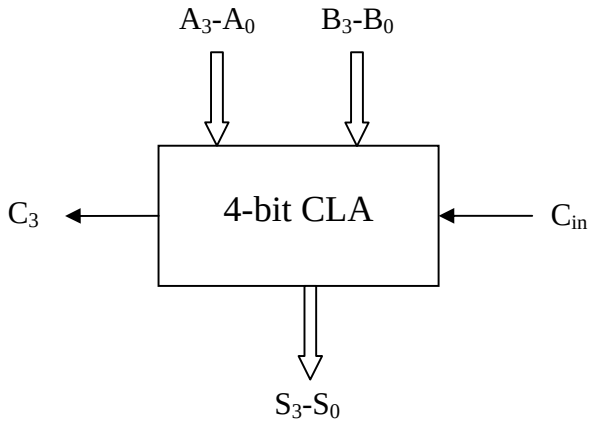
COE 405, Term 062

Design & Modeling of Digital Systems

HW# 1 Solution

Due date: Wednesday, March. 14

Q.1. Consider the 4-bit carry-look-ahead adder (CLA) block shown below:



(i) Describe an Entity **CLA4** in VHDL for this 4-bit CLA using exactly the same port names shown above.

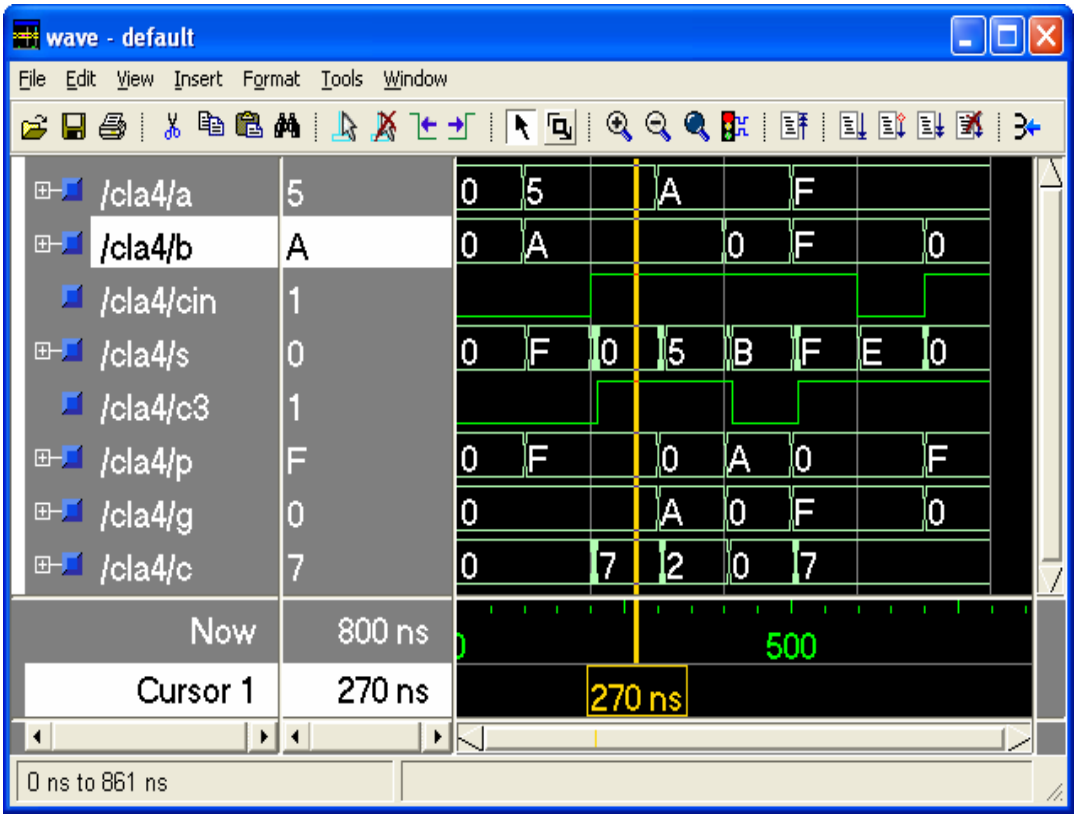
```
Entity CLA4 is
   PORt ( A, B: IN Bit_Vector(3 Downto 0);
          Cin: IN Bit;
          S: OUT Bit_Vector(3 Downto 0);
          C3: OUT Bit
         );
END ;
```

