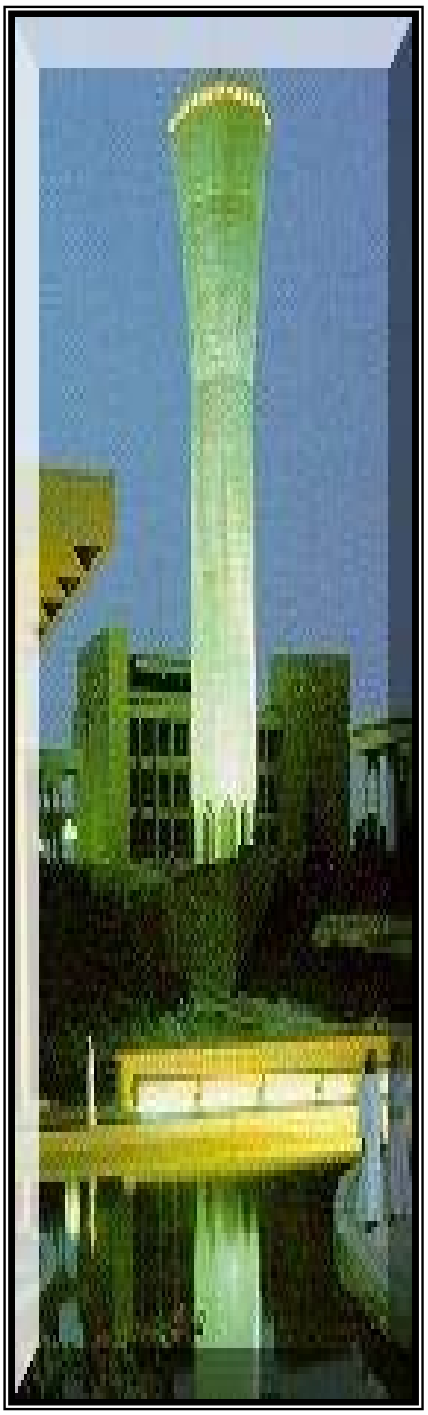


COE 405

VHDL

By Dr.

Aiman El-Maleh

Rayan Jamal Mufti
985023
Burhan Ali
987528

Assignment: 1

Q.1.
(i)

The intity description of the four bit carry look ahead adder is as follow:

```
--------------------------------------------------------------------------------
--  I N T I T Y   D E S C R I P T I O N  O F  C L A
--------------------------------------------------------------------------------
--
ENTITY CLA is
PORT (      A,B:   in    Bit_Vector (3 DOWNTO 0);      -- A & B ==> four bit inputs each.
            C0:    in    Bit;                          -- C0 ==> Carry in (C zero).
            S:     out   Bit_Vector (3 DOWNTO 0);      -- S  ==> four bit output.
            C4:    out   Bit);                         -- C4 ==> Carry out.
END CLA;
--
```

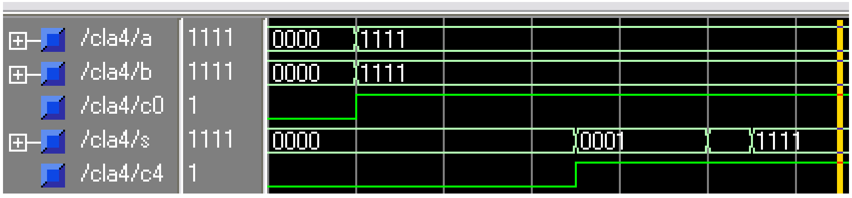
Q.1.
(ii)

```
--------------------------------------------------------------------------------
--  A R C H I T E C T U R E  D E S C R I P T I O N  OF  C L A
--------------------------------------------------------------------------------
--
ARCHITECTURE      Arch1 of CLA4 is
SIGNAL C1,C2,C3: Bit;
SIGNAL P,G: Bit_Vector (3 DOWNTO 0);
BEGIN
      P(0)  <= A(0)XOR B(0);              -- The propagate function of A & B.
      G(0)  <= A(0)AND B(0);              -- The generate function of A & B.
      S(0)  <= P(0)XOR C0 after 5 ns;     -- The sum of A & B.
      C1    <= G(0)OR                     -- The carry is generated after 2 gates only.
               (P(0)AND C0) after 3 ns;
--------------------------------------------------------------------------------
      P(1)  <= A(1)XOR B(1);              -- The propagate function of A & B.
      G(1)  <= A(1)AND B(1);              -- The generate function of A & B.
      S(1)  <= P(1)XOR C1 after 5 ns;     -- The sum of A & B.
      C2    <= G(1)OR(P(1)AND G(0))OR     -- The carry is generated after 2 gates only.
              (P(1)AND P(0)AND C0) after 3 ns;
--------------------------------------------------------------------------------
      P(2)  <= A(2)XOR B(2);              -- The propagate function of A & B.
      G(2)  <= A(2)AND B(2);              -- The generate function of A & B.
      S(2)  <= P(2)XOR C2 after 5 ns;     -- The sum of A & B.
      C3    <= G(2)OR (P(2)AND G(1))OR
              (P(2)AND P(1)AND G(0))OR    -- The carry is generated after 2 gates only.
              (P(2)AND P(1)AND P(0)AND C0) after 4 ns;
--------------------------------------------------------------------------------
      P(3)  <= A(3)XOR B(3);              -- The propagate function of A & B.
      G(3)  <= A(3)AND B(3);              -- The generate function of A & B.
      S(3)  <= P(3)XOR C3 after 5 ns;     -- The sum of A & B.
      C4    <= G(3)OR (P(3)AND G(2))OR
              (P(3)AND P(2)AND G(1))OR    -- The carry is generated after 2 gates only.
              (P(3)AND P(2)AND P(1)AND G(0))OR
              (P(3)AND P(2)AND P(1)AND P(0)AND C0) after 5 ns;
END Arch1;
```

