

**Department of Electronics & Communication Engineering**

**Lab Manual**

**E-CAD Lab**



**Prasad V. Potluri Siddhartha Institute of Technology**

(Sponsored by: Siddhartha Academy of General & Technical Education)

Affiliated to JNTU- Kakinada

Approved by AICTE- New Delhi

Kanuru, Vijayawada-7

## List of Experiments

1. LOGIC GATES
2. IC7474—A POSITIVE EDGE TRIGGERING D FLIP FLOP
3. IC 74x90 – DECADE COUNTER
4. IC 74x93 – 4 -BIT BINARY COUNTER
5. IC 74x95 – SHIFT REGISTER
6. IC 74x194 –UNIVERSAL SHIFT REGISTER
7. 3x8 DECODER
8. IC 74x85 – 4-BIT COMPARATOR
9. 8x1 MULTIPLEXER
10. 16X1 MULTIPLEXER
11. IC 74X189—READ AND WRITE OPERATIONS OF RAM

---

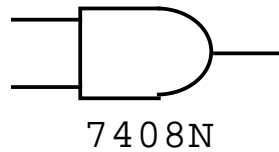
---

## 1.LOGIC GATES

**AIM:** Write a VHDL code for all the logic gates.

**#1-TITLE:** AND gate

**LOGIC GATE SYMBOL:**



**TRUTH TABLE:**

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

**VHDL CODE:**

```
Library IEEE;  
use IEEE.std_logic_1164.all;  
  
entity AND2 is  
    port(  
        x : in STD_LOGIC;  
        y : in STD_LOGIC;  
        z : out STD_LOGIC  
    );  
end AND2;
```

*--Dataflow model*

```
architecture behav1 of AND2 is
begin
```

```
    Z<= x and y;      --Signal Assignment Statement
```

```
end behav1;
```

*-- Behavioral model*

```
architecture behav2 of AND2 is
begin
```

```
    process (x, y)
    begin
```

```
        if (x='1' and y='1') then -- Compare with truth table
```

```
            Z <= '1';
```

```
        else
```

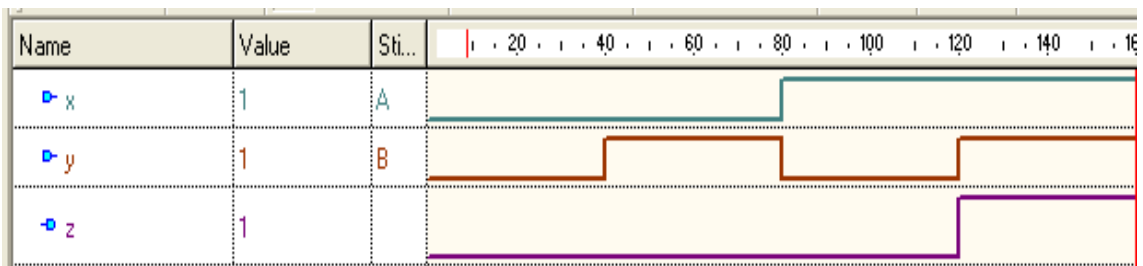
```
            Z <= '0';
```

```
        end if;
```

```
    end process;
```

```
end behav2;
```

### OUT PUT WAVE FORM:

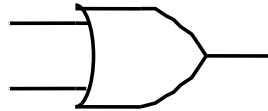


---

---

#2-TITLE: OR gate

LOGIC GATE SYMBOL:



7432

TRUTH TABLE:

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

VHDL CODE:

```
Library IEEE;
use IEEE.std_logic_1164.all;

entity OR2 is
  port(
    x : in STD_LOGIC;
    y : in STD_LOGIC;
    z : out STD_LOGIC
  );
end OR2;

--Dataflow model
architecture behav1 of OR2 is
begin

  Z <= x or y;      --Signal Assignment Statement

end behav1;
```

-- Behavioral model

```
architecture behav2 of OR2 is
begin

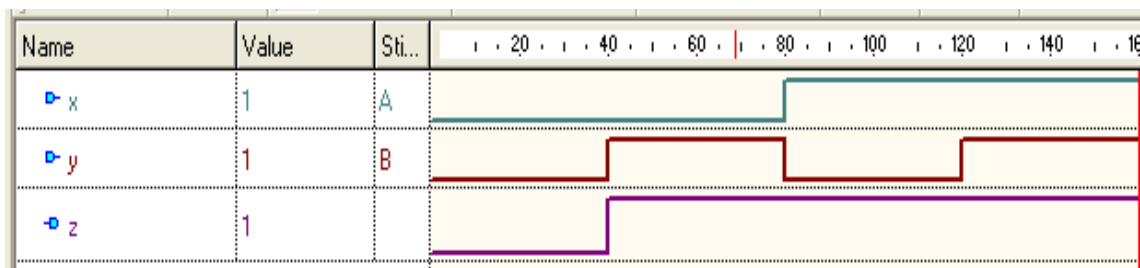
    process (x, y)
    begin

        if (x='0' and y='0') then -- Compare with truth table
            Z <= '0';
        else
            Z <= '1';
        end if;

    end process;

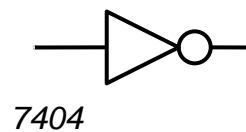
end behav2;
```

**OUTPUT WAVEFORM:**



**#3-TITLE:** NOT gate

**LOGIC GATE SYMBOL:**



**TRUTH TABLE:**

x	z
0	1
1	0

**VHDL CODE:**

```
Library IEEE;
use IEEE.std_logic_1164.all;

entity not1 is
  port(
    X: in STD_LOGIC;
    Z: out STD_LOGIC
  );
end not1;
```

*--Dataflow model*

```
architecture behav1 of not1 is
begin

  Z<= not X; --Signal Assignment Statement

end behav1;
```

*-- Behavioral model*

```
architecture behav2 of not1 is
begin

  process (X)
  begin

    if (x='0') then -- Compare with truth table
      Z <= '1';
    else
      Z <= '0';
    end if;
  end process;
end behav2;
```

```

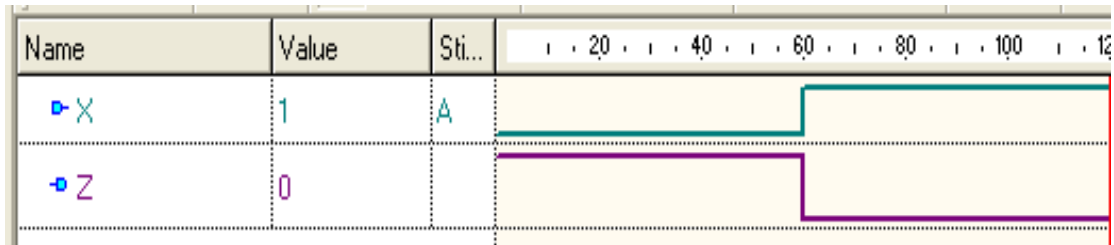
end if;

end process;

end behav2;

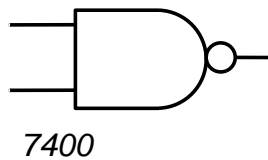
```

**OUTPUT WAVEFORM:**



**#4-TITLE:** NAND gate

**LOGIC GATE SYMBOL:**



**TRUTH TABLE:**

x	y	z
0	0	1
0	1	1
1	0	1
1	1	0



---

**VHDL CODE:**

```
Library IEEE;
use IEEE.std_logic_1164.all;

entity nand2 is
    port(
        x : in STD_LOGIC;
        y : in STD_LOGIC;
        z : out STD_LOGIC
    );
end nand2;
```

*--Dataflow model*

```
architecture behav1 of nand2 is
begin

    z <= x nand y;      --Signal Assignment Statement

end behav1;
```

*-- Behavioral model*

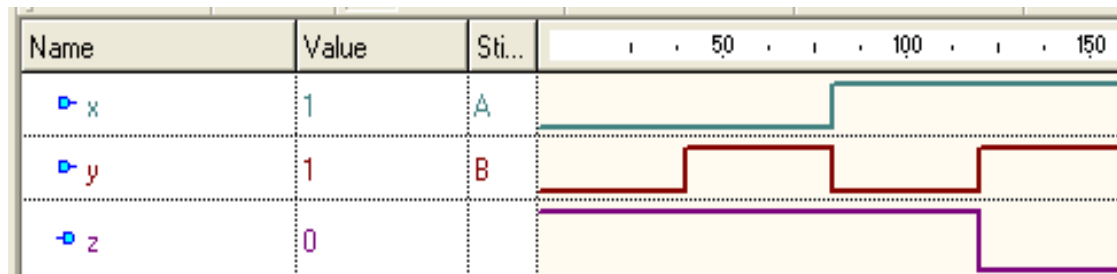
```
architecture behav2 of nand2 is
begin

    Process (x, y)
    Begin

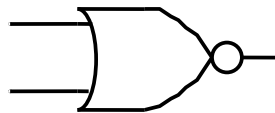
        If (x='1' and y='1') then -- Compare with truth table
            Z <= '0';
        else
            Z <= '1';
        end if;

    end process;

end behav2;
```

**OUTPUT WAVEFORM:**

#5- TITLE: NOR gate

**LOGIC GATE SYMBOL:**

7402

**TRUTH TABLE:**

x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

**VHDL CODE:**

```

Library IEEE;
use IEEE.std_logic_1164.all;

entity nor2 is
  Port (
    X: in STD_LOGIC;
    Y: in STD_LOGIC;
    Z: out STD_LOGIC
  );
end nor2;

```

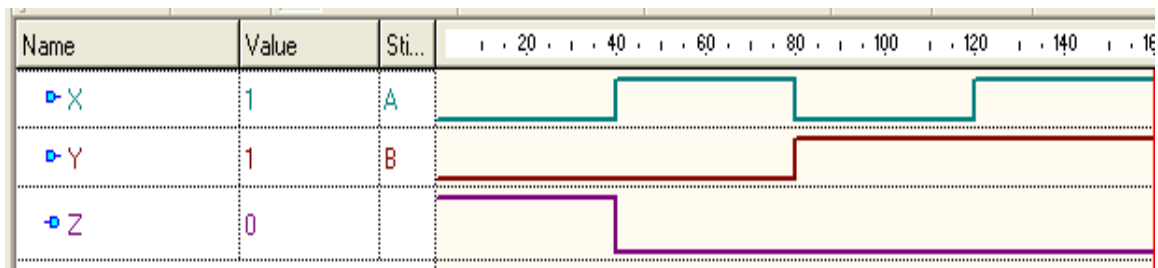
--Dataflow model

```
architecture behav1 of nor2 is
begin
    Z<= x nor y; --Signal Assignment Statement
end behav1;
```

-- Behavioral model

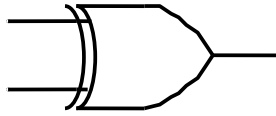
```
architecture behav2 of nor2 is
begin
    process (x, y)
    begin
        If (x='0' and y='0') then -- Compare with truth table
            Z <= '1';
        else
            Z <= '0';
        end if;
    end process;
end behav2;
```

**OUTPUT WAVEFORM:**



#6-TITLE: EX-OR gate

LOGIC GATE SYMBOL:



TRUTH TABLE:

7486

x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

VHDL CODE:

```

Library IEEE;
use IEEE.std_logic_1164.all;

entity xor2 is
  Port (
    X: in STD_LOGIC;
    Y: in STD_LOGIC;
    Z: out STD_LOGIC
  );
end xor2;
```

