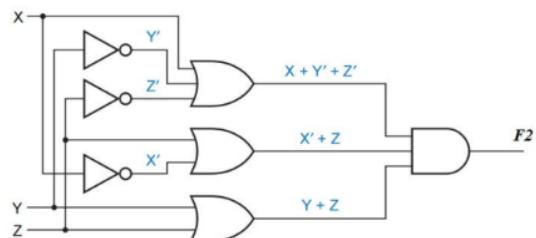
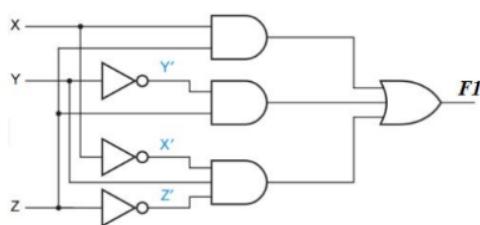




Universidad Tecnológica de la Mixteca  
 Cómputo Reconfigurable  
 Frida Ximena Martínez Lorenzo  
 Profesora: M.C. Arturo Pablo Sandoval García  
 Tercera Evaluación Parcial  
 Ingeniería en Computación || Grupo: 402-B

O A X A C P

1.-- Mediante VHDL en la representación de funciones booleanas represente los siguientes circuitos



```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity EX_1 is
5     Port ( x : in STD_LOGIC;
6             y : in STD_LOGIC;
7             z : in STD_LOGIC;
8             f1 : out STD_LOGIC;
9             f2 : out STD_LOGIC);
10 end EX_1;
11
12 architecture Funciones_Booleanas of EX_1 is
13 begin
14
15     F1 <= (not x and y and not z) or (not y and z) or (x and z);
16
17
18     F2 <= (x or not y or not z) and (not x or z) and (y or z);
19
20 end Funciones_Booleanas;
21

```



**4. -En un archivo jerárquico en representación de esquemático represente un Restador completo utilizando un decodificador en VHDL y Compuertas externas represente.**

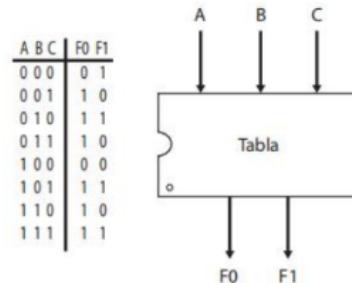
```
5 entity CINCO is
6     Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
7             F : out STD_LOGIC);
8 end CINCO;
9
10 architecture Behavioral of CINCO is
11
12 begin
13     WITH A SELECT
14         F <= '0' WHEN "0000",
15             '0' WHEN "0001",
16             '0' WHEN "0010",
17             '1' WHEN "0011",
18             '0' WHEN "0100",
19             '0' WHEN "0101",
20             '0' WHEN "0110",
21             '1' WHEN "0111",
22             '0' WHEN "1000",
23             '0' WHEN "1001",
24             '0' WHEN "1010",
25             '1' WHEN "1011",
26             '0' WHEN "1100",
27             '1' WHEN "1101",
28             '1' WHEN "1110",
29             '1' WHEN OTHERS;
30
31 end Behavioral;
```

```
11 -- COMPONENT DECLARATION FOR THE UNIT UNDER TEST (UUT)
12 COMPONENT CINCO
13 PORT (
14     A : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
15     F : OUT STD_LOGIC
16 );
17 END COMPONENT;
18
19
20 -- INPUTS
21 SIGNAL A: STD_LOGIC_VECTOR(3 DOWNTO 0) := (OTHERS => '0');
22 -- OUTPUTS
23 SIGNAL F: STD_LOGIC;
24
25 BEGIN
26     -- INSTANCIANDO LA UNIT UNDER TEST UUT
27     uut : CINCO PORT MAP (
28         A => A,
29         F => F
30 );
31
32     -- Stimulus process
33     stim_proc0: process
34     begin
35         wait for 100 ns;
36         a(0) <= not a(0);
37     end process;
```

```
33      stim_proc0: process
34      begin
35          wait for 100 ns;
36          a(0) <= not a(0);
37      end process;
38
39      stim_proc1: process
40      begin
41          wait for 200 ns;
42          a(1) <= not a(1);
43      end process;
44
45      stim_proc2: process
46      begin
47          wait for 400 ns;
48          a(2) <= not a(2);
49      end process;
50
51      stim_proc3: process
52      begin
53          wait for 800 ns;
54          a(3) <= not a(3);
55      end process;
56
57  end Behavioral;
58
```



