**COE 405** **Term: 031**

**HW#1 Solution**

**Q1-i**

```
Entity CLA4 is

port ( a , b : IN Bit_Vector(3 downto 0) ; s : OUT Bit_Vector(3 downTo 0) ;
     Cin : in Bit; C3 : out Bit) ;
End;
```

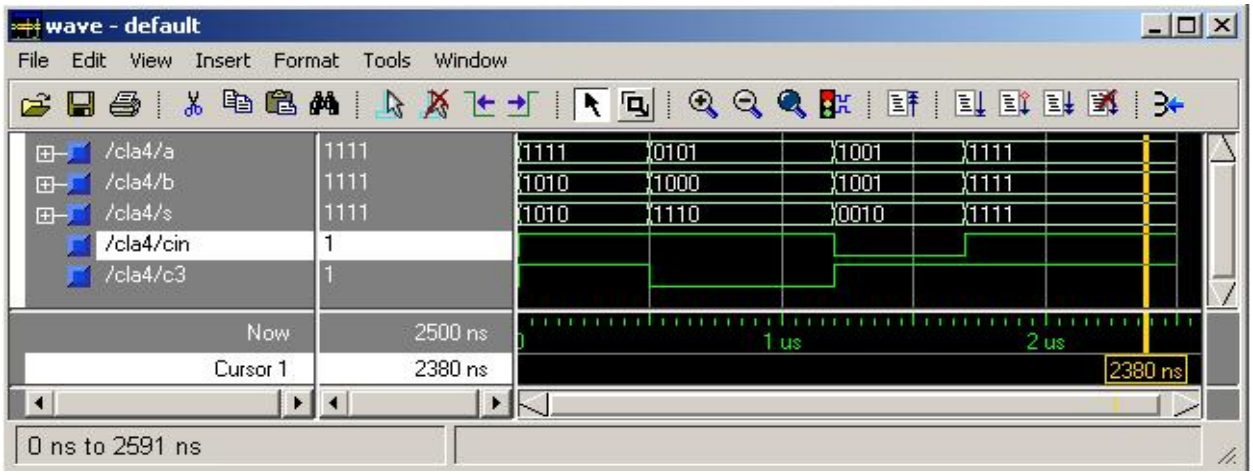
---

**Q1-ii**

```
Architecture Arch1 of CLA4 is
SIGNAL P , G : Bit_Vector(3 downto 0);
SIGNAL  c0 , c1 , c2 : bit;

Begin
 -- Pi = Ai Xor Bi
 -- Gi = Ai And Bi
 -- Si = Pi Xor Ci
 -- C(i+1) = Gi Or (Pi And Ci)
  P(0) <= a(0) Xor b(0);
  G(0) <= a(0) And b(0);

  P(1) <= a(1) Xor b(1);
  G(1) <= a(1) And b(1);

  P(2) <= a(2) Xor b(2);
  G(2) <= a(2) And b(2);

  P(3) <= a(3) Xor b(3);
  G(3) <= a(3) And b(3);

  s(0) <= P(0) Xor Cin ;
  c0   <= G(0) OR (p(0) AND Cin);

  s(1) <= P(1) XOR c0;
  c1   <= G(1) OR (P(1) And G(0))
            OR (P(1) And P(0) And Cin);

  s(2) <= P(2) XOR c1;
  c2   <= G(2) OR (P(2) AND G(1))
            OR (P(2) AND P(1) AND G(0))
            OR (P(2) And  P(1) And P(0) And Cin);
```

```
s(3) <= P(3) XOR c2;
c3   <= G(3) OR (P(3) And G(2))
        OR (P(3) And P(2) And G(1))
        OR (P(3) AND P(2) And P(1) And G(0))
        OR (P(3) AND P(2) And P(1) AND P(0) And Cin);
End Arch1 ;
```

**Q1-iii**




**Q1-iv**

```
-------------------------------------------------------------------
 -------------------------------------------------------------------
 -- Define the entities needed in gate level architecture
 -------------------------------------------------------------------
 Entity AND2 is
 PORT(a , b : IN bit ; c : OUT bit);
 End ;

 Architecture and2Arch of AND2 is
 Begin
 c <= a AND b after 2 ns;
 End and2Arch;
 -------------------------------------------------------------------
 -------------------------------------------------------------------
 Entity OR2 is
 PORT(a , b : IN bit ; c : OUT bit);
 End ;

 Architecture OR2Arch of OR2 is
 Begin
 c <= a OR b after 2 ns;
 End OR2Arch;
 -------------------------------------------------------------------
```

