

# Tecnología de Computadores

## Hoja de Ejercicios

Luis Alberto Aranda Barjola



©2023 Luis Alberto Aranda Barjola.

Algunos derechos reservados. Este documento se distribuye bajo la licencia “Atribución-CompartirIgual 4.0 Internacional” de Creative Commons, disponible en <https://creativecommons.org/licenses/by-sa/4.0/deed.es>.

1. Dibuje el esquemático del circuito descrito por el siguiente código VHDL. Simplifique primero las ecuaciones lógicas para obtener el circuito mínimo.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ej1 is
    Port ( A, B, C : in  STD_LOGIC;
          Y, Z     : out STD_LOGIC
        );
end ej1;

architecture Dataflow of ej1 is
begin
    Y <= (A and B and C) or (A and B and (not C)) or (A and (not B) and C);
    Z <= (A and B) or ((not A) and (not B));
end Dataflow;

```

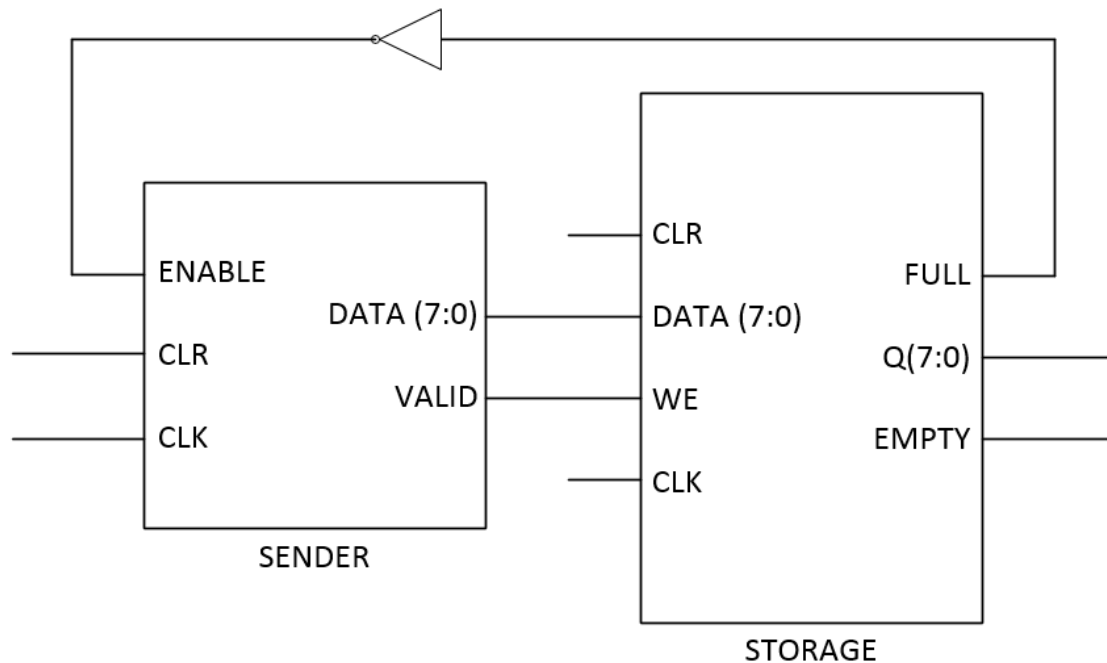
2. Escriba un módulo en VHDL que realice la función XOR para cuatro entradas. La entrada del módulo es “a<sub>3:0</sub>” y la salida es “y”.
3. Escriba un módulo en VHDL llamado “*minority*” con tres entradas A, B y C, y una salida Z. La salida del módulo es TRUE cuando al menos dos de las tres entradas son FALSE.
4. Escriba el código VHDL de un decodificador de 3:8.
5. Suponiendo que dispone del código VHDL de un multiplexor de 8:1. Escriba otro código VHDL que implemente la función lógica  $y = a\bar{b} + \bar{b}\bar{c} + \bar{a}bc$  utilizando uno de estos multiplexores y las puertas lógicas que necesite. Suponga que la entidad del multiplexor está definida como sigue:

```

entity mux8_1 is
    generic ( width : integer );
    port ( D0, D1, D2, D3, D4, D5, D6, D7 : in std_logic_vector(width-1 downto 0);
          S : in  std_logic_vector(2 downto 0);
          Y : out std_logic_vector(width-1 downto 0)
        );
end mux8_1;

```

6. Se desea diseñar un circuito digital en VHDL denominado MEMORY que consta de los módulos SENDER y STORAGE mostrados en el diagrama de bloques de la siguiente figura. Suponiendo que ya se dispone de los códigos VHDL de dichos módulos, escriba el código VHDL del circuito MEMORY que instancia los módulos tal y como se presentan en la figura.



7. Se desea sumar dos números en binario A y B con un sumador con acarreo anticipado. Calcule el valor de los Gi y Pi para poder implementar el sumador.

$$A = 1010\ 0001\ 1100\ 0011$$

$$B = 0101\ 1111\ 0001\ 0111$$

8. Para los mismos datos del ejercicio anterior, calcule el acarreo de salida del primer bloque de 4 bits si dicho bloque recibe un acarreo de entrada de '1'.

9. Hacer un LSR, ASR y ROR de 3 bits cada uno para los siguientes números:

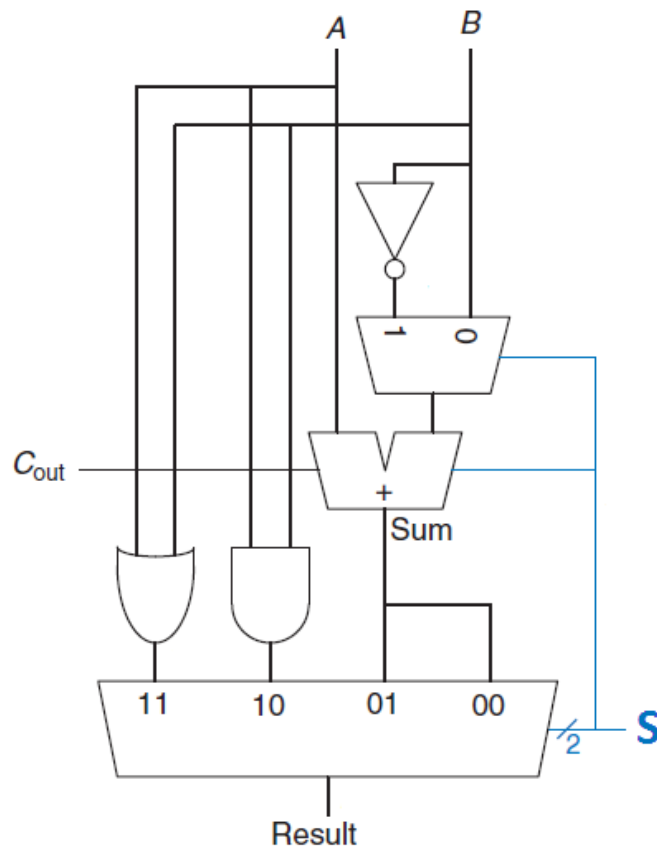
a) 24

c) 10

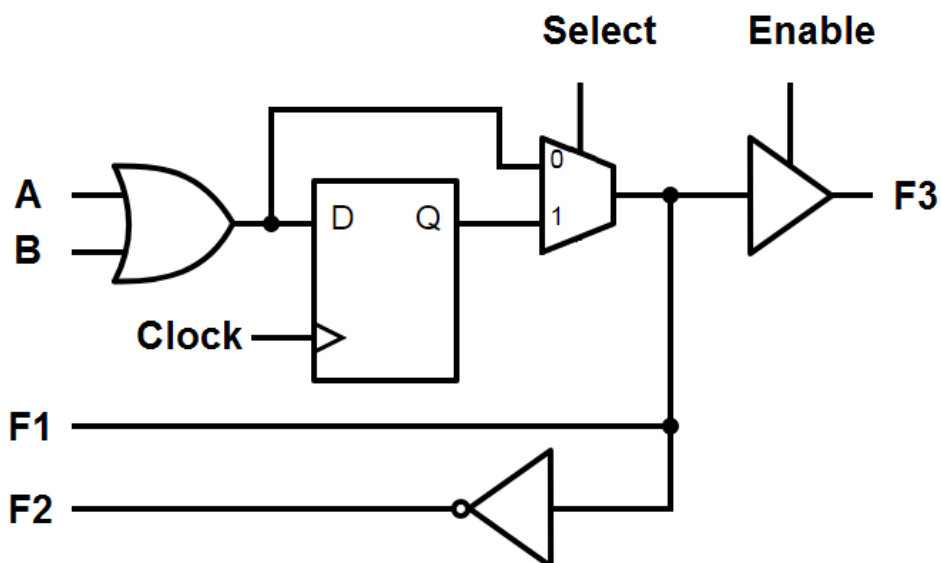
b) -18

d) 32

10. Escriba el código VHDL de la ALU de la siguiente figura suponiendo que ya dispone de los códigos VHDL del sumador y los multiplexores.



11. Escriba el código VHDL del circuito presentado en la siguiente figura suponiendo que se conocen los códigos VHDL de todos los componentes a excepción de la puerta OR y del inversor.



12. Escriba el código VHDL de un flip-flop JK. Este tipo de flip-flops tiene 3 entradas (J, K y CLK) y una salida (Q). En el flanco de reloj de subida, Q mantiene el valor anterior si  $J = K = 0$ . Si  $J = 1$ , entonces  $Q = 1$  y si  $K = 1$  entonces  $Q = 0$ . Si ambas señales  $J = K = 1$ , entonces el valor de Q es invertido.
13. Escriba el código VHDL de un flip-flop D con señales de set (S) y reset (R). De tal modo que si se activan ambas a la vez la salida del flip-flop (Q) será el valor de la entrada D, pero invertido. En los otros casos funciona normalmente según los valores de D, S y R.
14. Dado el siguiente código VHDL, determine qué error de sintaxis se está cometiendo que hace que dicho código no funcione. Una vez funcione el código, ¿a qué frecuencia de reloj se realizará la simulación?

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity tb is
--   Port ( );
end tb;

architecture Behavioral of tb is
component traffic_lights is
    Port (
        clk      : in  STD_LOGIC;
        rst      : in  STD_LOGIC;
        Green    : out STD_LOGIC;
        Red       : out STD_LOGIC;
        Yellow   : out STD_LOGIC
    );
end component;

signal clk, rst, Green, Red, Yellow : std_logic;
constant clk_period : time := 10 ns;

begin
DUT : traffic_lights PORT MAP (
    clk => clk;
    rst => rst;
    Green => Green;
    Red => Red;
    Yellow => Yellow
);

process begin
    clk <= '0';
    wait for clk_period/2;
    clk <= '1';
    wait for clk_period/2;
end process;

process begin
    rst <= '0';
    wait for clk_period * 15;

    rst <= '1';
    wait for clk_period;

    rst <= '0';
    wait;
end process;

end Behavioral;