6.- Utilizando la sentencias (**case**) en VHDL represente la simulación de la siguiente tabla de verdad.



```

2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4 -----
5 entity SEIS is
6     Port ( A : in  STD_LOGIC_VECTOR (2 downto 0);
7             F : out  STD_LOGIC_VECTOR (1 downto 0));
8 end SEIS;
9 -----
10 architecture Behevioral of SEIS is
11 begin
12
13 process(A) is
14 begin
15     case(A) is
16         when "000" => F <= "01";
17         when "001" => F <= "10";
18         when "010" => F <= "11";
19         when "011" => F <= "10";
20         when "100" => F <= "00";
21         when "101" => F <= "11";
22         when "110" => F <= "10";
23         when others => F <= "11";
24     end case;
25 end process;
26
27 end Behevioral;

```

```

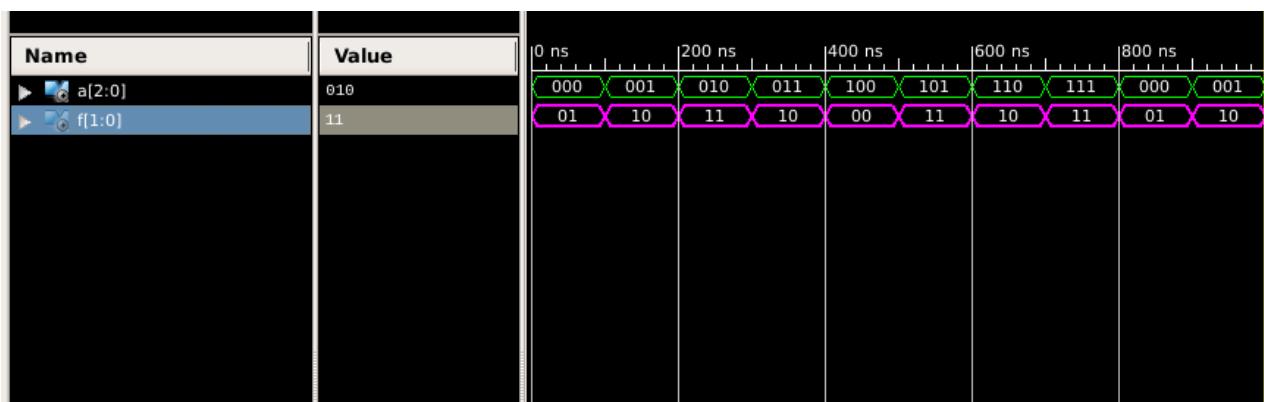
1 -----
2 LIBRARY ieee;
3 USE ieee.std_logic_1164.ALL;
4 -----
5 ENTITY TB_ex6 IS
6 END TB_ex6;
7
8 ARCHITECTURE behavior OF TB_ex6 IS
9     -- Component Declaration for the Unit Under Test (UUT)
10    COMPONENT SEIS
11        PORT(
12            A : IN  std_logic_vector(2 downto 0);
13            F : OUT  std_logic_vector(1 downto 0)
14        );
15    END COMPONENT;
16
17    --Inputs
18    signal A : std_logic_vector(2 downto 0) := (others => '0');
19    --Outputs
20    signal F : std_logic_vector(1 downto 0);
21
22 BEGIN
23     -- Instantiate the Unit Under Test (UUT)
24     uut: SEIS PORT MAP (
25         A => A,
26         F => F
27     );