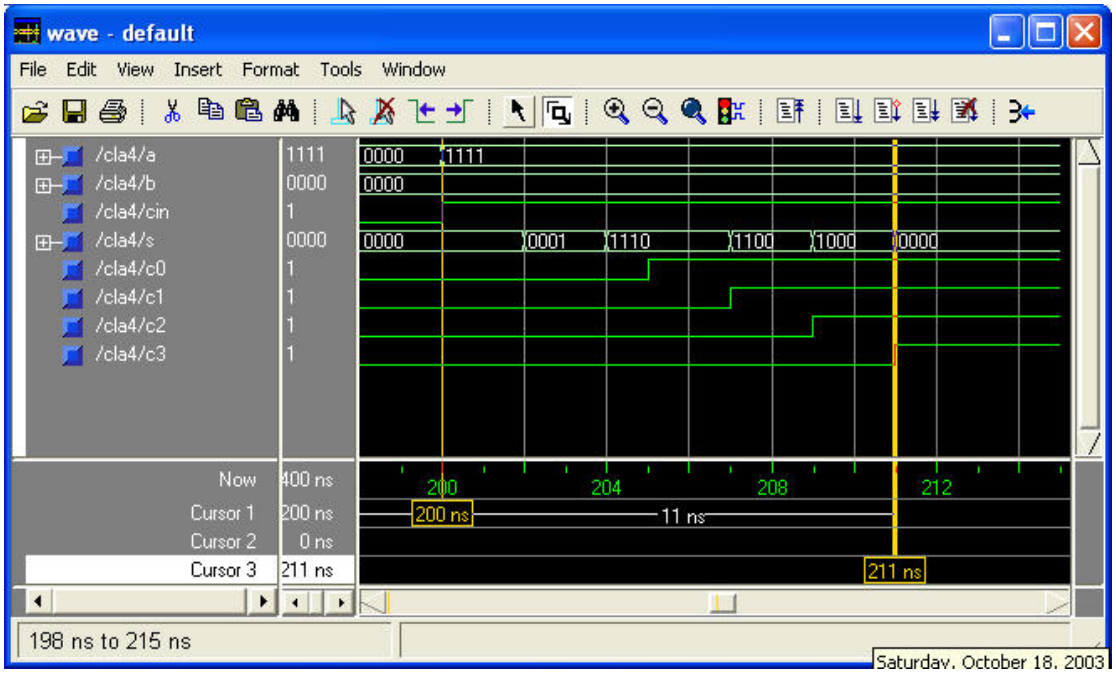
```
--------------------------------------------------------------------------
Entity XOR2 is
PORT(a , b : IN bit ; c : OUT bit);
End ;

Architecture XOR2Arch of XOR2 is
Begin
c <= a XOR b after 2 ns;
End XOR2Arch;
--------------------------------------------------------------------------
--------------------------------------------------------------------------

Entity C0Eval is
PORT(a , b , c : IN bit ; d : OUT bit);
End ;
----- c0  <= G(0) OR (p(0) AND Cin);
Architecture C0Arch of C0Eval is
Begin
d <= a OR (b And c) after 3 ns; -- since we have 3 inputs
End C0Arch;

--------------------------------------------------------------------------
--------------------------------------------------------------------------

Entity C1Eval is
PORT(a , b , c , d , e : IN bit ; f : OUT bit);
End ;

----- c1   <= G(1) OR (P(1) And G(0))
-----              OR (P(1) And P(0) And Cin);
Architecture C1Arch of C1Eval is
Begin
f <=  a OR (b And c) OR (b AND d And e) after 5 ns;

 End C1Arch;
--------------------------------------------------------------------------
--------------------------------------------------------------------------

Entity C2Eval is
PORT(a, b , c ,d, e, f , g : IN bit ; h : OUT bit);
End ;

----- c2 <= G(2) OR (P(2) AND G(1))
-----            OR (P(2) AND P(1) AND G(0))
-----            OR (P(2) And  P(1) And P(0) And Cin);

Architecture C2Arch of C2Eval is
Begin

h      <=  a   OR (b AND c)
```

```
            OR (b AND d AND e)
            OR (b AND d AND f And g) after 7 ns; -- since we have 7 inputs
End C2Arch;
--------------------------------------------------------------------
--------------------------------------------------------------------

Entity C3Eval is
PORT(a, b , c ,d, e, f , g , h , i : IN bit ; j : OUT bit);
End ;

----  c3   <= G(3) OR (P(3) And G(2))
----              OR (P(3) And P(2) And G(1))