(ii) Write in VHDL an Architecture **Conc** for this 4-bit CLA using concurrent statements.

```
Architecture Conc of CLA4 is

   Signal P, G : Bit_Vector(3 Downto 0);
   Signal C: Bit_Vector(2 Downto 0);
```

```
Begin
    G <= A AND B after 2 ns;
    P <= A XOR B after 2 ns;

    S(0) <= P(0) XOR Cin after 2 ns;
    S(1) <= P(1) XOR C(0) after 2 ns;
    S(2) <= P(2) XOR C(1) after 2 ns;
    S(3) <= P(3) XOR C(2) after 2 ns;

    C(0) <= G(0) OR (P(0) AND Cin) after 4 ns;
    C(1) <= G(1) OR (P(1) AND G(0)) OR (P(1) AND P(0) AND Cin) after 6 ns;
    C(2) <= 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) after 8 ns;
    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) after 10 ns;

end;
```

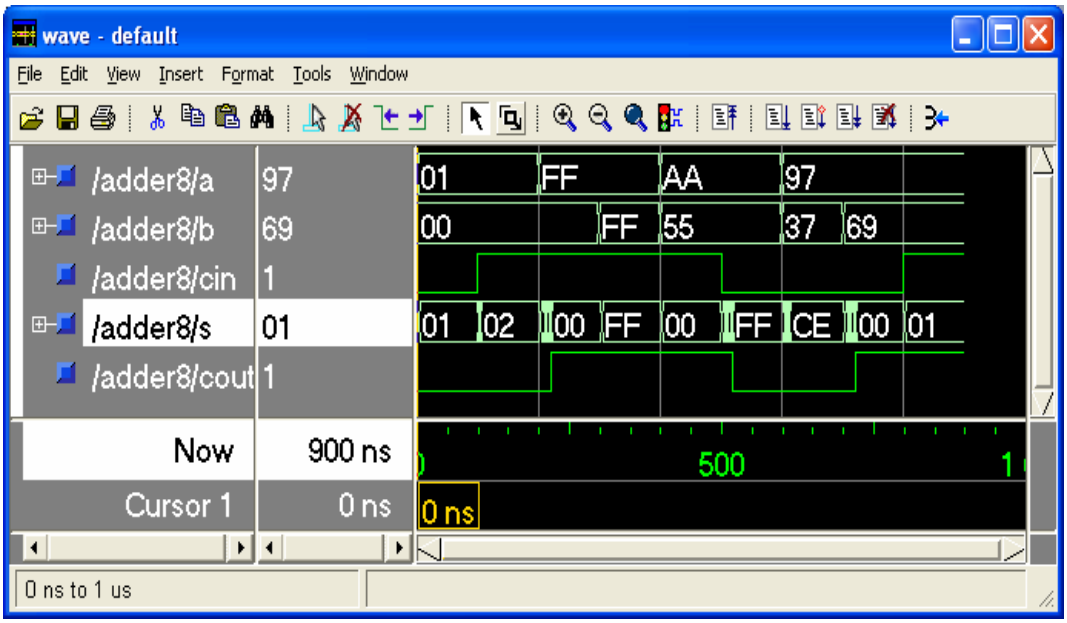
(iii) Simulate the VHDL model **Conc** and verify that it is working properly. Include a snapshot of the simulated waveform.



(iv) Model an 8-bit adder, **Adder8**, based on cascading two CLA4 components by connecting the carryout of the first component to the carry-in of the second component.

```
Entity Adder8 is
    PORT ( A, B: IN Bit_Vector(7 Downto 0);
           Cin: IN Bit;
           S : OUT Bit_Vector(7 Downto 0);
           Cout: OUT Bit
          );
End;

Architecture Structural of Adder8 is
Component CLA4
    PORt ( A, B: IN Bit_Vector(3 Downto 0);
                Cin: IN Bit;
                S: OUT Bit_Vector(3 Downto 0);
                C3: OUT Bit
               );
END Component;
Signal C3: Bit;
Begin
    Add1: CLA4 PORT MAP (A(3 Downto 0), B(3 Downto 0), Cin, S(3 Downto 0),
    C3);
    Add2: CLA4 PORT MAP (A(7 Downto 4), B(7 Downto 4), C3, S(7 Downto 4),
    Cout);
End;
```

(v) Simulate the VHDL model for the 8-bit adder and verify that it is working properly. Include a snapshot of the simulated waveform.