*--Dataflow model*

```

architecture behav1 of xor2 is
begin

  Z<= x xor y; --Signal Assignment Statement

end behav1;
```

**-- Behavioral model**

```

architecture behav2 of xor2 is
begin

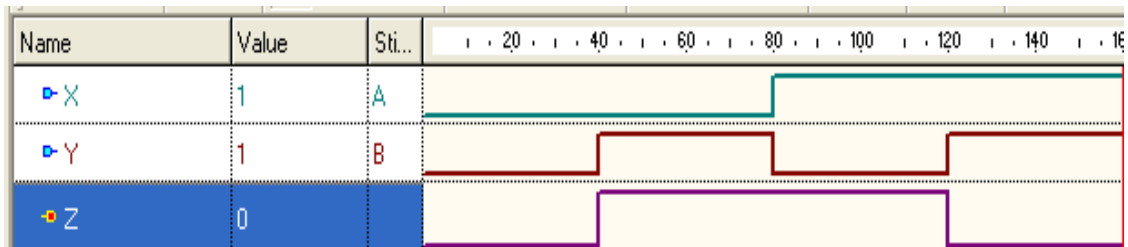
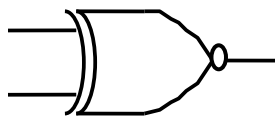
    process (x, y)
    begin

        If (x/=y) then    -- Compare with truth table
            Z <= '1';
        else
            Z <= '0';
        end if;

    end process;

end behav2;

```

**OUTPUT WAVEFORM:****#7-TITLE:** EX-NOR gate**LOGIC GATE SYMBOL:****74135**

**TRUTH TABLE:**

x	y	z
0	0	1
0	1	0
1	0	0
1	1	1

**VHDL CODE:**

```
Library IEEE;  
use IEEE.std_logic_1164.all;
```

```
entity xnor2 is  
    Port (  
        X: in STD_LOGIC;  
        Y: in STD_LOGIC;  
        Z: out STD_LOGIC  
    );  
end xnor2;
```

*--Dataflow model*

```
architecture behav1 of xnor2 is  
begin  
    Z<= x xnor y; --Signal Assignment Statement  
end behav1;
```

*-- Behavioral model*

```
architecture behav2 of xnor2 is  
begin  
    process (x, y)  
    begin
```

```

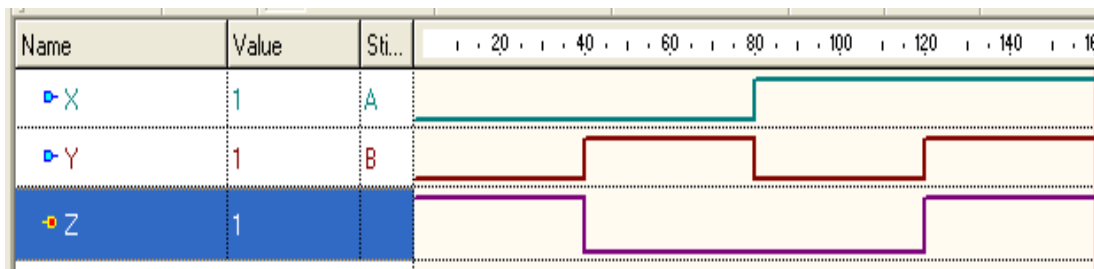
    If (x=y) then      -- Compare with truth table
        Z <= '1';
    else
        Z <= '0';
    end if;

end process;

end behav2;

```

### OUTPUT WAVEFORM:



### VIVA QUESTIONS:

1. Implement the following function using VHDL coding. (Try to minimize if you can).

$$F(A,B,C,D)=(A'+B+C) \cdot (A+B'+D') \cdot (B+C'+D') \cdot (A+B+C+D)$$

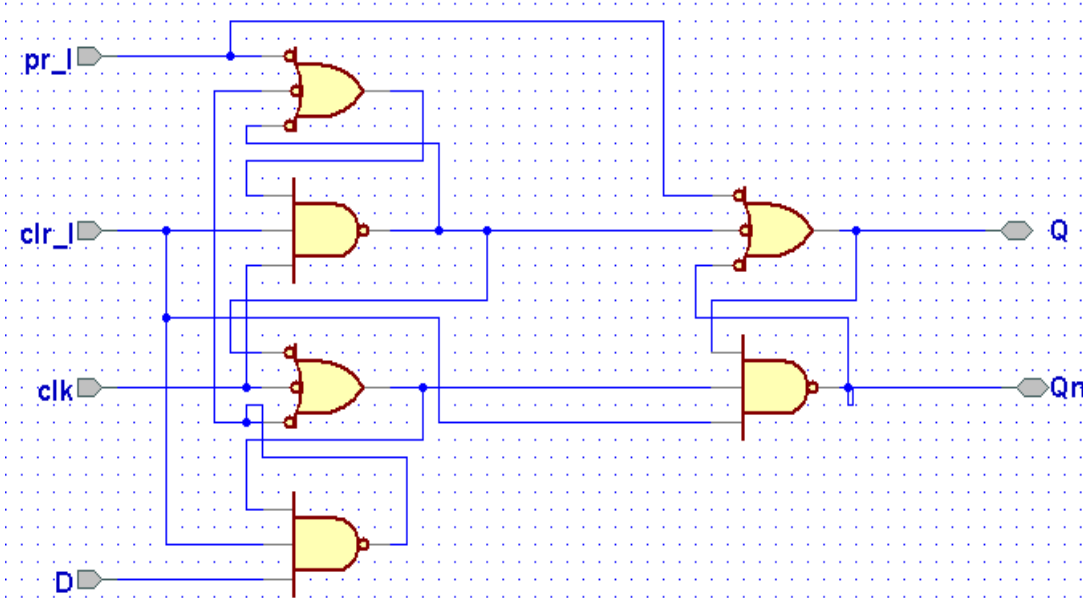
2. What will be the no. of rows in the truth table of N variables?
3. What are the advantages of VHDL?
4. Design Ex-OR gate using behavioral model?
5. Implement the following function using VHDL code  
 $f=AB+CD$ .
6. What are the differences between half adder and full adder?
7. What are the advantages of minimizing the logical expressions?
8. What does a combinational circuit mean?
9. Implement the half adder using VHDL code?
10. Implement the full adder using two half adders and write VHDL program in structural model?

**2.IC7474—A POSITIVE EDGE TRIGGERING D FLIP FLOP**

**AIM:** Write a VHDL code for IC7474—a positive edge triggering D flip flop.

**TITLE:** IC7474—a positive edge triggering D flip flop.

**CIRCUIT DIAGRAM:**



**TRUTH TABLE:**

clr_l	pr_l	Clk	d	q	qn
0	0	X	X	1	1
0	1	X	X	0	1
1	0	X	X	1	0
1	1		0	0	1
1	1		1	1	0



**VHDL CODE:**

*--VHDL code for the circuit*

```

library IEEE;
use ieee.std_logic_1164.all;
entity dff is
    port (
        pr_1: in STD_LOGIC;    -- active low preset input
        clr_1: in STD_LOGIC;    -- active low clear input
        clk :in STD_LOGIC;      -- clock input
        d   :in STD_LOGIC;      -- D input
        q   :inout STD_LOGIC;   -- output of D flip flop
        qn  :inout STD_LOGIC    -- inverted output
    );
end dff;
architecture dff of dff is
    signal e,f,g,h:std_logic;
    component nand3
    port (
        a,b,c: in STD_LOGIC;
        d    : out STD_LOGIC
    );
end component;
begin
    g1:nand3 port map(pr_1,h,f,e);    -- creates g1 gate
    g2:nand3 port map(clr_1,e,clk,f); -- creates g2 gate
    g3:nand3 port map(f,clk,h,g);     -- creates g3 gate
    g4:nand3 port map(g,clr_1,d,h);   -- creates g4 gate
    g5:nand3 port map(pr_1,f,qn,q);   -- creates g5 gate
    g6:nand3 port map(q,g,clr_1,qn);  -- creates g6 gate
end dff;

```

*--VHDL code for 3 i/p nand gate*

```

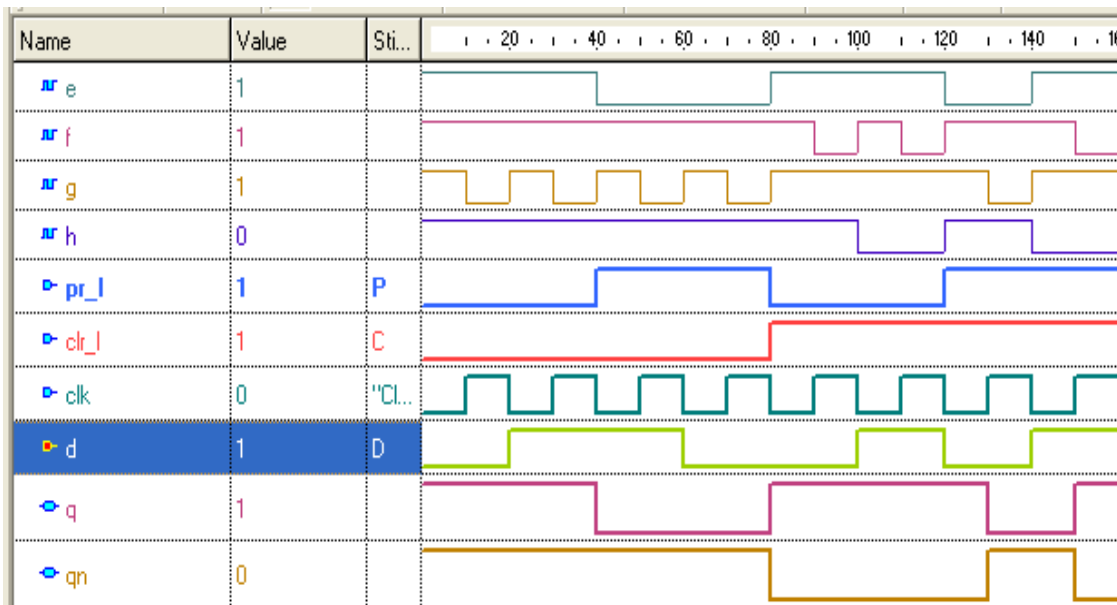
library IEEE;
use IEEE.std_logic_1164.all;
entity nand3 is
    port (
        a,b,c: in STD_LOGIC;
        d    : out STD_LOGIC
    );
end nand3;

```

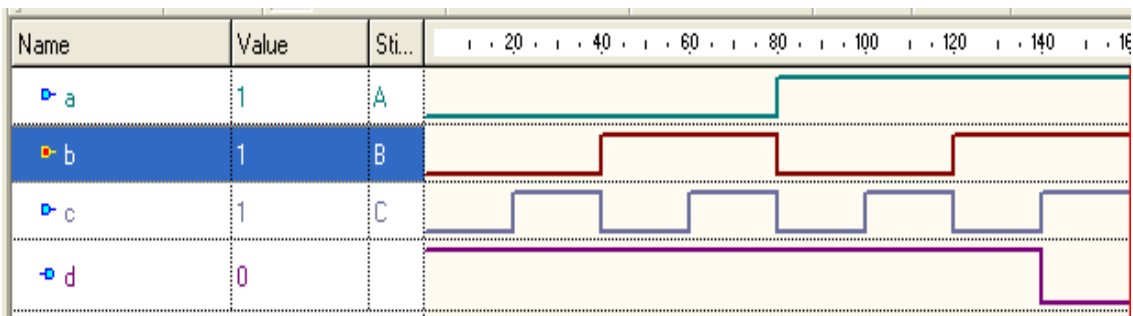
```
architecture \nand\ of nand3 is
begin
d<= not (a and b and c); -- creates a 3 i/p nand gate
end \nand\;
```

