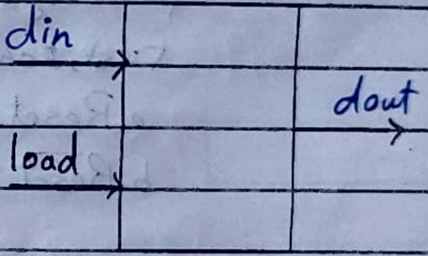
Subject : VLSI L6- Sequential Circuit / /

Sequential circuits must use sequential statments ~~in it~~ inside it's architecture.

Sequential statments :- if else and Swith Case.

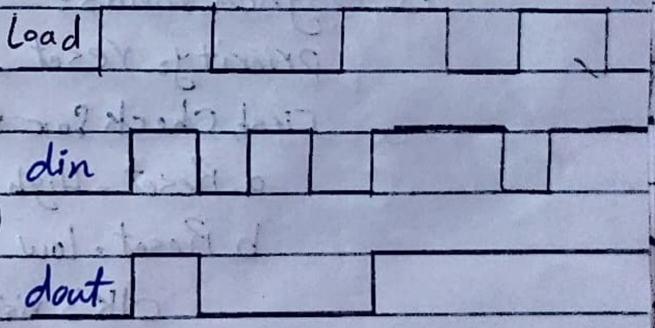
Latches:-

It used to store signal "1-bit".
It doesn't have clk or Reset.
It has Control signal Called "Load", which determine latch's behaviour.
when load is:
ⓐ High → dout <= din
ⓑ Low → dout <= last stored value.



In the Code, we use if without else statment, to latch value.

* Latch a Value: means to Keep this value stored until next Trigger



```
entity latch is
    Port(din, load : in std-logic;
         dout       : out std-logic);
end entity latch;
```

```
architecture rtl of latch is
begin
    process (din, load)
    begin
     if(lead = '1') then
      dout <= din;
     end if;
    endprocess;
end rtl;
```

FlipFlops:- "D-flipflog"

It used to store signal "1-bit".
It has reset and clock (Clk).

- Clk has two modes:-
  a. Rising edge. b. falling edge.

- Reset has two modes:- (Hardware ≈ PUSH Button)
  a. Synchronus ~~~→ In Sync with clk
  priority:- clk → reset
  First, check for clk's rising edge, then check Reset:
  a. Reset = High → q = initial value.
  b. Reset = low → q = din.

  b. Asynchronus:
  priority: reset → clk
  First, check for reset:-
  a. Reset = High → q = initial value.
  b. Reset = low → check for clk's:-
  clk's rising edge → q = din

* Change in sequential circuit o/p depends on clk.*
↳ dout changes at the moment of clk's rising edge and stay fixed between two rising edges.

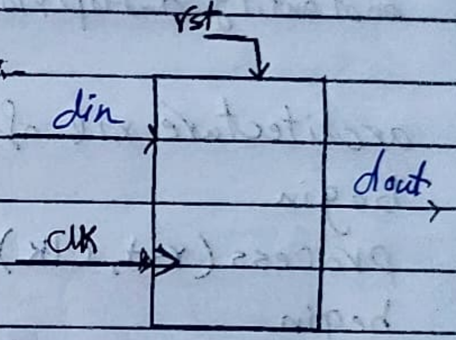
Reset active high:- Reset = 1 → change in o/p
" " low:- Reset = 0 → change in o/p

note: Any statment typed inside a process which it's sensitivity list contains clk only or clk with reset, it translated in hardware to:-

@ flip flop if→ input is signal "1-bit".

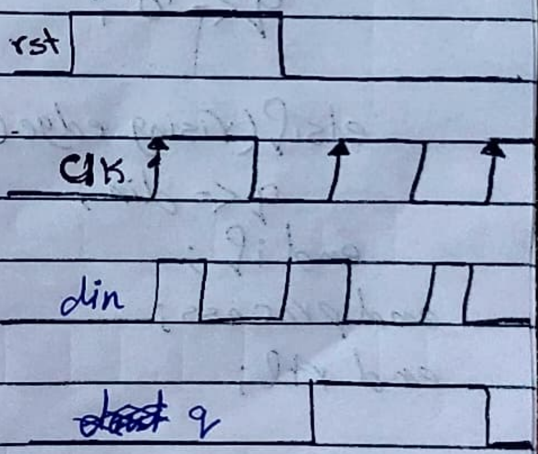
ⓑ Register if→ input is bus. "n-bits".

D flip flop VHDL Code with Synchronus reset:-



```
- entity d_flip_flop is
      Port(din,clk,rst: in std_logic;
           ~~dout~~ q :out std_logic);
  end entity d_flip_flop;

  architecture rtl of d_flip_flop is
    begin
    process(clk)
    begin
    if(rising_edge(clk))then
       if(rst='1')then
          q<='0';
       else
          q<=din;
       end if;
    end if;
    end process;
    end rtl;
```

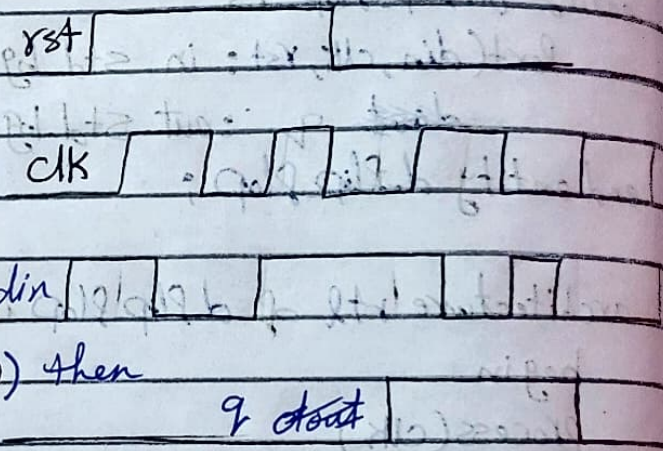


D-flip flop VHDL Code with Asynchronus reset :-

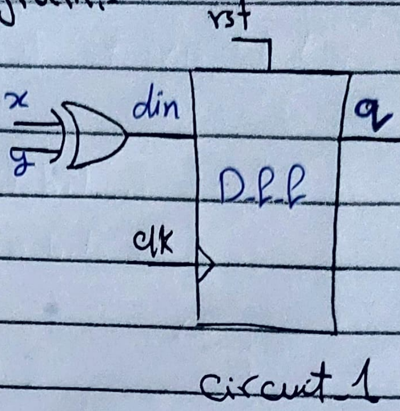
```
entity d_flip_flop is
   Port(din, rst, clk : in std_logic;
                    q  :out std_logic);
end entity d_flipflop;

architecture rtl of d_flip_flop is
begin
process (rst, clk)
begin
    if(rst='1') then
        q<='0';

    elsif(rising_edge(clk)) then
        q<= din;
    end if;
end process;
end rtl;
```



write VHDL Code for the following Diagram:-



```
entity circuit 1 is
      Port(x,Y, clk, rst: in std_logic;
                           q: out std_logic);
end entity circuit_1;

 architecture rtl of circuit 1 is
 begin
        process(clk, rst)
        variable din: std_logic;
        begin
               if(rst='1') then
                   q<='0';

               elsif(rising_edge(clk)) then
                   ~~q<=din~~
                   din<=x XOR y;
                    q<=din;
               endif;

        end processor;
        end rtl;
```

* Variable:- بنستخدمه لما يكون عندي Comparatorier و sequential وعايز الـ o/P
بتاع الـ Comparatorier يفضل زي ما هو ميتحولش لـ sequential.