```

15. Escribir el código VHDL de un multiplexor de 4 a 1 utilizando asignación condicional. Repita el mismo diseño, pero ahora usando un “case-when”.

16. Escribir la architecture de un multiplexor de 2 a 1 cuya entidad se muestra a continuación. Utilice descripción basada en comportamiento.

```
entity mux2to1 is
port ( w : IN  STD_LOGIC_VECTOR(1 DOWNTO 0);
      s : IN  STD_LOGIC;
      f : OUT STD_LOGIC );
end mux2to1;
```

17. Determine si las siguientes arquitecturas son equivalentes, es decir, si proporcionan el mismo comportamiento para una misma entidad dada. Justifique su respuesta.

```
architecture Caso1 of ejemplo is
begin
  Z <= A when B = '1' else
        C when D = '0' else
        '0';
end Caso1;
```

```
architecture Caso2 of ejemplo is
begin
  process (B) begin
    if (B = '1') then
      Z <= A;
    elsif (D = '0') then
      Z <= C;
    else
      Z <= '0';
    end if;
  end process;
end Caso2;
```

18. Escribir el código VHDL de un proceso que simula un reloj de 20 MHz en un testbench.

19. Escriba el código VHDL de un testbench que permita simular la unidad aritmicológica descrita en la tabla de la derecha sabiendo que A y B son señales de 8 bits. Explique los casos de ejemplo que ha elegido para la simulación y los valores de salida esperados en cada caso.