**WAVEFORMS:**

**D FLIPFLOP**



**NAND GATE**



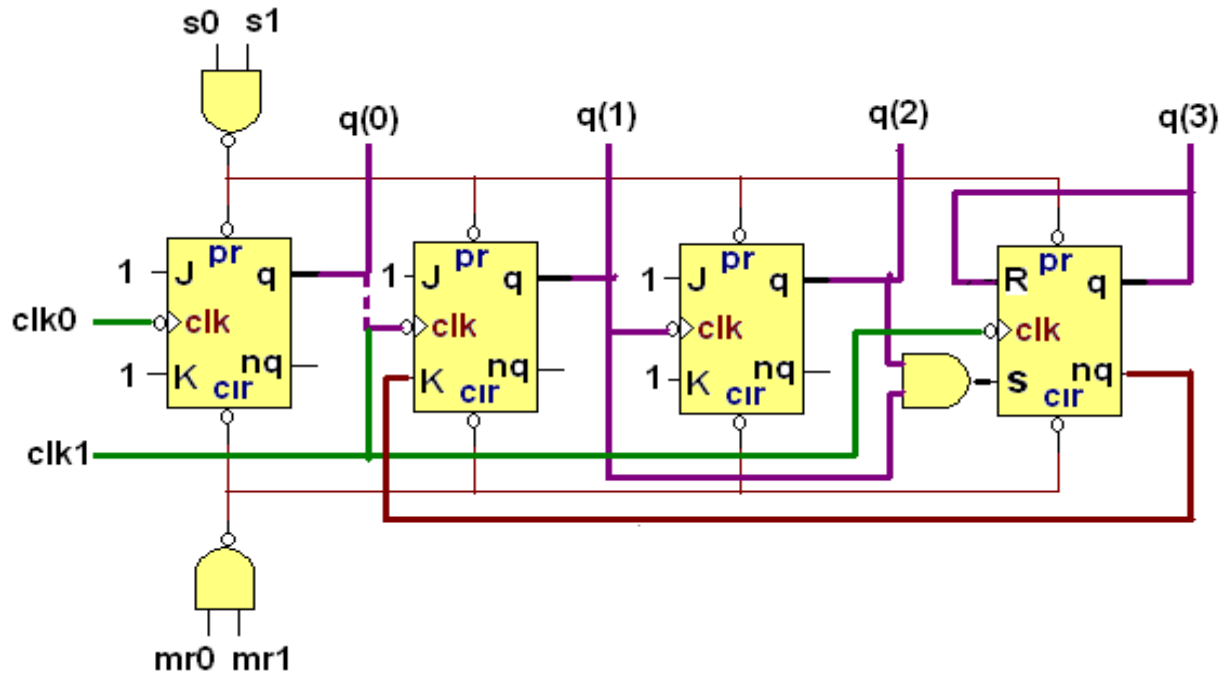
VIVA QUESTIONS:

1. Write the behavioral code for the IC 74x74.
2. Write the dataflow code for the IC 74x74.
3. What is the difference between sequential and combinational circuit?
4. What is a flip-flop?
5. Explain the functions of preset and clear inputs in flip-flop?
6. What is meant by a clocked flip-flop?
7. What is meant by excitation table?
8. What is the difference between flip-flop and latch?
9. What are the various methods used for triggering flip-flops?
10. Explain level triggered flip-flop?
11. Write the behavioral code for IC 74X74.
12. Write the syntax of IF statement?

### 3.IC 74x90 – DECADE COUNTER

**AIM:** To write the VHDL code for IC 74x90 – decade counter.

**CIRCUIT DIAGRAM OF IC 74x90:**



**TRUTH TABLE:**

OUTPUT			
Q(0)	Q(3)	Q(2)	Q(1)
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0

## VHDL CODE:

*--To work as a decade counter*

library IEEE;

Use IEEE.std\_logic\_1164.all;

entity count is

port (

S0, s1, r0, r1: in STD\_LOGIC; *--set and reset i/ps for mod2 and  
-- Mod5 counters*

Clk0: in STD\_LOGIC; *--Clock signal for mod2 counter*

Clk1: inout STD\_LOGIC; *--Clock signal for mod5 counter*

q : inout STD\_LOGIC\_VECTOR(3 downto 0) *--o/p of  
-- mod2 X mod5= mod10*

);

end count;

architecture count of count is

component jk\_ff *-- jk flip flop instantiation*

port (

jk : in STD\_LOGIC\_VECTOR(1 downto 0);

clk,pr\_1,clr\_1 : in STD\_LOGIC;

q,nq : inout STD\_LOGIC

);

end component;

signal preset,clear,S, q3bar:STD\_LOGIC;

begin

preset <= s0 nand s1; *-- common preset inputs for mod2 and mod5 counters*

clear <=r0 nand r1; *-- common reset inputs for mod2 and mod5 counters*

S<=q(2) and q(1); *-- to set the last flip flop*

q3bar <= not q(3); *-- complemented output of q(3)*

clk1<=q(0); *--to work as asynchronous mod10 counter*

jk1:jk\_ff port map("11",clk0,preset,clear,q(0),open);

jk2:jk\_ff port map(jk(1)=> q3bar,

jk(0)=>'1',

clk=>clk1,

pr\_1=>preset,

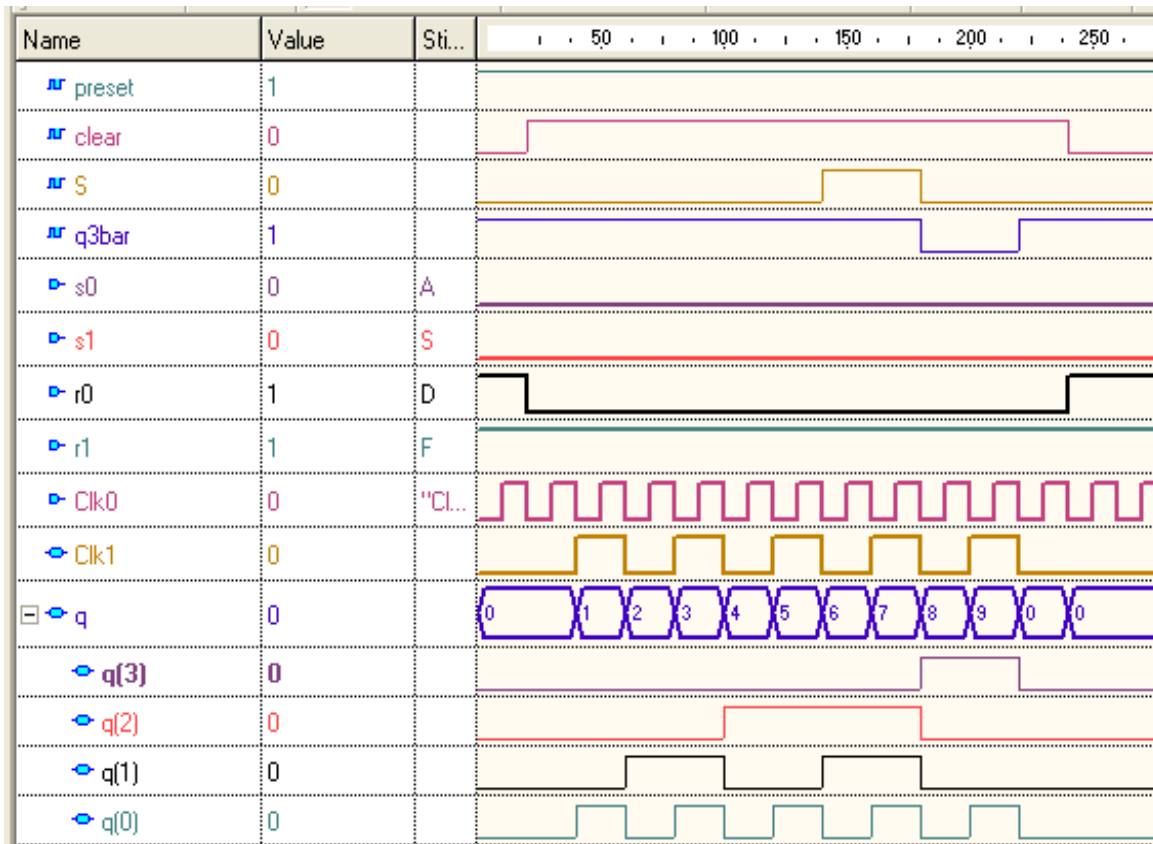
```

        clr_l=>clear,
        q=>q(1),
        nq=>open); -- jk1,jk2,jk3,jk4 create four JK flip flops
jk3:jk_ff port map("11",q(1),preset,clear,q(2),open);
jk4:jk_ff port map(jk(0)=>q(3),
        jk(1)=>s,
        clk=>clk1,
        pr_l=>preset,
        clr_l=>clear,
        q=>q(3),
        nq=> q3bar);

end count;

```

### WAVEFORMS:



---



---

*--Program for JK flip-flop*

```

library IEEE;
use IEEE.std_logic_1164.all;

entity jk_ff is
  port (
    jk          : in STD_LOGIC_VECTOR(1 downto 0);
                                                    --jk(1)=J;jk(0)=K;
    clk,pr_l,clr_l : in STD_LOGIC;
    q,nq        : inout STD_LOGIC
  );
end jk_ff;

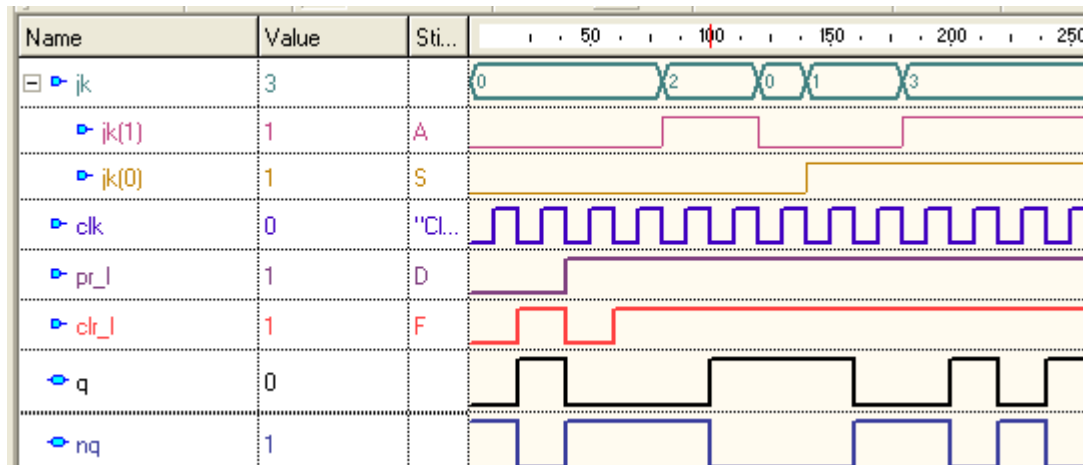
architecture jk of jk_ff is
begin
  process(clk,pr_l,clr_l,jk)
  variable temp:std_logic:='0';
  begin
    q<='0';nq<='1';
    if (pr_l='1' and clr_l='0') then
      q<='0';nq<='1';
    elsif (pr_l='0' and clr_l='1') then
      q<='1';nq<='0';
    elsif (pr_l='1' and clr_l='1') then
      if (clk 'event and clk='0') then --performs during the falling edge of clock

        case jk is
          when "00"=>temp:=temp;
          when "01"=>temp:='0';
          when "10"=>temp:='1';
          when "11"=>temp:=not temp;
          when others=>null;
        end case;

      end if;
      q<=temp;
      nq<= not temp;
    end if;
  end process;
end jk;

```

## WAVEFORMS:



## VIVA QUESTIONS:

1. Write the behavioral code for IC 74x90.
1. What is a sequential circuit?
2. Differentiate between synchronous and asynchronous counter?
3. How many no. of flip-flops are required for decade counter?
4. What is meant by excitation table?
5. What are the meanings of different types of values in std\_ulogic?
6. What are the objects in VHDL?
7. Write the syntax for a signal?
8. Write the difference between signal and variable?
9. Explain about enumeration types?
10. If the modulus of a counter is 12 how many flip-flops are required?