```

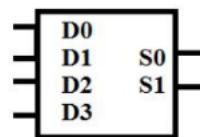
```

24     uut: SEIS PORT MAP (
25         A => A,
26         F => F
27     );
28
29     -- Stimulus process
30     stim_proc0: process
31     begin
32         wait for 100 ns;
33         a(0) <= not a(0);
34     end process;
35
36     stim_proc1: process
37     begin
38         wait for 200 ns;
39         a(1) <= not a(1);
40     end process;
41
42     stim_proc2: process
43     begin
44         wait for 400 ns;
45         a(2) <= not a(2);
46     end process;
47
48 END;
49

```



7.--Represente un Codificador de 4 a 2. **mediante (whit \_\_select\_\_ else)** presente el código en VHDL y su respectiva simulación en Test bench.



```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity EX7 is
5     Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
6             F : out STD_LOGIC_VECTOR (1 downto 0));
7 end EX7;
8
9 architecture Codificador of EX7 is
10 begin
11     with A select
12         F <= "00" when "0001",
13             "01" when "0010",
14             "10" when "0100",
15             "11" when others;
16 end Codificador;
17

```

```

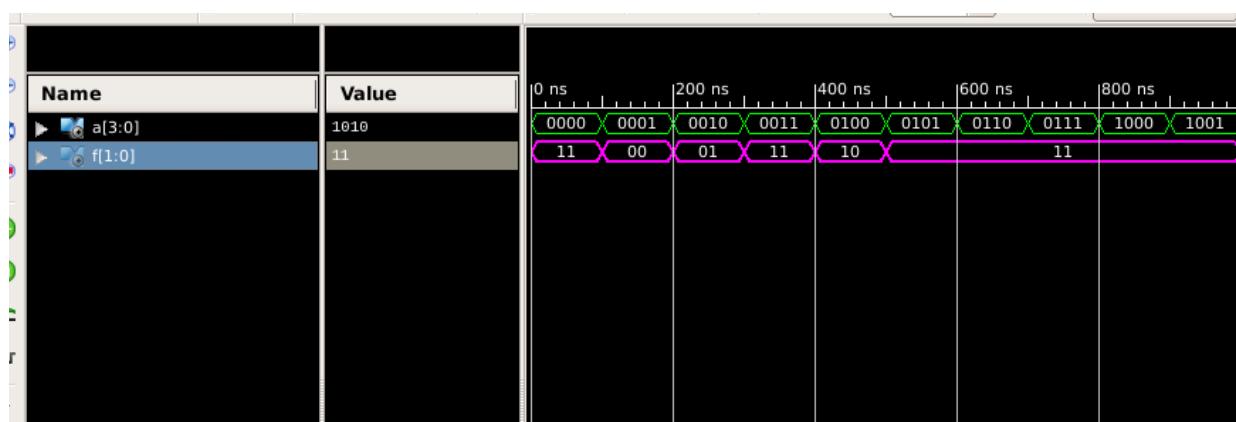
1 LIBRARY ieee;
2 USE ieee.std_logic_1164.ALL;
3
4 ENTITY TB_ex7 IS
5 END TB_ex7;
6
7 ARCHITECTURE behavior OF TB_ex7 IS
8     -- Component Declaration for the Unit Under Test (UUT)
9     COMPONENT EX7
10    PORT(
11        A : IN std_logic_vector(3 downto 0);
12        F : OUT std_logic_vector(1 downto 0)
13    );
14    END COMPONENT;
15
16    --Inputs
17    signal A : std_logic_vector(3 downto 0) := (others => '0');
18    --Outputs
19    signal F : std_logic_vector(1 downto 0);
20
21 BEGIN
22     -- Instantiate the Unit Under Test (UUT)
23     ut: EX7 PORT MAP (
24         A => A,
25         F => F
26     );
27