----              OR (P(3) AND P(2) And P(1) And G(0))
----              OR (P(3) AND P(2) And P(1) AND P(0) And Cin);


Architecture C3Arch of C3Eval is
Begin

j    <=   a  OR  (b And c)
            OR (b And d And e)
            OR (b AND d And f And g)
            OR (b AND d And f AND h And i) after 9 ns;
End C3Arch;
--------------------------------------------------------------------
--------------------------------------------------------------------

Architecture Arch2 of CLA4 is
SIGNAL P , G : Bit_Vector(3 downto 0);
SIGNAL  c0 , c1 , c2 : bit;

component AND2
port(a,b:in bit; c:out bit);
end component ;

component OR2
port(a,b:in bit; c:out bit);
end component ;

component XOR2
port(a,b:in bit; c:out bit);
end component ;

component c0Eval
port(a,b,c:in bit; d:out bit);
end component ;

component c1Eval
port(a,b,c,d,e:in bit; f:out bit);
end component ;
```

```
component c2Eval
port(a,b,c,d,e,f,g:in bit; h:out bit);
end component ;

component c3Eval
port(a,b,c,d,e,f,g,h,i:in bit; j:out bit);
end component ;


Begin
g0   : AND2 PORT MAP (a(0), b(0), G(0) );
g1   : AND2 PORT MAP (a(1), b(1), G(1) );
g2   : AND2 PORT MAP (a(2), b(2), G(2) );
g3   : AND2 PORT MAP (a(3), b(3), G(3) );

p0   : XOR2 PORT MAP (a(0), b(0), P(0) );
p1   : XOR2 PORT MAP (a(1), b(1), P(1) );
p2   : XOR2 PORT MAP (a(2), b(2), P(2) );
p3   : XOR2 PORT MAP (a(3), b(3), P(3) );


sum0 : XOR2    PORT MAP (P(0) , Cin  , s(0));
car0 : C0Eval  PORT MAP (G(0) , P(0) , Cin , c0 );

sum1 : XOR2    PORT MAP (P(1)  , c0 , s(1));
car1 : c1Eval  PORT MAP (a => G(1) ,b=> P(1) ,c=> G(0) ,d=> P(0) ,e=> Cin , f=> c1);


sum2 : XOR2    PORT MAP (P(2) , c1 , s(2));
car2 : c2Eval  PORT MAP (G(2) , P(2) ,G(1),P(1),G(0),P(0),Cin, c2);


sum3 : XOR2    PORT MAP (P(3) , c2 , s(3));
car3 : c3Eval  PORT MAP (G(3), P(3) , G(2) ,P(2), G(1), P(1) , G(0) , P(0) , Cin , c3);

End Arch2;
```