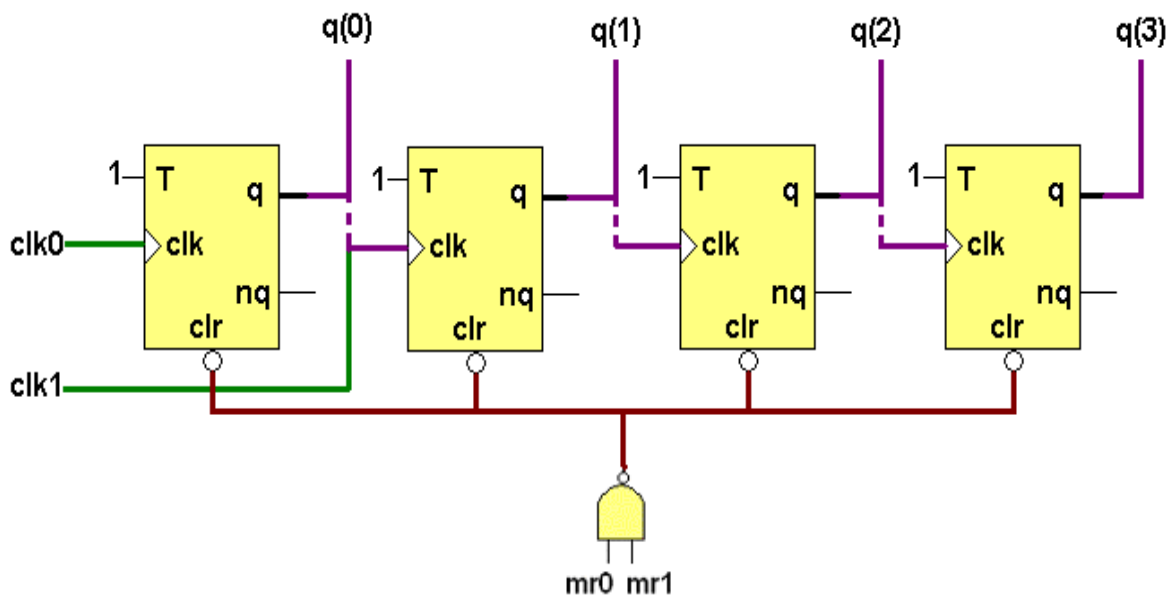
### 4.IC 74x93 – 4 -BIT BINARY COUNTER

**AIM:** To write the VHDL code for IC 74x93 – 4 -bit binary counter.

**TRUTH TABLE:**

OUTPUT			
Q(3)	Q(2)	Q(1)	Q(0)
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

**CIRCUIT DIAGRAM OF IC74X93:**



**VHDL CODE:**

*--Program for 4-bit counter*

```

library IEEE;
use IEEE.std_logic_1164.all;

entity cnt is
    port (
        clk0: in STD_LOGIC;
        mr0: in STD_LOGIC;
        mr1: in STD_LOGIC;
        clk1: inout STD_LOGIC;
        Q: inout STD_LOGIC_VECTOR(3 downto 0)
    );
end cnt;

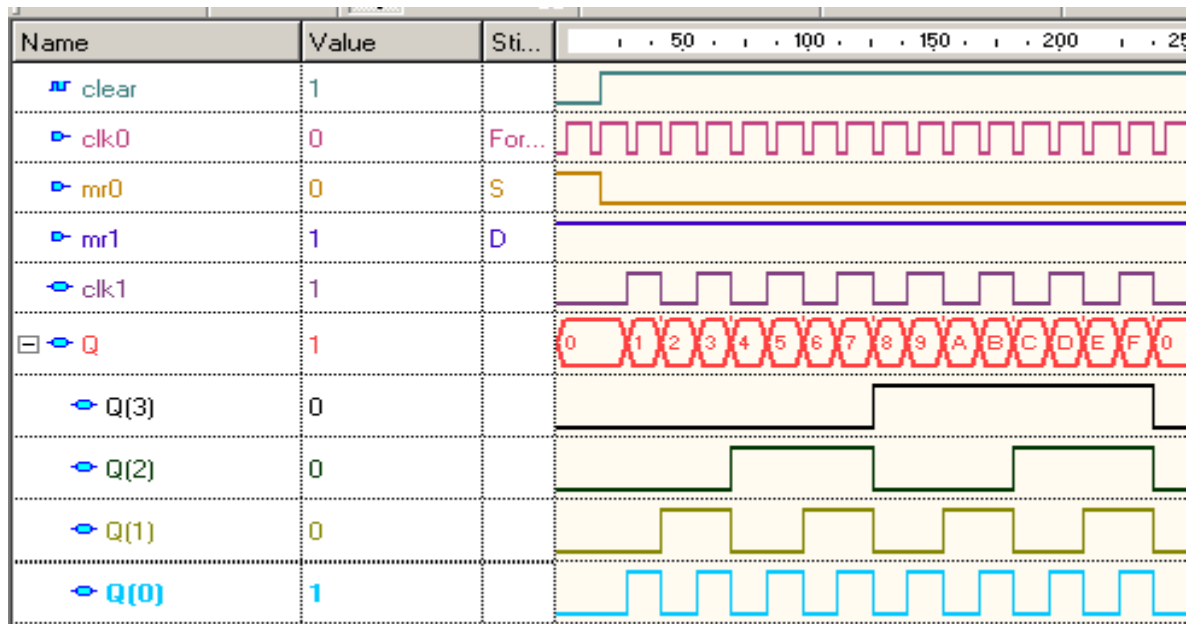
architecture cnt of cnt is

Component tff -- T- flip flop instantiation
    port (
        t      : in STD_LOGIC;
        clk   : in STD_LOGIC;
        clr_l : in STD_LOGIC;
        q,nq  : out STD_LOGIC
    );
end component;
signal clear : std_logic;
begin
    clear<= mr0 nand mr1; -- common reset inputs for mod2 and mod8
                        --counters
    CLK1<=q(0); --to work as asynchronous mod16 counter
    t1:tff port map('1',clk0,clear,Q(0),open);--t1,t2,t3,t4 create four T-flip flops
    t2:tff port map('1',clk1,clear,Q(1), open);
    t3:tff port map('1',Q(1),clear,Q(2), open);
    t4:tff port map('1',Q(2),clear,Q(3), open);

end cnt;

```

## WAVEFORMS:



*--Program for T flip-flop*

```

library IEEE;
use IEEE.std_logic_1164.all;

entity tff is
    port (
        t    : in STD_LOGIC;--input to the T-flip flop
        clk  : in STD_LOGIC;--Clock signal for T-flip flop
        clr_l : in STD_LOGIC;--active low clear input
        q,nq : out STD_LOGIC--actual and complemented outputs of T-flip flop
    );
end tff;

architecture tff of tff is
begin
    process(t,clk,clr_l)
        variable temp:STD_LOGIC:= '0';
    begin
        if (clr_l='0') then
            temp:= '0';
        end if;
    end process;
end architecture tff;

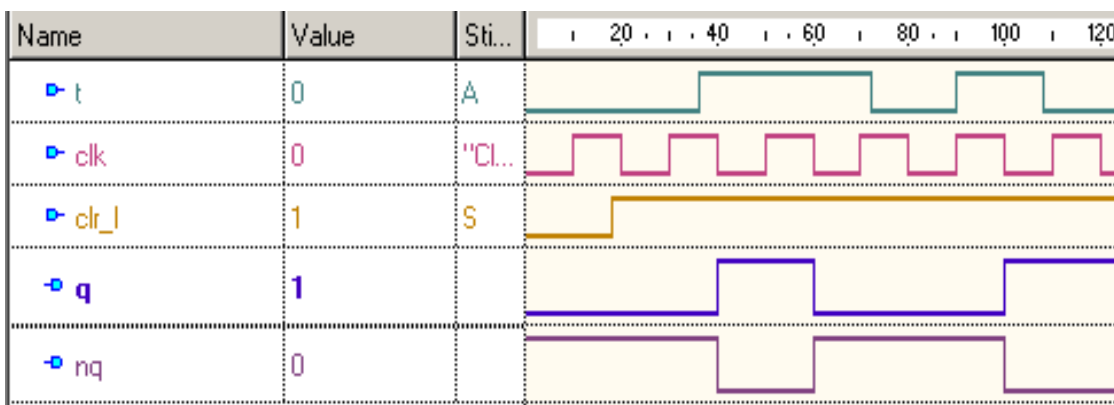
```

```

elsif ((clr_l='1') and (clk'event and clk='0')) then--performs during falling edge
  if ( t='0') then
    temp:=temp;
  else temp:= not temp;
  end if;
end if;
q<= temp;
nq<= not temp;
end process;
end tff;

```

### WAVEFORMS:



### VIVA QUESTIONS:

1. Write the behavioral code for IC 74x93.
2. What is the difference between decade counter and 4 bit counter?
3. What is meant by a modulus of a counter?
4. Write the behavioral code for IC74X93?
5. Explain the operation of IC74X93?
6. Write the syntax for component instantiation?
7. What is netlist?
8. Briefly explain about generics?
9. Write the difference between sequential statement and concurrent statement?
10. Write the syntax for loop statements?
11. Write the syntax for generate statements?
12. Write the differences between loop and generate?