```

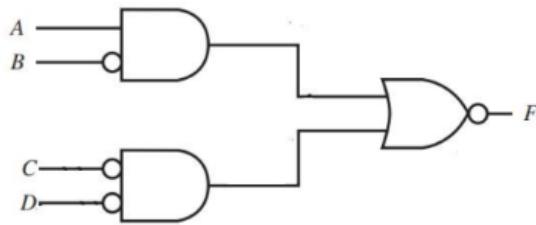
```

28
29      -- Stimulus process
30      stim_proc0: process
31      begin
32          wait for 100 ns;
33          A(0) <= not A(0);
34      end process;
35
36      stim_proc1: process
37      begin
38          wait for 200 ns;
39          A(1) <= not A(1);
40      end process;
41
42      stim_proc2: process
43      begin
44          wait for 400 ns;
45          A(2) <= not A(2);
46      end process;
47
48      stim_proc3: process
49      begin
50          wait for 800 ns;
51          A(3) <= not A(3);
52      end process;
53  END;

```



--Utilice la representación de la **estructura case** para la siguiente función en VHDL



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

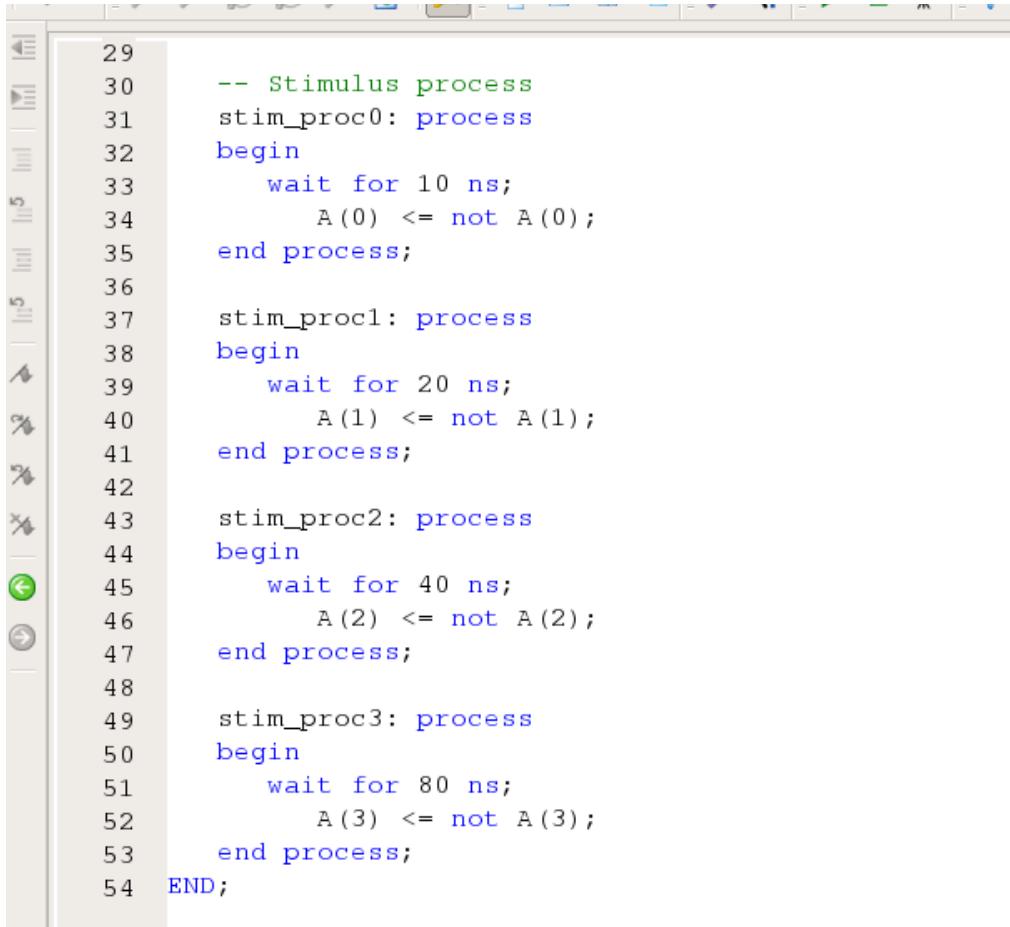
entity NUEVE is
    Port ( A : in STD_LOGIC_VECTOR (3 DOWNTO 0);
           F : out STD_LOGIC);
end NUEVE;