Q.2. It is required to model a 4-bit up-down counter with a synchronous rising-edge clock, and asynchronous reset. Assume that the counter has an input DIR when set to 0, it counts up otherwise it counts down.

(i) Describe an Entity **UDC4** in VHDL for the 4-bit up-down counter.

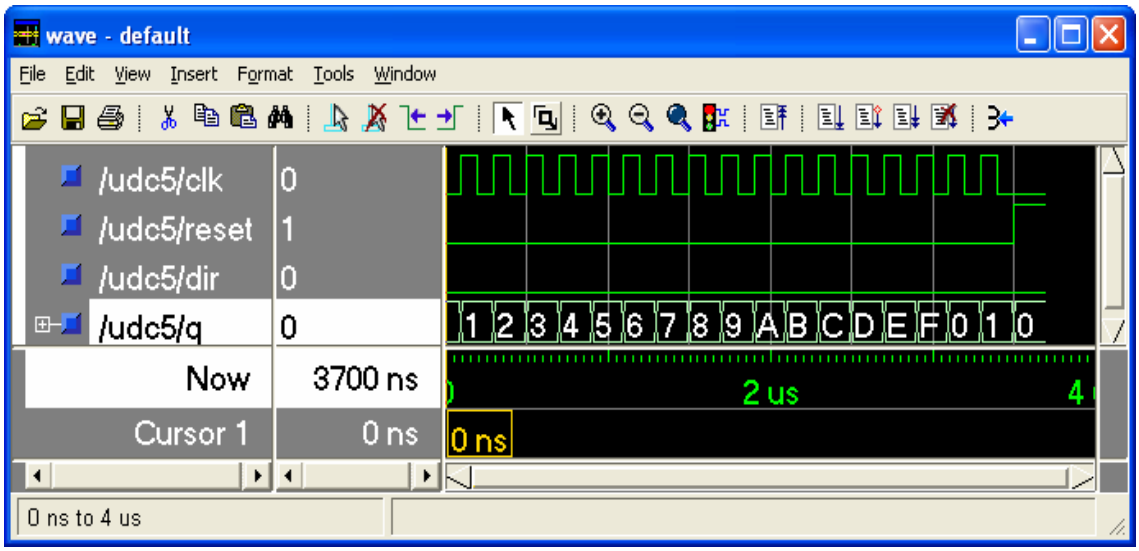
```
Entity UDC5 is
    PORT( CLK, RESET, DIR : IN Bit;
          Q: OUT Bit_Vector(3 Downto 0)
        );
End;
```