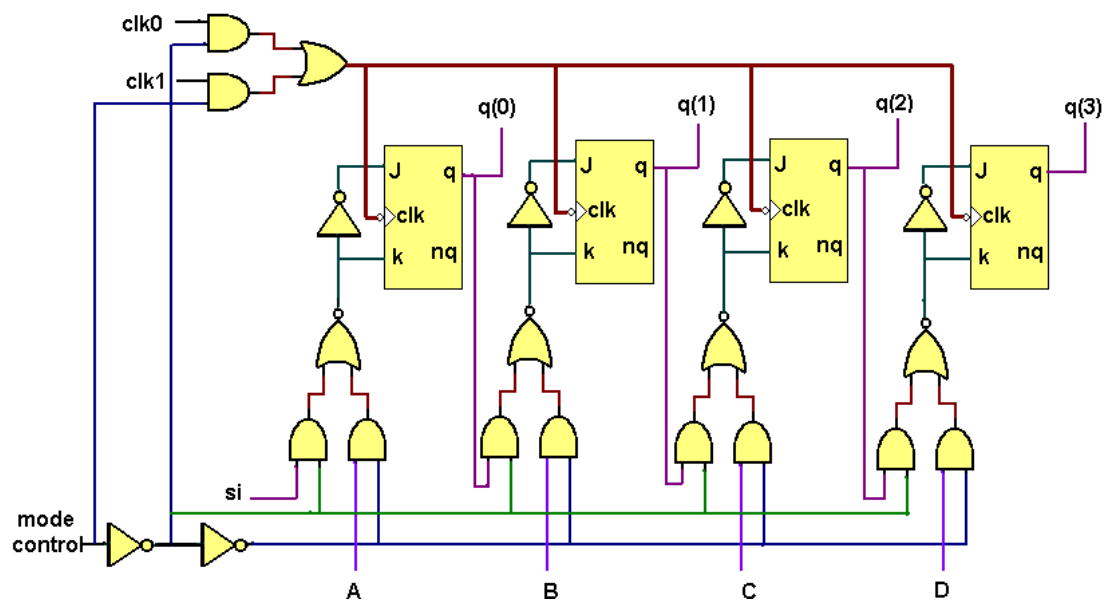
### 5.IC 74x95 – SHIFT REGISTER

**AIM:** To write the structural program for IC 74x95 – SHIFT REGISTER.

#### TRUTH TABLE:

mode control	clock	function
0	clk0	<b>Serial operation</b> q(2) to q(3), q(1) to q(2), q(0) to q(1), si to q(0)
1	clk1	<b>Parallel operation</b> A to q(0) B to q(1) C to q(2) D to q(3)

#### CIRCUIT DIAGRAM OF IC 74X95:



**VHDL CODE:***--Structural model**--Program for shift register*

```

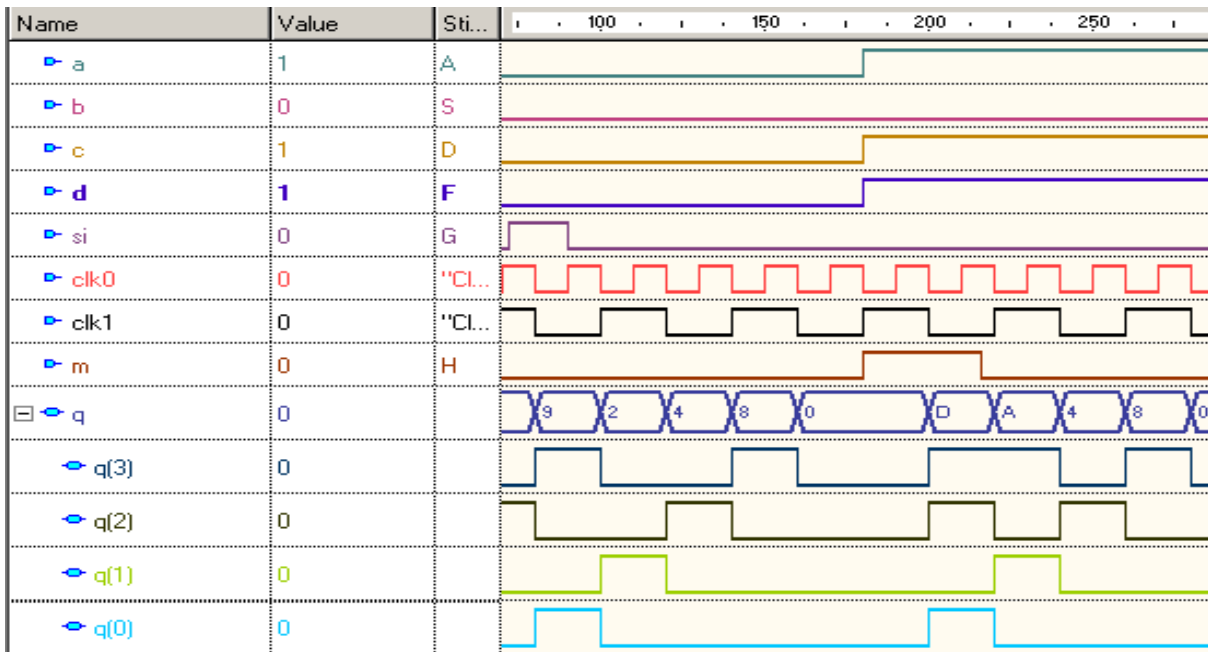
library IEEE;
use IEEE.std_logic_1164.all;

entity shift_reg is
    port (
        a,b,c,d: in STD_LOGIC;  --four parallel inputs
        si      : in STD_LOGIC;  --one serial input
        m       : in STD_LOGIC;  --mode control
        clk0    : in STD_LOGIC;  --clock for serial input
        clk1    : in STD_LOGIC;  --clock for parallel input
        q       : inout STD_LOGIC_VECTOR (3 downto 0) --4-bit output
    );
end shift_reg;

architecture shift_reg of shift_reg is
    component mux  -- multiplexer instantiation
        port (
            a,b,c,d: in STD_LOGIC;
            z      : out STD_LOGIC
        );
    end component ;
    component dff  -- D- flip flop instantiation
        port (
            d,clk: in STD_LOGIC;
            q     : out STD_LOGIC
        );
    end component;
    signal nm,c0,do,d1,d2,d3:STD_LOGIC;
begin
    nm<= not m;
    g1:mux port map(clk0,nm,clk1,m,c0);  --to select the clock based on mode
                                         -- control
    g2:mux port map(si,nm,a,m,do);       --g2,g3,g4,g5 are used to select
    g3:mux port map(q(0),nm,b,m,d1);     --either serial input or parallel input
    g4:mux port map(q(1),nm,c,m,d2);     --based on mode control
    g5:mux port map(q(2),nm,d,m,d3);
    d11:dff port map(do,c0,q(0));        --d11,d12,d13,d14 creates four D flip flops
    d12:dff port map(d1,c0,q(1));        --to perform either serial or parallel shift
    d13:dff port map(d2,c0,q(2));        -- operations
    d14:dff port map(d3,c0,q(3));
end shift_reg;

```

## WAVEFORMS:

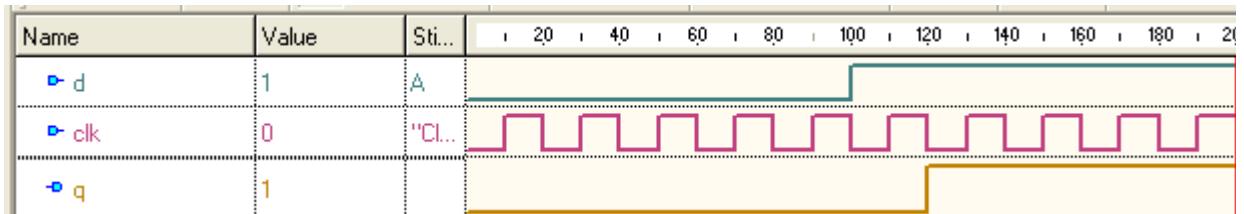


*--program for D-flip-flop*

```
library IEEE;
use IEEE.std_logic_1164.all;
```

```
entity dff is
  port (
    d,clk: in STD_LOGIC;
    q      : out STD_LOGIC
  );
end dff;
```

```
architecture dff of dff is
begin
  process(clk)
  begin
    if( clk'event and clk='0') then      --performs during falling edge
      q<=d;
    else null;
    end if;
  end process;
end dff;
```

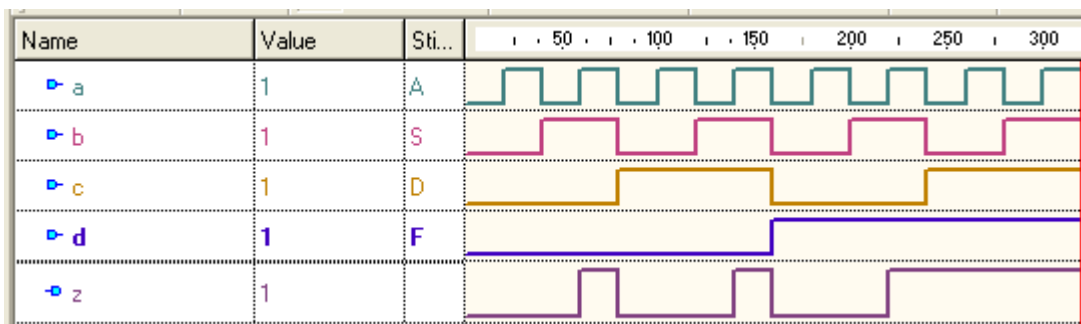
**WAVEFORMS:**

--Program for multiplexer

```
library ieee;
use ieee.std_logic_1164.all;

entity mux is
    port (
        a,b,c,d: in STD_LOGIC;
        z      : out STD_LOGIC
    );
end mux;

architecture mux of mux is
begin
    z<=((a and b) or (c and d));
end mux;
```

**WAVEFORMS:**



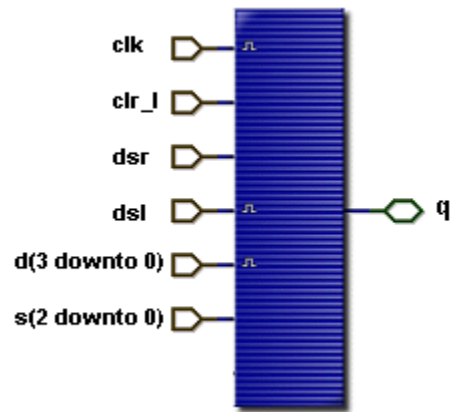
VIVA QUESTIONS:

1. Write the behavioral code for IC 74x95.
2. What is a shift register?
3. Write some applications of shift register?
4. Explain briefly about BLOCK?
5. Write the syntax for function?
6. Write the syntax for procedure?
7. How to define variable in VHDL?
8. Write the syntax for CASE statement?
9. What is the advantage of case statement over if-else statement?
10. Write the difference between with-select and when-else statement?