Q.1.
(iii)

The simulation is working properly.

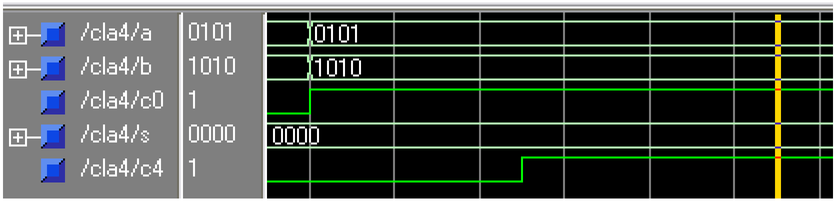
```
Example:      A: 1111
              B: 1111 +
              C0:    1 +
              -------
              S: 1111
              C4     1
```




```
>force A 1111 2ns
>force B 1111 2ns
>force C0   1 2ns
>run 15
```

- -

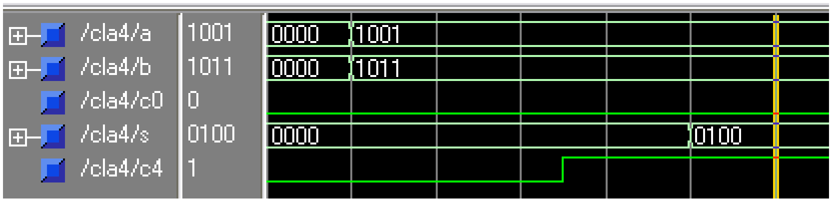
```
Example:      A: 0101
              B: 1010 +
              C0:    1 +
              -------
              S: 0000
              C4     1
```




```
>force A 0101 2ns
>force B 1010 2ns
>force C0   1 2ns
>run 15
```

- -

```
Example:      A: 1001
              B: 1011 +
              C0:    0 +
              -------
              S: 0100
              C4     1
```

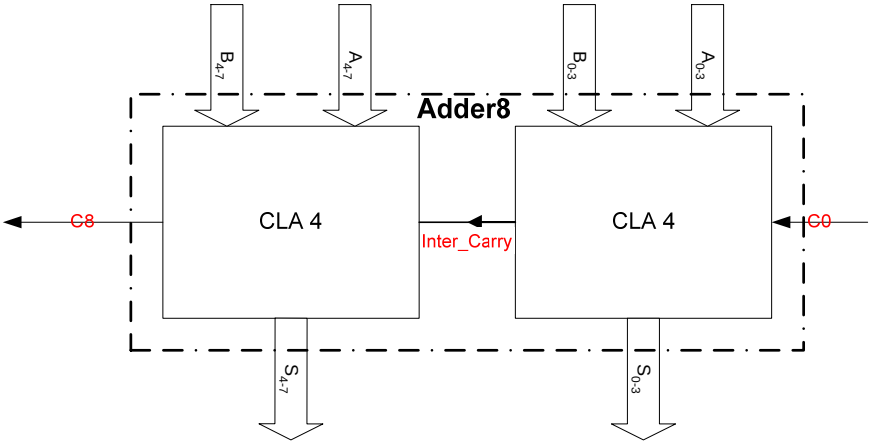



```
>force A 1001 2ns
>force B 1011 2ns
>force C0   0
>run 15
```

- -

Q.2.
(i)