Opcode	Operación
00	A + B
01	A ⊕ B
10	asr (B)
11	A or B

20. Escriba el código VHDL de un testbench para simular un flipflop T. Simule un reloj con un periodo de 20 ns, un pulso de reset de 30 ns de duración y una señal de entrada T que tenga el siguiente comportamiento cíclico:

- Comienza valiendo '0' durante 10 ns
- Pasa a valer '1' y permanece así durante 40 ns
- Vuelve a valer '0' durante 30 ns

21. Escriba el código VHDL de la máquina de estados definida por la siguiente tabla de transición/salida sabiendo que se comienza en el estado S1 y que tiene un reset asíncrono activo a nivel alto. ¿Qué tipo de máquina es? Justifique su respuesta.

Estado actual	Estado siguiente			Salida (XYZ)
	AB = 0X	AB = 10	AB = 11	
S0	S0	S1	S2	100
S1	S0	S1	S2	110
S2	S0	S1	S2	111

22. Se desea simular una máquina de estados de Moore cuya función es la de detectar un cambio de '0' a '1' en la señal X de entrada. Escriba el código VHDL de un testbench que permita verificar su funcionamiento sabiendo que la máquina activa su salida Z cuando detecta dicho cambio, que trabaja a una frecuencia de 50 MHz y que tiene un reset síncrono activo a nivel bajo.
23. Dibuje el diagrama de estados de la máquina de estados descrita por el siguiente código VHDL.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FSM is
    port ( CLK, RST, T, B : in  STD_LOGIC;
          P                : out STD_LOGIC
        );
end FSM;

architecture Behavioral of FSM is
    type estados is (S0, S1, S2, S3, S4);
    signal CS, NS : estados;
begin
    process (CLK, RST) begin
        if (RST = '1') then
            CS <= S2;
        elsif (rising_edge(CLK)) then
            CS <= NS;
        end if;
    end process;
end process;

```

```
process (CS, T) begin
  case CS is
    when S0 =>
      if (T = '1') then
        NS <= S1;
      else
        NS <= S0;
      end if;
    when S1 =>
      if (T = '1') then
        NS <= S2;
      else
        NS <= S0;
      end if;
    when S2 =>
      if (T = '1') then
        NS <= S3;
      else
        NS <= S1;
      end if;
    when S3 =>
      if (T = '1') then
        NS <= S4;
      else
        NS <= S2;
      end if;
    when S4 =>
      if (T = '1') then
        NS <= S4;
      else
        NS <= S3;
      end if;
    when others => NS <= S2;
  end case;
end process;

-- Output Logic
P <= '1' when ((CS = S4) or (CS = S3) or
              ((CS = S2) and (B = '1')))
           else '0';
end Behavioral;
```



24. Dibuje el diagrama de estados asociado a la máquina de estados descrita por el siguiente código VHDL.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FSM is
    port ( CLK, RST, X : in  STD_LOGIC;
          Z1, Z2      : out STD_LOGIC
    );
end FSM;

architecture Behavioral of FSM is
    type estados is (A, B, C);
    signal CS, NS : estados;
begin
    process (CLK, RST) begin
        if (RST = '0') then
            CS <= C;
        elsif (rising_edge(CLK)) then
            CS <= NS;
        end if;
    end process;

    process (CS, X) begin
        case CS is
            when A =>
                Z1 <= '0';
                if (X = '0') then
                    NS <= A; Z2 <= '1';
                else
                    NS <= B; Z2 <= '0';
                end if;
            when B =>
                Z1 <= '1';
                if (X = '0') then
                    NS <= A; Z2 <= '0';
                else
                    NS <= C; Z2 <= '1';
                end if;
            when C =>
                Z1 <= '1';
                if (X = '0') then
                    NS <= B; Z2 <= '1';
                else
                    NS <= A; Z2 <= '0';
                end if;
            when others =>
                NS <= A; Z1 <= '1'; Z2 <= '0';
        end case;
    end process;
end Behavioral;

```

25. Escribir el código VHDL de un registro de 8 bits activo por flanco de subida con señal de habilitación asíncrona activa por nivel alto.