architecture Behavioral of NUEVE is
begin
    process (A)
    begin
        case (A) is
            when "0000" =>
                F <= '1';
            when "0001" =>
                F <= '0';
            when "0010" =>
                F <= '1';
            when "0011" =>
                F <= '1';
            when "0100" =>
                F <= '0';
            when "0101" =>
                F <= '0';
            when "0110" =>
                F <= '0';
            when "0111" =>
                F <= '1';
            when "1000" =>
                F <= '1';
            when "1001" =>
                F <= '0';
            when "1010" =>
                F <= '1';
            when "1011" =>
                F <= '0';
            when "1100" =>
                F <= '0';
            when "1101" =>
                F <= '0';
            when "1110" =>
                F <= '0';
            when "1111" =>
                F <= '0';
            when others =>
                F <= '0'; -- Default case
        end case;
    end process;
end Behavioral;
```

```

1 LIBRARY ieee;
2 USE ieee.std_logic_1164.ALL;
3
4 ENTITY TB_NUEVE IS
5 END TB_NUEVE;
6
7 ARCHITECTURE behavior OF TB_NUEVE IS
8     -- Component Declaration for the Unit Under Test (UUT)
9     COMPONENT NUEVE
10    PORT(
11        A : IN  std_logic_vector(3 downto 0);
12        F : OUT std_logic
13    );
14    END COMPONENT;
15
16    --Inputs
17    signal A : std_logic_vector(3 downto 0) := (others => '0');
18    --Outputs
19    signal F : std_logic;
20
21 BEGIN
22
23    -- Instantiate the Unit Under Test (UUT)
24    uut: NUEVE PORT MAP (
25        A => A,
26        F => F
27

```



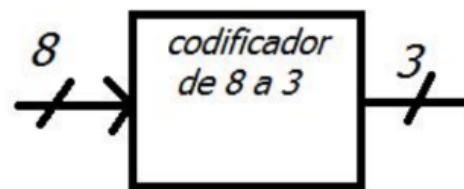
```

29      -- Stimulus process
30      stim_proc0: process
31      begin
32          wait for 10 ns;
33          A(0) <= not A(0);
34      end process;
35
36
37      stim_proc1: process
38      begin
39          wait for 20 ns;
40          A(1) <= not A(1);
41      end process;
42
43      stim_proc2: process
44      begin
45          wait for 40 ns;
46          A(2) <= not A(2);
47      end process;
48
49      stim_proc3: process
50      begin
51          wait for 80 ns;
52          A(3) <= not A(3);
53      end process;
54  END;

```



- Utilizando la estructura case, implemente un codificador de 8 a 3



```

3  use ieee.std_logic_1164.all;
4  -----
5  entity DIEZ is
6      port (
7          input : in std_logic_vector(7 downto 0);
8          output : out std_logic_vector(2 downto 0)
9      );
10 end entity;
11 -----
12 architecture behavioral of DIEZ is
13 begin
14     process (input)
15     begin
16         case input is
17             when "00000001" => output <= "000";
18             when "00000010" => output <= "001";
19             when "00000100" => output <= "010";
20             when "00001000" => output <= "011";
21             when "00010000" => output <= "100";
22             when "00100000" => output <= "101";
23             when "01000000" => output <= "110";
24             when "10000000" => output <= "111";
25             when others => output <= "000"; -- Valor predeterminado
26         end case;
27     end process;
28 end architecture;