**Q1-v**

Critical path highlighted when carry-in is propagated through all stages to C3. The critical path delay is 11ns.




**Q2-i**

Entity FAR is
port ( d,clk,reset : in BIT ; q : out BIT);
End FAR;

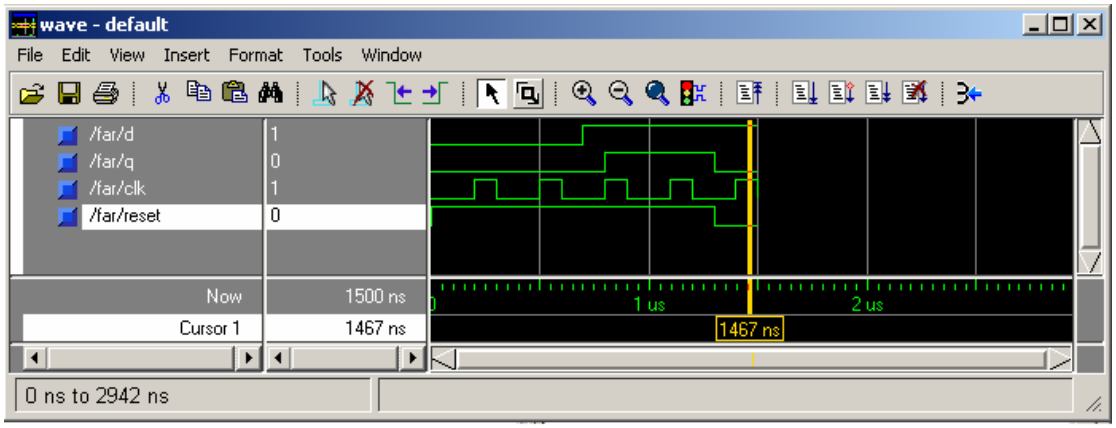
**Q2-ii**

Architecture Arch1 of FAR is
Begin

```
  process (clk , reset)
    Begin
     if reset = '0' then
     q <= '0' ;
     else
        if (clk= '1' and clk'Event) then
        q <= d ;
        end if;
  end if ;
    End process;
End Arch1;
```

**Q2-iii**

The following figure shows the operation of D-flip flop with a synchronous rising edge clock and asynchronous reset .




**Q2-iv**

Entity Reg4 is
Port(i:In Bit_Vector(3 downto 0);clock,res:IN BIT;o:OUT bit_Vector(3 downto 0));
End Reg4;

**Q2-v**

Architecture ArchReg1 of Reg4 is

component FAR
port (d,clk,reset : in bit ; q : OUT bit);
End component;

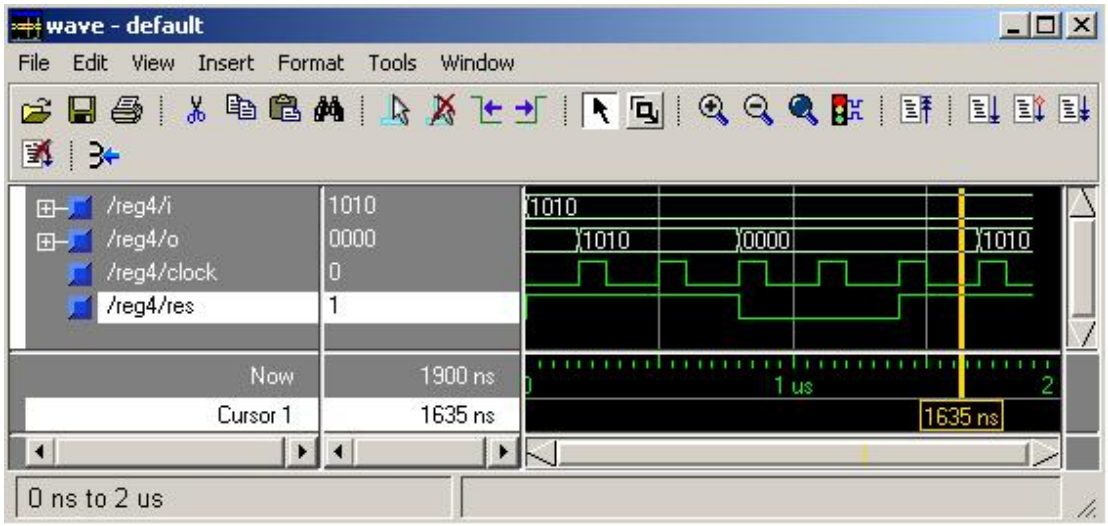
begin

bit0 : FAR PORT MAP (i(0),clock,res,o(0));
bit1 : FAR PORT MAP (i(1),clock,res,o(1));
bit2 : FAR PORT MAP (i(2),clock,res,o(2));
bit3 : FAR PORT MAP (i(3),clock,res,o(3));

End ArchReg1;

**Q2-vi**

The following figure shows the operation of the Register that has 4 D-flip flops.