## 6.IC 74x194 –UNIVERSAL SHIFT REGISTER

**AIM:** To write the VHDL code for IC 74x194 –universal shift register.

**BLOCK DIAGRAM:**



**TRUTH TABLE:**

Clr_1	S(1)	S(0)	Clk	Output function
0	X	X	X	1
1	0	0		no change
1	0	1		shift right ( dsr to q(0))
1	1	0		shift left ( dsl to q(3))
1	1	1		load data (parallel shifting)

**VHDL code:**

```

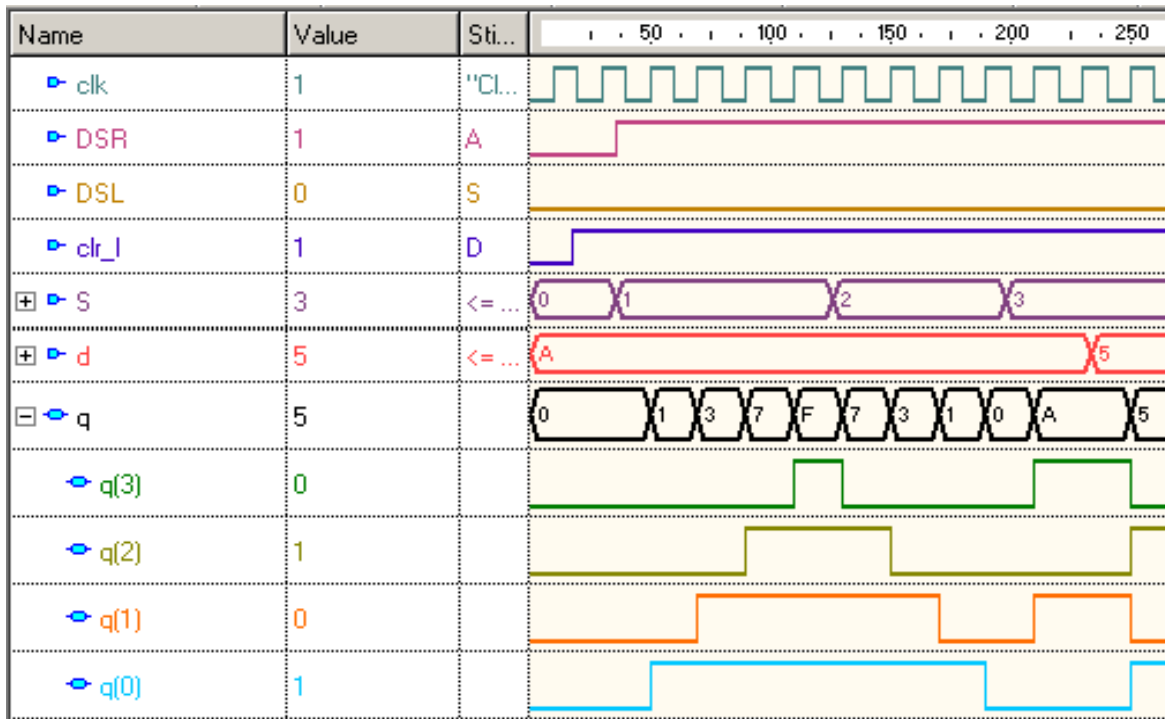
library IEEE;
use IEEE.std_logic_1164.all;

entity shift194 is
  port (
    clk      : in STD_LOGIC;--Clock signal
    dsr,dsl  : in STD_LOGIC;--serial input for right shift and left shift
                                --operation
    clr_l    : in STD_LOGIC;--active low clear input
    S:in STD_LOGIC_VECTOR(1 downto 0);--mode control bits
    d: in STD_LOGIC_VECTOR (3 downto 0);--four parallel input bits
    q: inout STD_LOGIC_VECTOR (3 downto 0) --4-bit output
  );
end shift194;

architecture shift194 of shift194 is
begin
  process(clk,s,clr_l)
  begin
    if clr_l='0' then
      q<=(others=>'0');
    elsif clr_l='1' then
      if(clk'event and clk='1') then
        case s is
          when "00" =>q<=q;--no change
          when "01"=>q<=q(2 downto 0) & dsr;--shift right(dsr to q(0))
          when "10" =>q<=dsl & q(3 downto 1);--shift left(dsl to q(3))
          when "11" =>q<=d(3) & d(2) & d(1) & d(0);--parallel operation
                                --d(3) to q(3),d(2) to q(2),d(1) to q(1),d(0) to q(0)
          when others=>null;
        end case;
      end if;
    end if;
  end process;
end shift194;

```

WAVEFORMS:

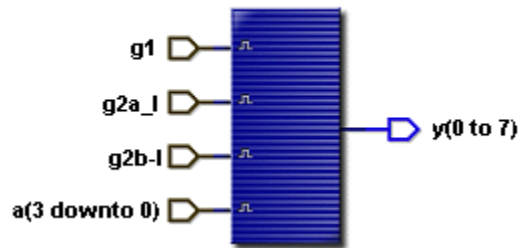


### 7.3x8 DECODER

**AIM:** Write a VHDL code for IC74138 -3X8 Decoder

**TITLE:** IC74138—3x8 Decoder.

**BLOCK DIAGRAM:**



**TRUTH TABLE:**

S.No	Enable inputs			Encoded inputs			Decoded output
	g1	g2a_1	g2b_1	A	B	C	
1	0	X	X	X	X	X	11111111
2	1	1	X	X	X	X	11111111
3	1	X	1	X	X	X	11111111
4	1	0	0	0	0	0	01111111
5	1	0	0	0	0	1	10111111
6	1	0	0	0	1	0	11011111
7	1	0	0	0	1	1	11101111
8	1	0	0	1	0	0	11110111
9	1	0	0	1	0	1	11111011
10	1	0	0	1	1	0	11111101
11	1	0	0	1	1	1	11111110

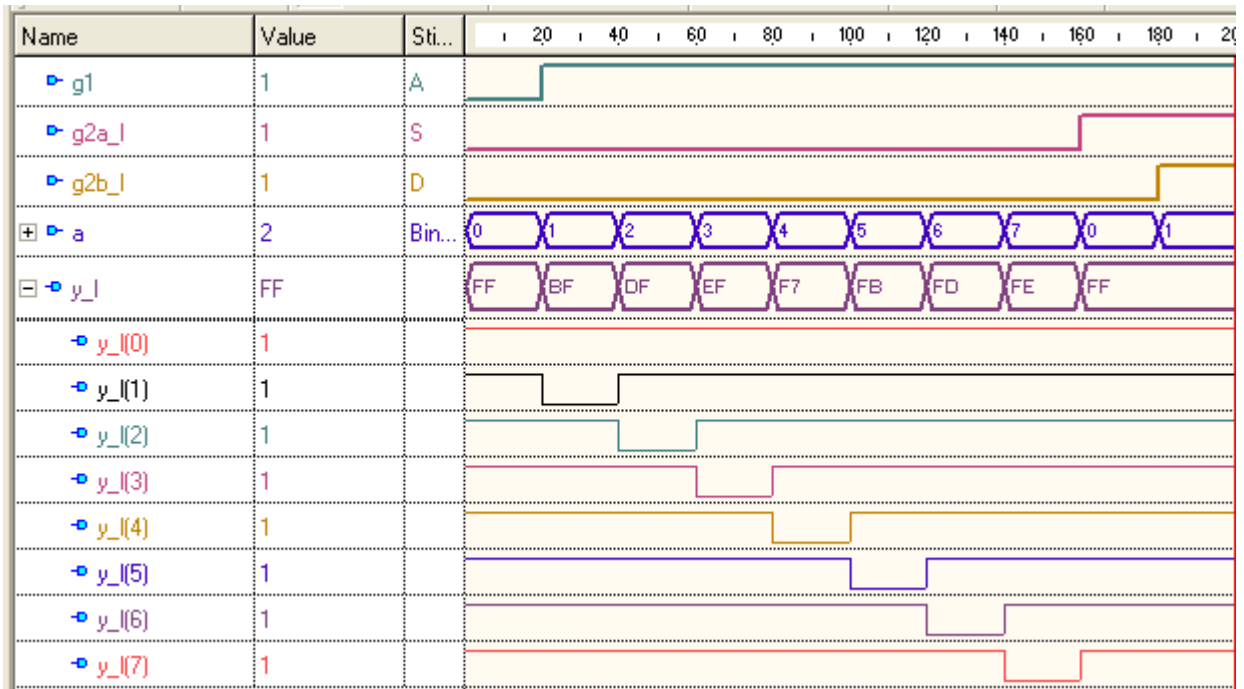
**VHDL CODE:**

```
library IEEE;
use IEEE.std_logic_1164.all;

entity decoder3X8 is
    port (
        g1      : in STD_LOGIC;--g1, g2a_1, g2b_1 cascade i/ps
        g2a_1   : in STD_LOGIC;
        g2b_1   : in STD_LOGIC;
        a       : in STD_LOGIC_VECTOR (2 downto 0);
        y_1     : out STD_LOGIC_VECTOR (0 to 7)
    );
end decoder3X8;

architecture deco38 of decoder3X8 is
begin
    process (a,g1,g2a_1,g2b_1)
    begin
        if (g1 and not g2a_1 and not g2b_1)='1'then
            if a <= "000"then y_1<= "01111111";
            elsif a <= "001"then y_1 <= "10111111";
            elsif a <= "010"then y_1<= "11011111";
            elsif a <= "011"then y_1 <= "11101111";
            elsif a <= "100"then y_1 <= "11110111";
            elsif a <= "101"then y_1 <= "11111011";
            elsif a <= "110"then y_1 <= "11111101";
            elsif a <= "111"then y_1 <= "11111110";
            else y_1<= "11111111";
            end if;
        else y_1 <= "11111111";
        end if;
    end process;
end deco38;
```

## WAVEFORMS:



## VIVA QUESTIONS

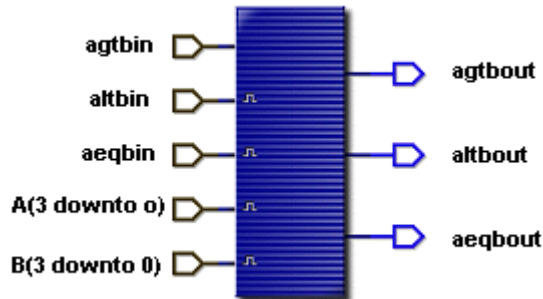
:

1. Write the behavioral code for the IC 74x138.
2. Write the VHDL code for the IC 74x138 using CASE statement.
3. Write the VHDL code for the IC 74x138 using WITH statement.
4. Write the VHDL code for the IC 74x138 using WHEN--ELSE statement.
5. Write the structural program for IC 74x138.
6. What does priority encoder mean?
7. How many decoders are needed to construct 4X16 decoder?
8. What is the difference between decoder and encoder?
9. Write the syntax for exit statement?
10. Explain briefly about next statement?
11. How to specify the delay in VHDL program?
12. Write the syntax for component declaration.

## 8.IC 74x85 – 4-BIT COMPARATOR

**AIM:** Write a VHDL code for IC 74x85 –4-bit comparator .

### BLOCK DIAGRAM:



### TRUTH TABLE:

S.No.	Cascade inputs	Present input condition			AGTBOUT	AEQBOUT	ALTBOUT
		A>B	A=B	A<B			
1	AGTBIN=1	X	X	X	1	0	0
2	AEQBIN=1	1	0	0	1	0	0
		0	1	0	0	1	0
		0	0	1	0	0	0
5	ALTBIN=1	X	X	X	0	0	1

### VHDL CODE:

```

library IEEE;
use IEEE.std_logic_1164.all;
entity comp is
  port (
    altbin: in STD_LOGIC;
    aeqbin: in STD_LOGIC;
    agtbin: in STD_LOGIC;
    a: in STD_LOGIC_VECTOR (3 downto 0);
    b: in STD_LOGIC_VECTOR (3 downto 0);
    agtbout: out STD_LOGIC;
    aeqbout: out STD_LOGIC;
    altbout: out STD_LOGIC
  );

```



```
end comp;
```

architecture comp of comp is

```
begin
```

```
process(a,b,agtbout,aeqbout,altbout)
```

```
begin
```

```
    agtbout<='0'; --initializes the outputs to '0'
```

```
    aeqbout<='0';
```

```
    altbout<='0';
```

```
    if aeqbin='1' then
```

```
        if a=b then aeqbout<='1';
```

```
        elsif a>b then agtbout<='1';
```

```
        elsif (a<b) then altbout<='1';
```

```
        end if;
```

```
    elsif (altbin/=agtbout)then
```

```
        agtbout<=agtbout;
```

