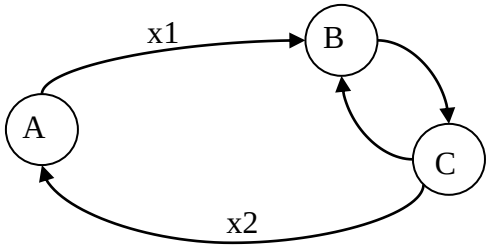

```
entity Seri is Port (
  clk  : in  STD_LOGIC;
  Sout : out STD_LOGIC);
end Seri;

architecture Seri of Seri is
  type MROM is array (0 to 255) of STD_LOGIC_VECTOR(7 downto 0);
  constant ROM : MROM := (.. ROM values here..);
  signal Addr : integer range 0 to 255;
  signal bcnt : integer range 0 to 7;
begin

  process(clk) is begin
    if(rising_edge(clk)) then
      if(bcnt=7) then
        bcnt <= 0;
        if(Addr=255) then
          Addr <= 0;
        else
          Addr <= Addr+1;
        end if;
      end if;
    end if;
  end process;
  Sout <= ROM(Addr)(bcnt);

end Seri;
```

3. Implement the state machine whose diagram is given in the following figure. Output L depends only on the current state and is 1 only when the state is B. It is zero otherwise. State transitions between B and C do not require any input except clock. A to B transition require x1 and C to A transition require x2 inputs to be 1.




```
entity SM is port (
  clk   : in  STD_LOGIC;
  x1,x2 : in  STD_LOGIC;
  L     : out STD_LOGIC);
end SM;

architecture SM of SM is
  type state is (A,B,C);
  signal pstate, nstate : state;

begin
process(clk) is begin
  if(RISING_EDGE(clk)) then
    pstate <= nstate;
  end if;
end process;

L <= '1' when pstate=B else '0';
process(x1,x2) is begin
  if(pstate=A) then
    if(x1='1') nstate <= B; else nstate <= A; end if;
  elsif(pstate=B) then
    nstate <= C;
  elsif(pstate=C) then
    if(x2='1') then nstate <= A; else nstate <= C; end if;
  else
    nstate <= pstate;
  end if;
end process;

end SM;
```