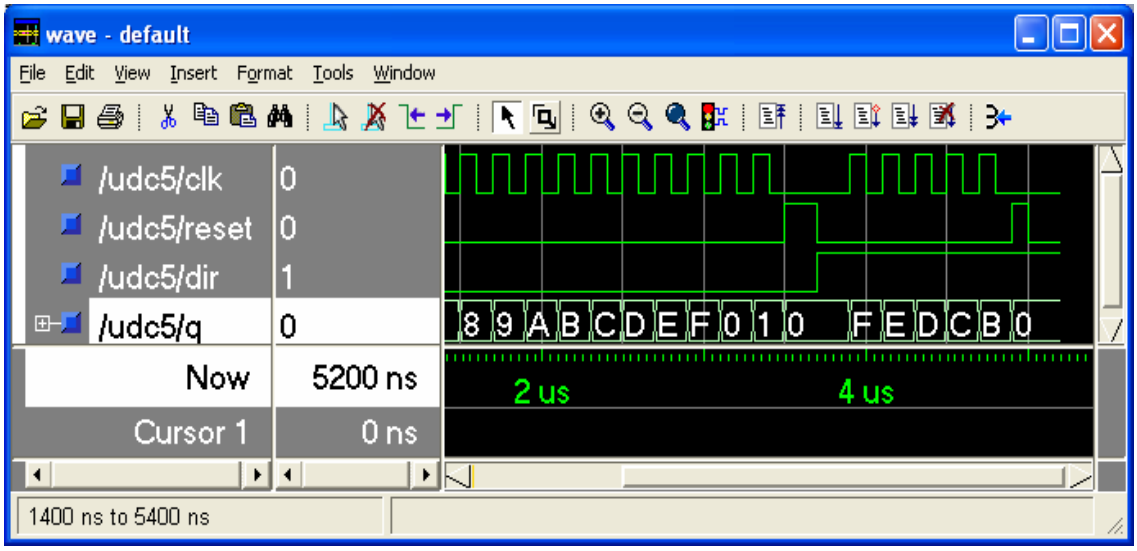
(ii) Model Architecture **Behavior** for the 4-bit up-down counter using a process.

```
Architecture Behavior of UDC5 is
Begin
    Process(RESET, CLK)
            Variable QT: Integer :=0;
    Begin
            If (RESET='1') Then
                    QT := 0;
            Elsif (CLK'Event AND CLK='1') Then
                If (DIR='0') Then
                  QT := QT + 1;
                  If (QT > 15) Then QT := 0; End if;
                Else
                  QT := QT - 1;
                  If (QT < 0) Then QT := 15; ENd if;
                End if;
            End if;

            Case QT is
              When 0 => Q <= "0000";
              When 1 => Q <= "0001";
              When 2 => Q <= "0010";
              When 3 => Q <= "0011";
              When 4 => Q <= "0100";
              When 5 => Q <= "0101";
              When 6 => Q <= "0110";
              When 7 => Q <= "0111";
              When 8 => Q <= "1000";
              When 9 => Q <= "1001";
              When 10 => Q <= "1010";
              When 11 => Q <= "1011";
              When 12 => Q <= "1100";
              When 13 => Q <= "1101";
              When 14 => Q <= "1110";
              When 15 => Q <= "1111";
              When Others => Q <= "0000";
```

```
        End Case;

    End Process;
End;
```

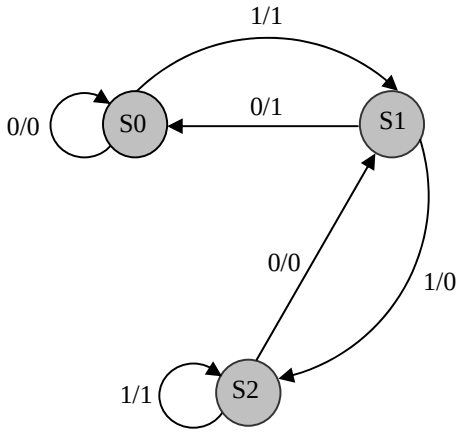
(iii) Simulate the VHDL model **Behavior** and verify that it is working properly. Include a snapshot of the simulated waveform.





Q.3. It is required to design a sequential circuit that receives a serial input X, and produces a serial output Z, equivalent to 3*X, i.e., Z=3*X. The input coming to the X input is transmitted from the least significant bit to the most significant bit. The number of bits transmitted will be equal to the number of bits needed in the output Z. For example, for sending the decimal value 3, the input sequence 0011 will be transmitted and the output sequence 1001 will be produced.

(i) Show the state diagram of this circuit assuming **Mealy** model.