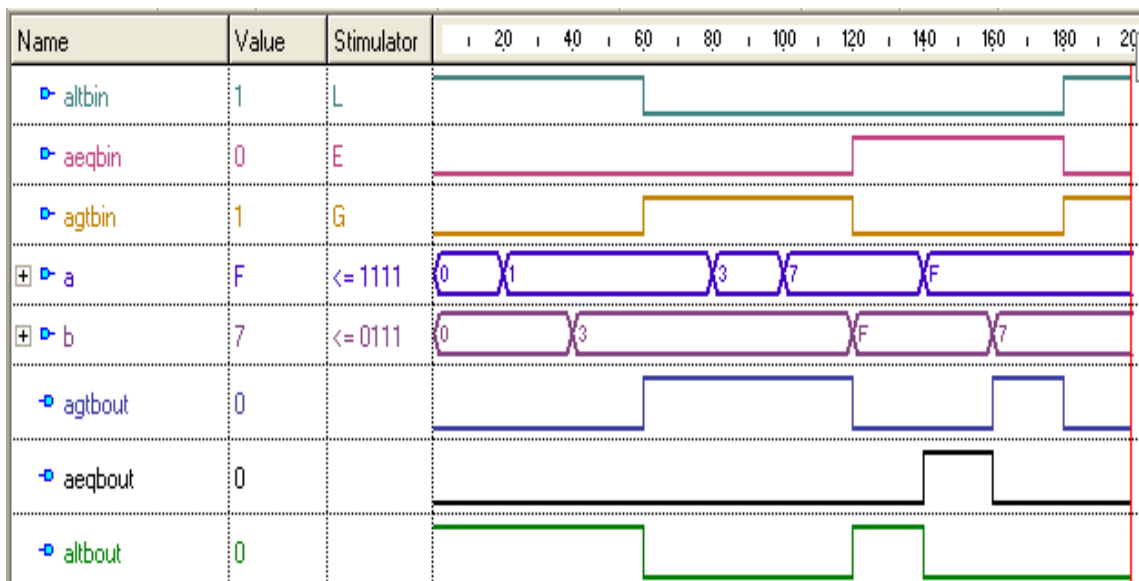
```
        altbout<=altbin;
```

```
    end if;
```

```
end process ;
```

```
end Comp;
```

### WAVEFORMS:



VIVA QUESTIONS:

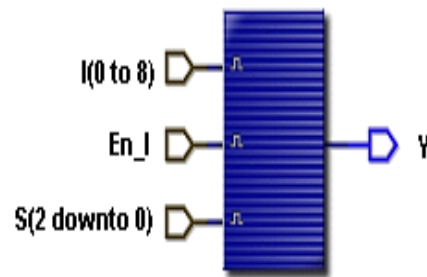
1. Write the dataflow model for the IC 74x85.
2. Write the VHDL code for the IC 74x85 using CASE statement.
3. Write the VHDL code for the IC 74x85 using WITH statement.
4. Write the VHDL code for the IC 74x85 using WHEN--ELSE statement.
5. Write the structural program for IC 74x85.
6. How many 4-bit comparators are needed to construct 12-bit comparator?
7. What does a digital comparator mean?
8. Design a 2-bit comparator using gates?
9. Explain the phases of a simulation?
10. Explain briefly about wait statement?

**9.8x1 MULTIPLEXER**

**AIM:** Write a VHDL code for IC74151—8x1 multiplexer.

**TITLE:** IC74151—8x1 multiplexer.

**BLOCK DIAGRAM:**



**TRUTH TABLE:**

S.No	en_1	Data select lines			Output Y
		A	B	C	
1	0	0	0	0	I(0)
2	0	0	0	1	I(1)
3	0	0	1	0	I(2)
4	0	0	1	1	I(3)
5	0	1	0	0	I(4)
6	0	1	0	1	I(5)
7	0	1	1	0	I(6)
8	0	1	1	1	I(7)
9	1	X	X	X	0

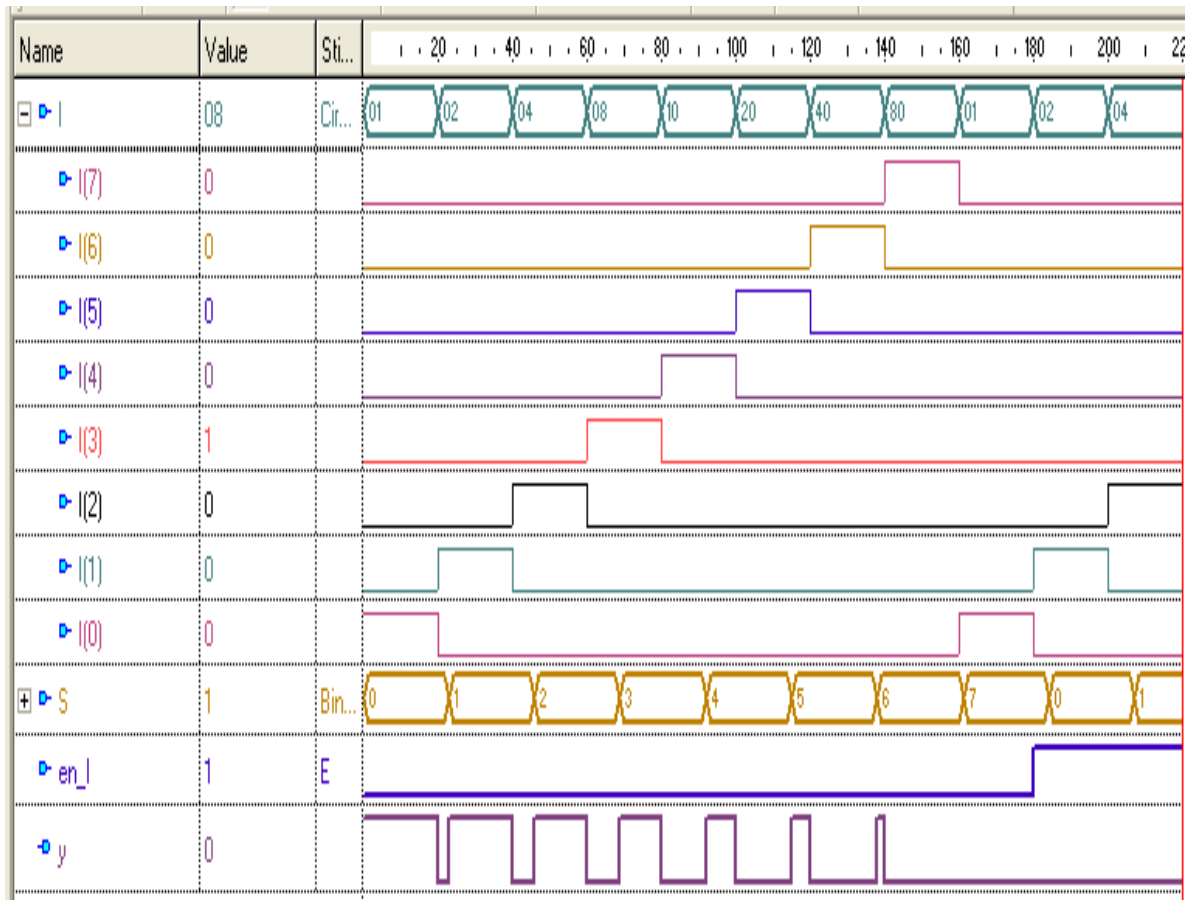
**VHDL CODE:**

```
library IEEE;
use IEEE.std_logic_1164.all;

entity mux151 is
  port (
    I   :in STD_LOGIC_VECTOR (7 downto 0); --8 i/p lines
    S   :in STD_LOGIC_VECTOR (2 downto 0); --3 data select lines
    en_l:in STD_LOGIC;                    --active low enable i/p
    y   :out STD_LOGIC                    --output line
  );
end mux151;

architecture mux151 of mux151 is
begin
  process (I,s,en_l)
  begin
    if en_l='0' then
      case s is
        when "000" => y <= I(0);
        when "001" => y <= I(1);
        when "010" => y <= I(2);
        when "011" => y <= I(3);
        when "100" => y <= I(4);
        when "101" => y <= I(5);
        when "110" => y <= I(6);
        when "111" => y <= I(7);
        when others=>null;
      end case;
    else y <= '0'; --y=0 when en_l=1
    end if;
  end process;
end mux151;
```

## WAVEFORMS:



## VIVA QUESTIONS

:

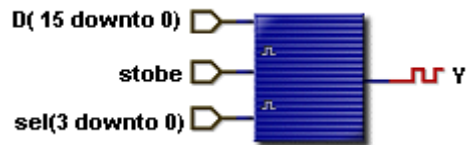
1. Write the behavioral code for the IC 74x151.
2. Write the VHDL code for the IC 74x151 using IF statement.
3. Write the VHDL code for the IC 74x151 using WITH statement.
4. Write the VHDL code for the IC 74x151 using WHEN--ELSE statement.
5. Write the structural program for IC 74x151.
6. What is meant by multiplexer?
7. What does demultiplexer mean?
8. How many 8X1 multiplexers are needed to construct 16X1 multiplexer?
9. Compare decoder with demultiplexer?
10. Design a full adder using 8X1 multiplexer?
11. What are the two kinds of subprograms?
12. What are the difference between function and procedure?
13. Explain briefly about subprogram overloading?

### 10.16X1 MULTIPLEXER

**AIM:** Write a VHDL code for IC74150—16x1 multiplexer.

**TITLE:** IC74150—16x1 multiplexer.

**BLOCK DIAGRAM:**



**TRUTH TABLE:**

S.No.	Data select lines					output
	strobe	A	B	C	D	Y
1	0	0	0	0	0	d'(0)
2	0	0	0	0	1	d'(1)
3	0	0	0	1	0	d'(2)
4	0	0	0	1	1	d'(3)
5	0	0	1	0	0	d'(4)
6	0	0	1	0	1	d'(5)
7	0	0	1	1	0	d'(6)
8	0	0	1	1	1	d'(7)
9	0	1	0	0	0	d'(8)
10	0	1	0	0	1	d'(9)
11	0	1	0	1	0	d'(10)
12	0	1	0	1	1	d'(11)
13	0	1	1	0	0	d'(12)
14	0	1	1	0	1	d'(13)
15	0	1	1	1	0	d'(14)
16	0	1	1	1	1	d'(15)
17	1	X	X	X	X	1