The Adder8 uses 2 instants of the four bits CLA. The carry goes out from CLA1 as C4 and connects to CLA2 C4 via an internal signal called “Inter_Carry”.




```
--------------------------------------------------------------------------------
--  I N T I T Y   D E S C R I P T I O N  O F  A d d e r 8
--------------------------------------------------------------------------------
--
ENTITY Adder8 IS
PORT (
      SIGNAL X,Y : IN  bit_vector (7 DOWNTO 0);       -- X & Y ==> eight bit inputs each.
      SIGNAL Z0  : IN  bit ;                          -- Carry in
      SIGNAL P   : OUT bit_vector (7 DOWNTO 0) ;      -- P ==> eight bit inputs each.
      SIGNAL R8  : OUT bit ) ;                        -- Carry out
END Adder8 ;
--
```

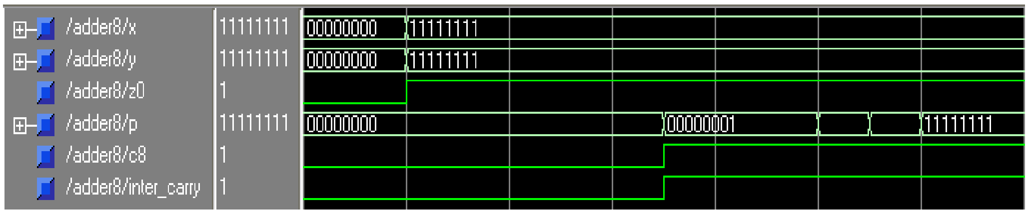
Q.2.
(ii)

```
--
--------------------------------------------------------------------------------
--  A R C H I T E C T U R E  D E S C R I P T I O N  O F  A d d e r 8
--------------------------------------------------------------------------------
--
ARCHITECTURE Struct OF Adder8 IS
      SIGNAL Inter_Carry : bit;
      COMPONENT CLA4
      PORT  (
            A,B:  in    Bit_Vector (3 downto 0);
            C0:   in    Bit;
            S:    out   Bit_Vector (3 downto 0);
            C4:   out   Bit
            );
      END COMPONENT;
BEGIN
      -- Instantiation:

      CLA1: CLA4 PORT MAP (   X(3 DOWNTO 0)                 --X(3),X(2),X(1),X(0)
                              Y(3 DOWNTO 0),                --Y(3),Y(2),Y(1),Y(0)
                              Z0,
                              P(3 DOWNTO 0),                --P(3),(2),P(1),P(0)
                              Inter_Carry
                          );

      CLA2: CLA4 PORT MAP (   X(7 DOWNTO 4),                --X(7),X(6),X(5),X(4)
                              Y(7 DOWNTO 4),                --Y(7),Y(6),Y(5),Y(4)
                              Inter_Carry,
                              P(7 DOWNTO 4),                --P(7),P(6),P(5),P(4)
                              C8
                          );
END Struct;
```

Q.2.
(iii)

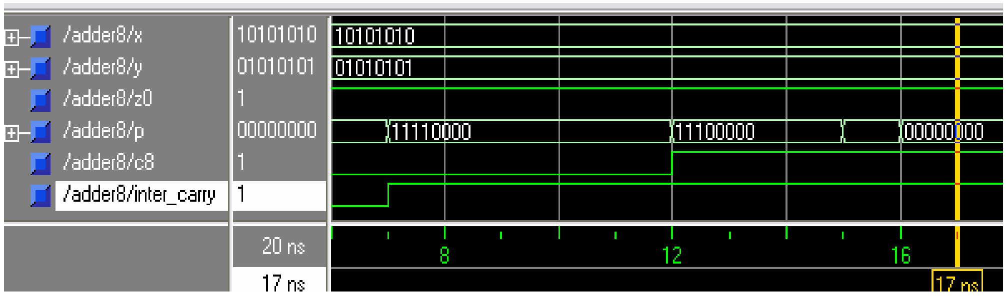
The simulation is working properly.

```
Example:     X: 1111 1111
             Y: 1111 1111 +
             Z0:        1 +
             ------------
             P: 1111 1111
             C8         1

>force X 11111111 2ns
>force Y 11111111 2ns
>force Z0        1 2ns
>run 20
```



- -

```
Example:     X: 1010 1010
             Y: 0101 0101 +
             Z0:        1 +
             ------------
             P: 0000 0000
             C8         1

>force X 10101010 2ns
>force Y 01010101 2ns
>force Z0        1 2ns
>run 20
```



- -

```
Example:     X: 1110 0111
             Y: 1001 1100 +
             Z0:        0 +
             ------------
             P: 1000 0011
             C8         1

>force X 11100111 2ns
>force Y 10011100 2ns
>force Z0        1 2ns
>run 20
```