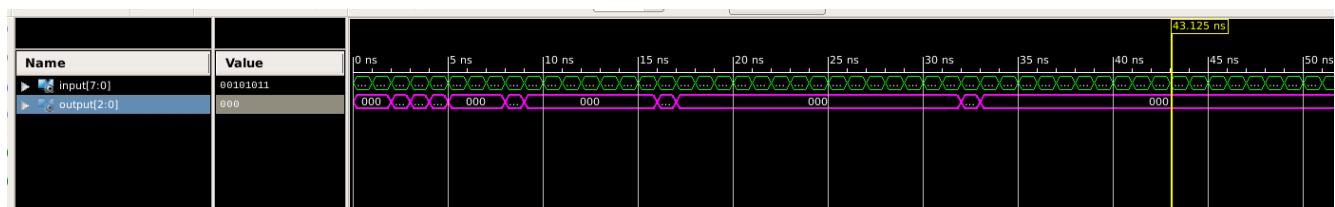
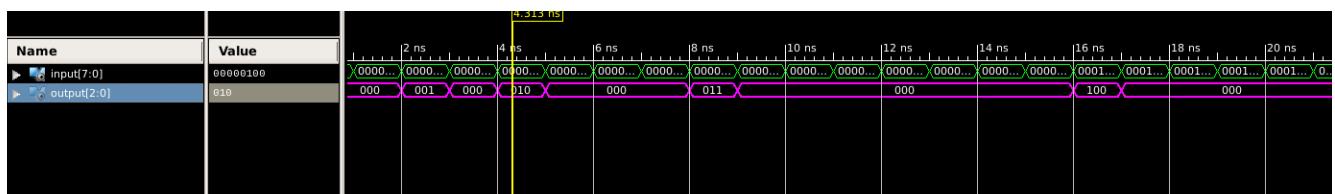
```

```
5 ENTITY TB_DIEZ IS
6 END TB_DIEZ;
7 -----
8 ARCHITECTURE behavior OF TB_DIEZ IS
9     -- Component Declaration for the Unit Under Test (UUT)
10    COMPONENT DIEZ
11        PORT(
12             input : IN  std_logic_vector(7 downto 0);
13             output : OUT std_logic_vector(2 downto 0)
14         );
15    END COMPONENT;
16
17    --Inputs
18    signal input : std_logic_vector(7 downto 0) := (others => '0');
19    --Outputs
20    signal output : std_logic_vector(2 downto 0);
21
22 BEGIN
23     -- Instantiate the Unit Under Test (UUT)
24     uut: DIEZ PORT MAP (
25         input => input,
26         output => output
27     );
28
29     -- Stimulus process
30     stim_proc0: process
31     begin
32         begin
33             wait for 1 ns;
34             input(0) <= not input(0);
35         end process;
36
37         stim_proc1: process
38         begin
39             wait for 2 ns;
40             input(1) <= not input(1);
41         end process;
42
43         stim_proc2: process
44         begin
45             wait for 4 ns;
46             input(2) <= not input(2);
47         end process;
48
49         stim_proc3: process
50         begin
51             wait for 8 ns;
52             input(3) <= not input(3);
53         end process;
54
55         stim_proc4: process
56         begin
57             wait for 16 ns;
58             input(4) <= not input(4);
```

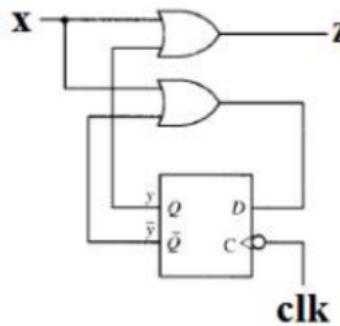
```

53
54     stim_proc4: process
55     begin
56         wait for 16 ns;
57         input(4) <= not input(4);
58     end process;
59
60     stim_proc5: process
61     begin
62         wait for 32 ns;
63         input(5) <= not input(5);
64     end process;
65
66     stim_proc6: process
67     begin
68         wait for 64 ns;
69         input(6) <= not input(6);
70     end process;
71
72     stim_proc7: process
73     begin
74         wait for 128 ns;
75         input(7) <= not input(7);
76     end process;
77
78 END;