---

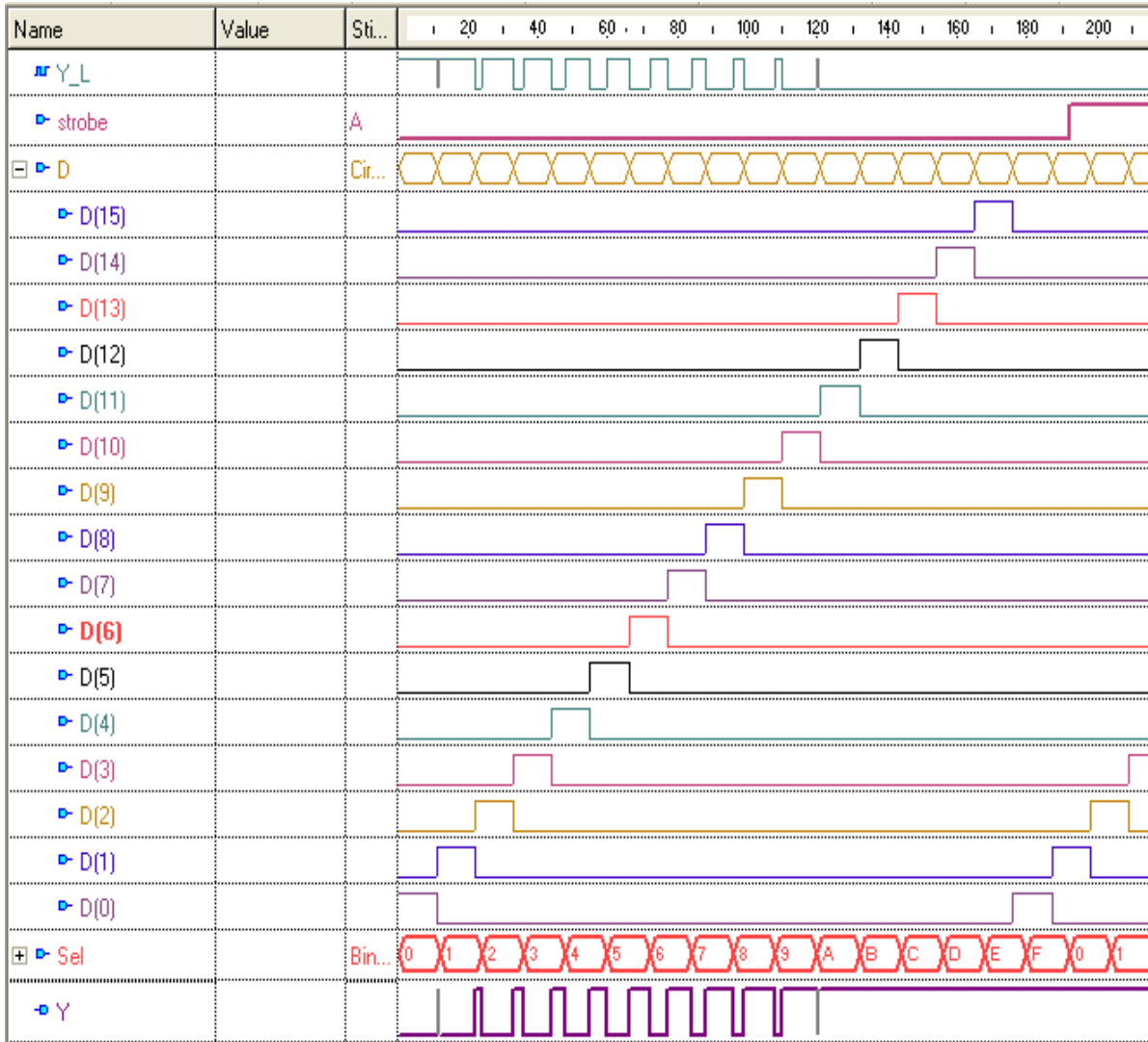
**VHDL CODE:**

```
library IEEE;
use IEEE.std_logic_1164.all;

entity mux16 is
    port(
        strobe : in STD_LOGIC;           --active low enable i/p
        D : in STD_LOGIC_VECTOR(15 downto 0); --16 i/p lines
        Sel : in STD_LOGIC_VECTOR(3 downto 0); --4 data select lines
        Y : out STD_LOGIC                --output line
    );
end mux16;

architecture mux16 of mux16 is
    signal Y_L:std_logic;
begin
    with Sel select
        Y_L <= D(0) when "0000",
              D(1) when "0001",
              D(2) when "0010",
              D(3) when "0011",
              D(4) when "0100",
              D(5) when "0101",
              D(6) when "0110",
              D(7) when "0111",
              D(8) when "1000",
              D(9) when "1001",
              D(10) when "1010",
              D(11) when "1011",
              D(12) when "1100",
              D(13) when "1101",
              D(14) when "1110",
              D(15) when "1111",
              unaffected when others;
        Y <= not Y_L when (strobe='0') else '1';
end mux16;
```

WAVEFORMS:





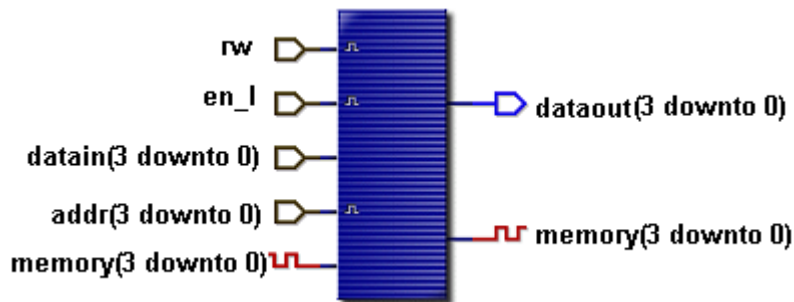
VIVA QUESTIONS:

1. Write the behavioral code for the IC 74x150.
2. Write the VHDL code for the IC 74x150 using IF statement.
3. Write the VHDL code for the IC 74x150 using CASE statement.
4. Write the VHDL code for the IC 74x150 using WHEN--ELSE statement.
5. Write the structural program for IC 74x150.
6. Implement 16X1 multiplexer using structural model?
7. Write the applications of multiplexer and demultiplexer?
8. Design 32X1 multiplexer using 16X1 multiplexer?
9. Explain briefly about operator overloading?
10. Explain the execution steps of subprogram?
11. Write the syntax of package declaration?
12. Write the syntax of package body?

## 11.IC 74X189—READ AND WRITE OPERATIONS OF RAM

**AIM:** To write the VHDL code for IC 74X189—read and write operations of RAM.

**BLOCK DIAGRAM:**



**TRUTH TABLE:**

en_l	rw	operation
0	0	Write
0	1	Read the complemented data
1	X	Inhibit

**VHDL code:**

```

library IEEE;
use IEEE.std_logic_1164.all;
entity ram is
  port (
    rw : in STD_LOGIC; --read or write enable pin
    en_l: in STD_LOGIC; --active low enable pin
    datain: in STD_LOGIC_VECTOR (3 downto 0); --4-bit input data line
    addr: in STD_LOGIC_VECTOR (3 downto 0); --4-bit address line
    dataout: out STD_LOGIC_VECTOR (3 downto 0) --4-bit input data line
  );
end ram;

```

---

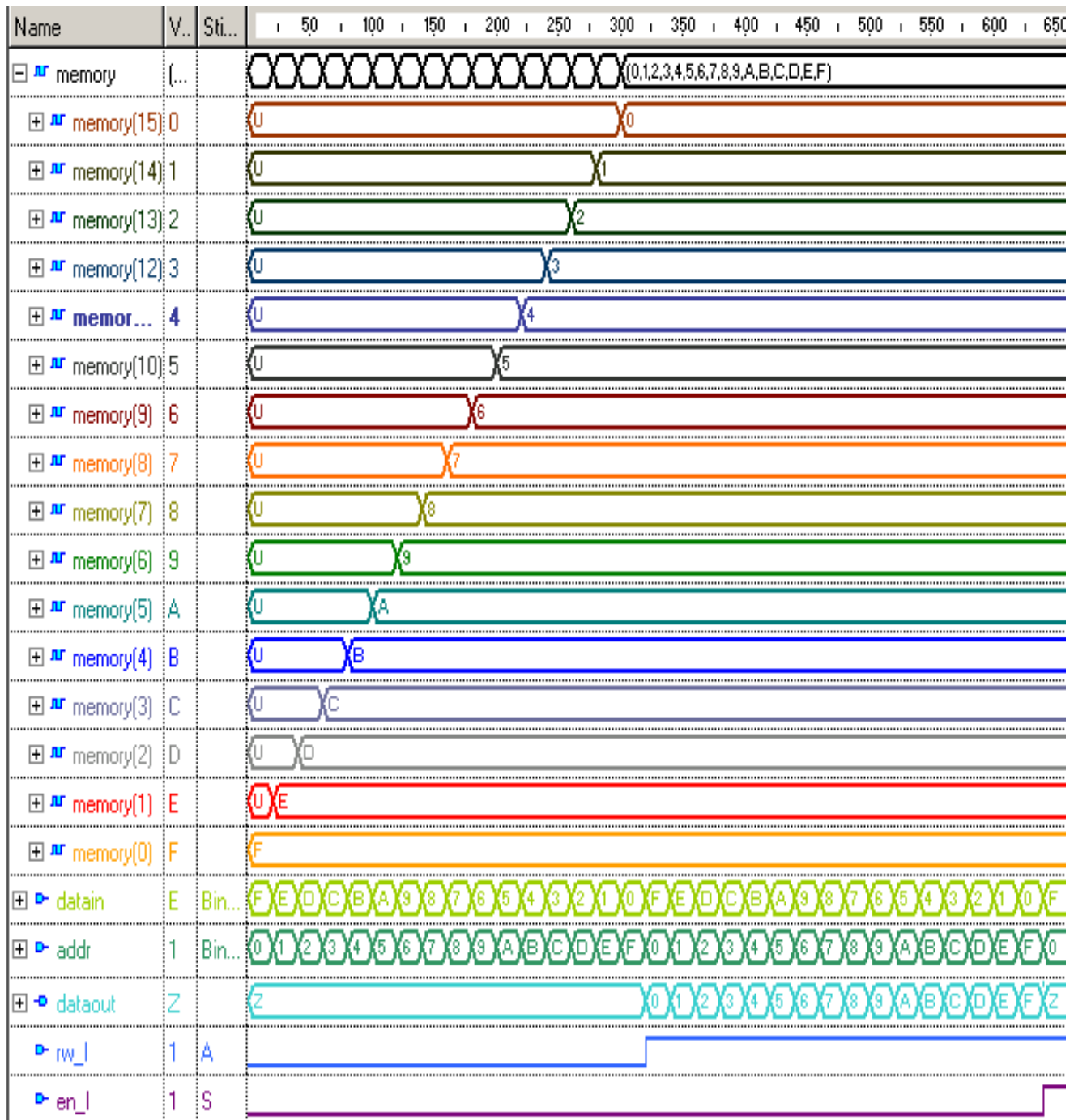
```

architecture ram of ram is
subtype wtype is STD_LOGIC_VECTOR (3 downto 0);
type mem_type is array (15 downto 0) of wtype;
signal memory:mem_type; --creates 16 memory locations.Each location can store
--4-bits
function conv_integer(x:std_logic_vector) return integer is --function to convert variable
--binary to integer
result:integer;
begin
    result:=0;
    for i in x'range loop
        if x(i)='1' then
            result:= result+2**i;
        else null;
        end if;
    end loop;
    return result;
end conv_integer;

begin
process(en_l,rw,addr)
begin
    if(en_l='0') then
        if (rw ='0') then --performs write operation
            memory(conv_integer(addr))<= datain;--stores the data in the
            dataout<="ZZZZ"; -- corresponding memory
        elsif (rw ='1') then -- the output performs read operation
            dataout<=not memory(conv_integer(addr));--places the data on
            -- the given address line
        end if;
    else
        dataout<=(others=>'Z'); --output is in inhibit state when en_l='1'(i.e.Hi-
        -- impedance)
    end if;
end process;
end ram;

```

WAVEFORMS:



VIVA QUESTIONS:

1. Write the behavioral code for IC 74x189 without declaring the function.
2. Explain about different types of RAMs?
3. How to specify the memory size?
4. Explain read and write operations?
5. What are the differences between ROM and RAM?
6. Explain the steps of a compilation process of a VHDL program?
7. Explain the types of design units?
8. Why configurations are needed?
9. What is binding?
10. What is subprogram in vhdl