**Q3-i**

```
Entity ACC Is
port( reset , clk : in BIT ; a  : inout   BIT_vector(3 downto 0);
                             b  : in   BIT_vector(3 downto 0));


End ACC;
```

**Q3-ii**

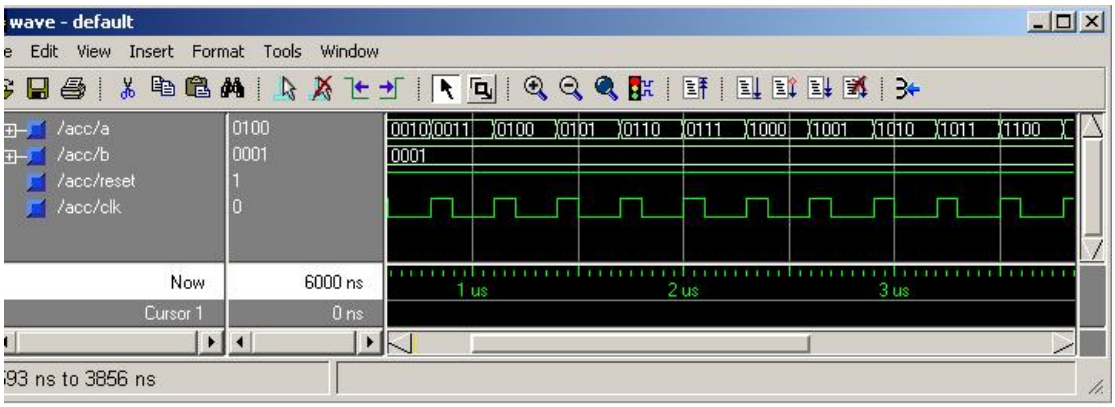
```
Architecture Arch1 of ACC is


signal sum : Bit_Vector(3 downto 0);
signal out_of_reg :  bit_vector(3 downto 0);
signal ci : bit:= '0';
signal c3 : bit;


component Reg4
Port(i:In Bit_Vector(3 downto 0);clock,res:IN BIT;o:OUT bit_Vector(3 downto 0));
End component ;

component CLA4
port ( a , b : IN Bit_Vector(3 downto 0) ; s : OUT Bit_Vector(3 downTo 0) ;
     Cin : in Bit; C3 : out Bit) ;
End component ;

Begin

Addi  :  CLA4  PORT MAP(a,b,sum,ci,c3);
rege  :  Reg4  PORT MAP(sum,clk,reset,out_of_reg);
```

process (out_of_reg)
Begin

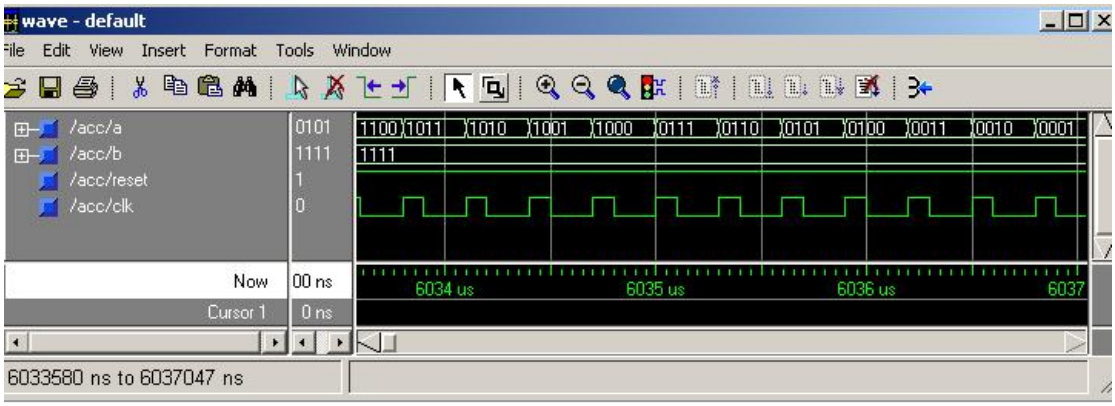
a <= out_of_reg;

end process ;
End Arch1;

**Q3-iii**




up counter




down counter