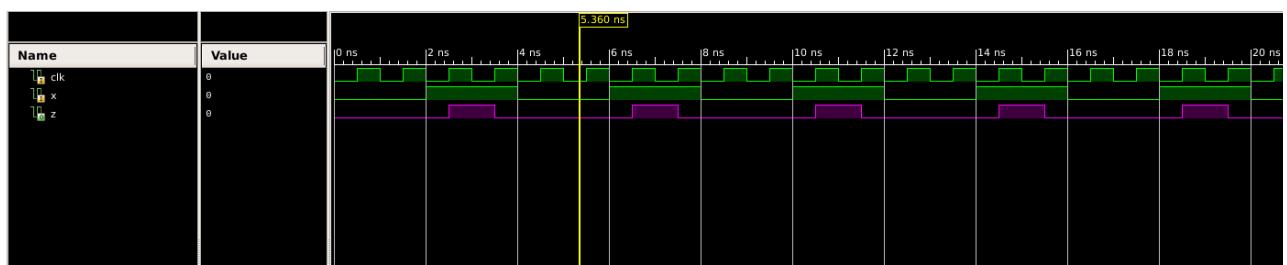
```



7. Representar el código y el diagrama de tiempos para el siguiente siguiente circuito.



```
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4
5 entity QUINCE is
6     Port ( X,CLK : in STD_LOGIC;
7             Z : out STD_LOGIC);
8 end QUINCE;
9
10 architecture Behavioral of QUINCE is
11 signal D, Q, NQ : std_logic;
12 begin
13     -- Compuerta AND para generar la señal D
14     D <= NQ and X;
15
16     -- Proceso de sensibilidad al flanko de subida del reloj
17     process (CLK)
18     begin
19         if rising_edge(CLK) then
20             -- Flip-flop D
21             Q <= D;
22             NQ <= not D;
23         end if;
24     end process;
25     -- Compuerta AND para generar la señal de salida Z
26     Z <= Q and X;
27 end architecture behavioral;
```



## IMPLEMENTE UN CONTADOR DESCENDENTE DE DOS BITS

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity CONTA_2BITS is
5     Port ( CLK,RESET : in STD_LOGIC;
6             Q : out STD_LOGIC_VECTOR (1 downto 0));
7 end CONTA_2BITS;
8
9
10 architecture Behavioral of CONTA_2BITS is
11     type StateType is (A, B, C, D);
12     signal CurrentState, NextState : StateType;
13 begin
14     process (CLK, RESET)
15     begin
16         if (RESET = '1') then
17             CurrentState <= D;
18         elsif (rising_edge(CLK)) then
19             CurrentState <= NextState;
20         end if;
21     end process;
22
23     process (CurrentState)
24     begin
25         case CurrentState is
26             when A =>
27                 NextState <= D;
28
29             when B =>
30                 NextState <= A;
31             when C =>
32                 NextState <= B;
33             when D =>
34                 NextState <= C;
35             when others =>
36                 NextState <= D;
37         end case;
38     end process;
39
40     process (CurrentState)
41     begin
42         case CurrentState is
43             when A =>
44                 Q <= "00";
45             when B =>
46                 Q <= "01";
47             when C =>
48                 Q <= "10";
49             when D =>
50                 Q <= "11";
51             when others =>
52                 Q <= "00";
53         end case;
54     end process;
55 end Behavioral;
```

```
29             when B =>
30                 NextState <= A;
31             when C =>
32                 NextState <= B;
33             when D =>
34                 NextState <= C;
35             when others =>
36                 NextState <= D;
37         end case;
38     end process;
39
40     process (CurrentState)
41     begin
42         case CurrentState is
43             when A =>
44                 Q <= "00";
45             when B =>
46                 Q <= "01";
47             when C =>
48                 Q <= "10";
49             when D =>
50                 Q <= "11";
51             when others =>
52                 Q <= "00";
53         end case;
54     end process;
55 end Behavioral;
```