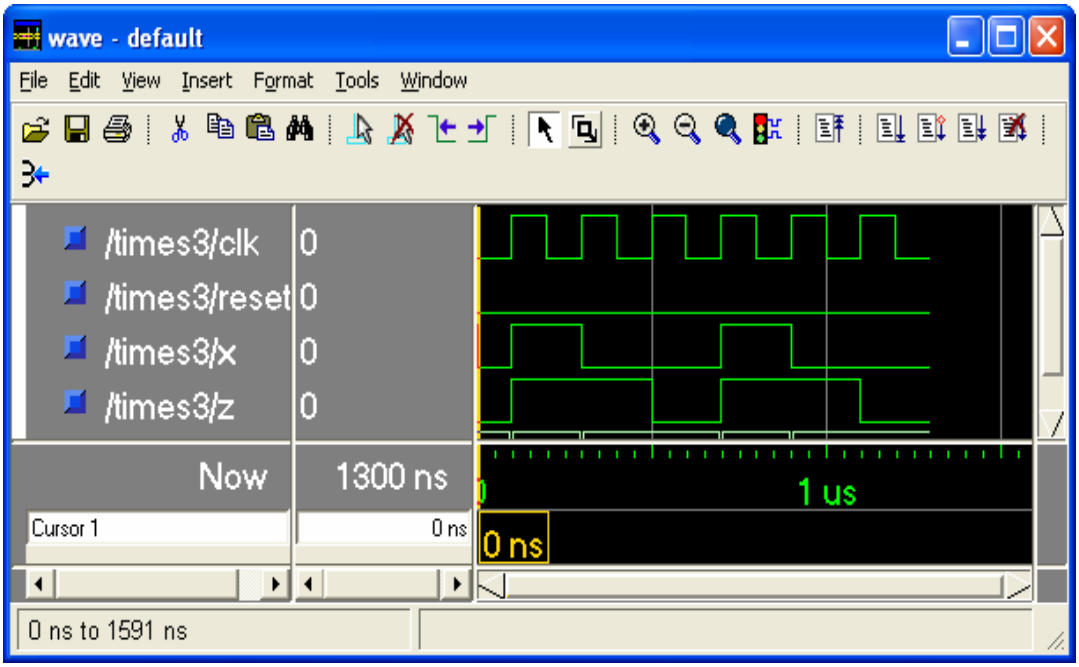


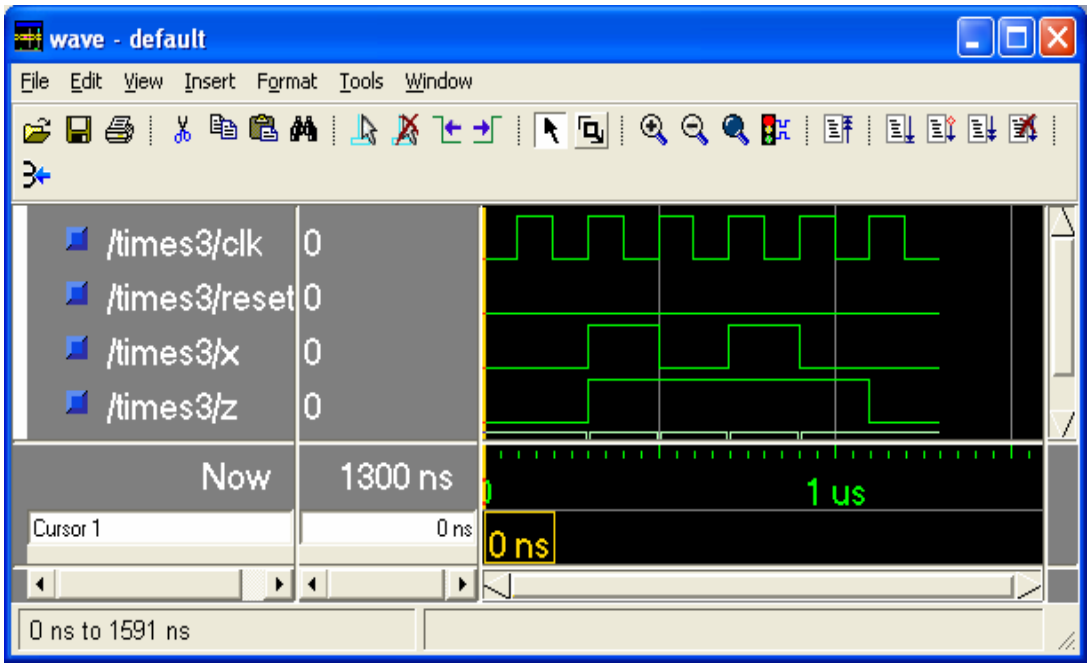
(ii) Model an Entity **Times3** and an Architecture **Behavior** for this circuit using a process. Assume that the circuit will have a synchronous reset and is rising-edge triggered.

