- Write the entity and architecture for a priority encoder with the following requirements:
  - Eight 1-bit inputs : I0 – I7
  - One 3-bit output: num_out
  - Use the following table

Active Input	Num_out	Priority
I7	111	1st highest priority
I6	110	
I5	101	
I4	100	
I3	011	
I2	010	
I1	001	
I0	000	8th lowest priority

```
ENTITY encode8to3 IS
    PORT ( I7,I6,I5,I4,I3,I2,I1,I0  : in  STD_LOGIC;
           num_out : out  STD_LOGIC_VECTOR(2 downto 0));
END encode8to3;

ARCHITECTURE priority OF encode8to3 IS
   BEGIN
      PROCESS (I7,I6,I5,I4,I3,I2,I1,I0)
         BEGIN
             IF I7 = '1' then
               num_out <= "111";
             ELSIF I6 = '1' then
               num_out <= "110";
             ELSIF I5 = '1' then
               num_out <= "101";
             ELSIF I4 = '1' then
               num_out <= "100";
             ELSIF I3 = '1' then
               num_out <= "011";
             ELSIF I2 = '1' then
               num_out <= "010";
             ELSIF I1 = '1' then
               num_out <= "001";
             ELSE
               num_out <= "000";
             END IF;
         END PROCESS;
   END priority;
```

Complete the output waves for the decode2to4 architecture below.

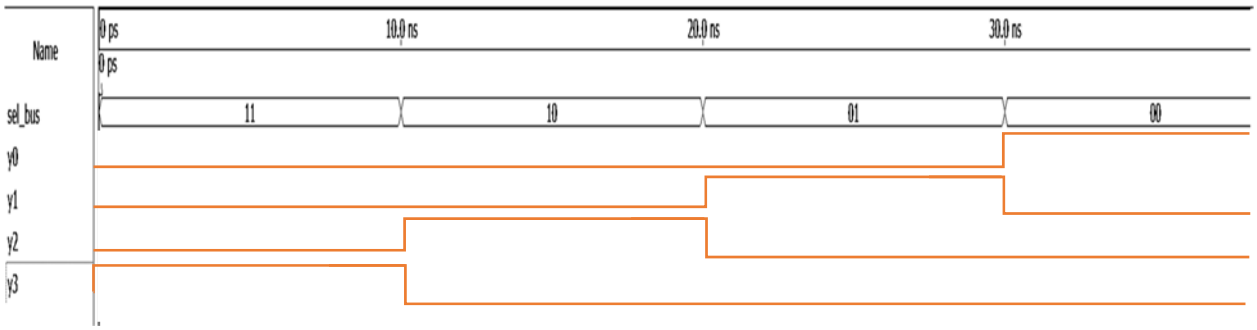
```
ARCHITECTURE behave OF decode2to4 IS

  signal sel_bus    : std_logic_vector(1 downto 0);
  signal Y_bus      : std_logic_vector(3 downto 0);

  BEGIN

     sel_bus <= s1 & s0;
     y0 <= Y_bus(0);
     y1 <= Y_bus(1);
     y2 <= Y_bus(2);
     y3 <= Y_bus(3);

     WITH sel_bus SELECT
        Y_bus <= "0001" WHEN "00",
                 "0010" WHEN "01",
                 "0100" WHEN "10",
                 "1000" WHEN OTHERS;
  END behave;
```



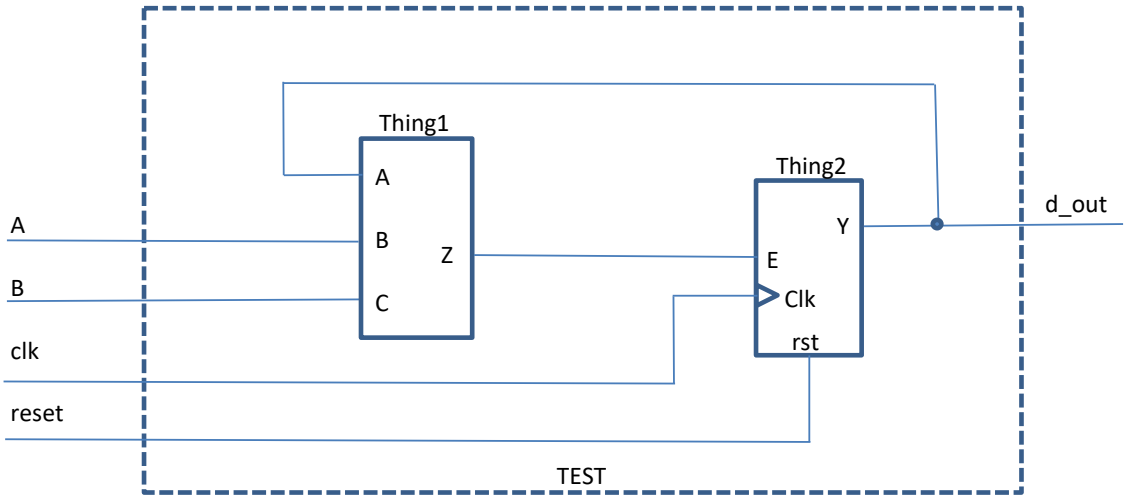
Rewrite the following mux architecture as a case statement

```
ARCHITECTURE behavioral OF mux IS
        SIGNAL selects: STD_LOGIC_VECTOR(1 DOWNTO 0);
BEGIN
        selects <= s1 & s0;
        output <= A when (selects = “00”) ELSE
                  B when (selects = “01”) ELSE
                  C when (selects = “10”) ELSE
                  D;
END behavioral;
```

```
--This is the case. Case requires a process
    PROCESS (selects, A, B, C, D ) is
        BEGIN
            CASE selects IS
                WHEN "00"   => Y_case <= A;
                WHEN "01"   => Y_case <= B;
                WHEN "10"   => Y_case <= C;
                WHEN OTHERS => Y_case <= D;
            END CASE;
        END PROCESS;
```

Complete the *STRUCTURAL* (hierarchical) VHDL model below. The component declarations for the two components are already given

1.




```
ENTITY test IS
   PORT (A, B, reset, clk :  IN  STD_LOGIC;
            d_out            : OUT STD_LOGIC);
end test;

ARCHITECTURE structural OF test IS

COMPONENT Thing2 IS
    PORT(E, rst, clk     : IN STD_LOGIC;
            Y            : OUT STD_LOGIC);
    END COMPONENT;

COMPONENT Thing1 IS
    PORT(A, B, C  : IN STD_LOGIC;
            Z         : OUT STD_LOGIC);
    END COMPONENT;
```

```
ENTITY test IS
    PORT (A, B, reset, clk :  IN  STD_LOGIC;
             d_out                : OUT STD_LOGIC);
end test;

ARCHITECTURE structural OF test IS

  COMPONENT Thing2 IS
     PORT(E, rst, clk     : IN STD_LOGIC;
             Y                  : OUT STD_LOGIC);
     END COMPONENT;

  COMPONENT Thing1 IS
     PORT(A, B, C  : IN STD_LOGIC;
             Z             : OUT STD_LOGIC);
     END COMPONENT;

     Signal int_d, ZtoE  : std_logic;

BEGIN

   U1: Thing1
       PORT MAP(A => int_d,
                B => A,
                C => B,
                Z => ZtoE);
   U2: Thing2
       PORT MAP(E => ZtoE,
                rst => reset,
                clk => clk,
                Y => int_d);
   d_out <= int_d;

END structural;
```