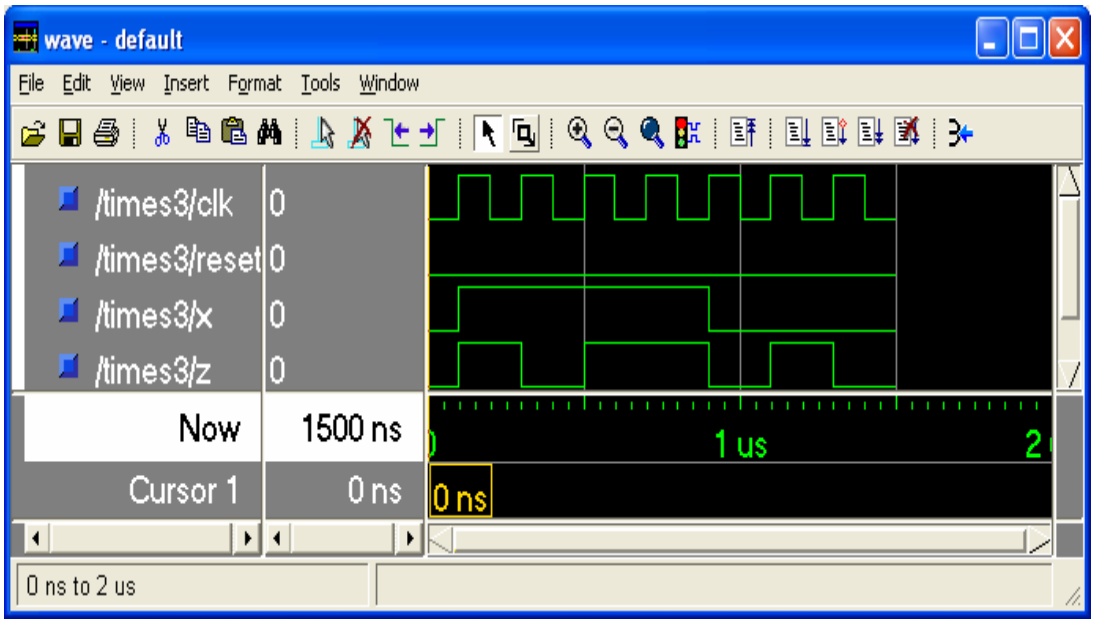
```
Entity Times3 is
   PORT ( CLK, RESET, X: IN BIT;
        Z: OUT BIT
       );
End;
Architecture Behavior of Times3 is
TYPE state IS (s0, s1, s2);
SIGNAL current : state := s0;
BEGIN
   PROCESS(CLK)
   BEGIN


     IF (CLK = '1' AND CLK'Event) THEN
          IF (RESET = '1') Then
                current <= s0;
          ELSE
            CASE current IS
```

```
            WHEN s0 =>          IF x = '1' THEN current <= s1; Z<='1';
                                ELSE current <= s0; Z<='0'; END IF;
            WHEN s1 =>          IF x = '1' THEN current <= s2; Z<='0';
                                ELSE current <= s0; Z<='1'; END IF;
            WHEN s2 =>          IF x = '1' THEN current <= s2; Z<='1';
                                ELSE current <= s1; Z<='0'; END IF;
        END CASE;
      END IF;
  END IF;
END PROCESS;

End;
```

(iii) Simulate the VHDL model **Behavior** and verify that it is working properly. Verify the correctness of your circuit by the following decimal numbers: 9, 10, and 15. Include a snapshot of the simulated waveform.




wave - default
File Edit View Insert Format Tools Window
/times3/clk 0
/times3/reset 0
/times3/x 0
/times3/z 0
Now 1300 ns
1 us
Cursor 1 0 ns
0 ns
0 ns to 1591 ns


wave - default
File Edit View Insert Format Tools Window
/times3/clk 0
/times3/reset 0
/times3/x 0
/times3/z 0
Now 1500 ns
1 us
Cursor 1 0 ns
0 ns
0 ns to 2 us