

1)

- una PAL está diseñada para ecuaciones en forma de SOP mientras que una GAL está diseñada para POS.
- Falso: Ambas son SOP.
- una GAL tiene una matriz AND y una OR, ambas reprogramables.
- Falso: La matriz de entrada AND es reprogramable. La matriz OR es fija
- una GAL puede implementar expresiones más complicadas, porque tiene mayor cantidad de términos producto que cualquier PAL.
- Verdadero. Depende del integrado.
- usa tecnología EECMOS, de modo que es reprogramable.
- Verdadero para la GAL. La PAL no es reprogramable

2)

- Los ASICs poseen arquitecturas con conexiones fijas.
- Falso.
- Un ASICs Full Custom está pensado para el desarrollo de un diseño en particular a gran escala.
- Verdadero.
- Un ASICs Full Custom se utiliza para el diseño de prototipos.
- Falso.
- Los Microprocesadores poseen arquitecturas fijas que no pueden reconfigurarse.
- Verdadero.
- Un ASICs Full Custom es un circuito prediseñado en el cual el usuario establece las conexiones.
- Falso.
- Las FPGA son configuradas por el usuario.
- Verdadero.
- Las FPGA son muy utilizadas para pruebas y desarrollo de prototipos.
- Verdadero.
- El consumo de potencia es mucho menor en las FPGA que en los circuitos Full-Custom.
- Falso.
- Un microcontrolador (con las mismas características) implementado dentro de una FPGA es más rápido que un microcontrolador dedicado.
- Falso.

3)

a) Con procesos

```
library ieee;  
use ieee.std_logic_1164.all;
```

```
entity sumador_completo is  
    port (A, B, Cin : in bit;
```

```

        Sum, Cout : out bit);
end sumador_completo;

```

architecture behavior of sumador_completo is

```
begin
```

```

    process (A, B, Cin)
    begin
        if (A = '0' and B = '0' and Cin = '0') then --000
            Sum <= '0';
            Cout <= '0';
        elsif (A = '0' and B = '0' and Cin = '1') then --001
            Sum <= '1';
            Cout <= '0';
        elsif (A = '0' and B = '1' and Cin = '0') then --010
            Sum <= '1';
            Cout <= '0';
        elsif (A = '0' and B = '1' and Cin = '1') then --011
            Sum <= '0';
            Cout <= '1';
        elsif (A = '1' and B = '0' and Cin = '0') then --100
            Sum <= '1';
            Cout <= '0';
        elsif (A = '1' and B = '0' and Cin = '1') then --101
            Sum <= '0';
            Cout <= '1';
        elsif (A = '1' and B = '1' and Cin = '0') then --110
            Sum <= '0';
            Cout <= '1';
        elsif (A = '1' and B = '1' and Cin = '1') then --111
            Sum <= '1';
            Cout <= '1';
        end if;
    end process;

```

```

    end behavior;

```

b) Sin procesos

```

    library ieee;
    use ieee.std_logic_1164.all;

```

entity sumador_completo is

```

    port (A, B, Cin : in bit;
          Sum, Cout : out bit);
end sumador_completo;

```

architecture behavior of sumador_completo is

begin

sum <= A xor B xor Cin;

Cout <= (A and B) or (A and Cin) or (B and Cin)

end behavior;

4) a) MUX de 4 entradas. Tabla de verdad:

sel	out
00	0
01	a
10	b
11	c

b) Buffer Tri State

op	Salida
0	Z
1	entrada

5)

- Bloques lógicos: realizan las operaciones lógicas del programa y están compuestos por Look Up Table, lógica de acarreo, multiplexores y registros.
- Bloques de entrada/salida: permiten que la FPGA se comunique con otros dispositivos externos, facilitando el intercambio de datos con el entorno.
- Interconexiones programables: permiten la comunicación entre los diferentes bloques lógicos y de Entrada/Salida en la FPGA. Estas conexiones pueden configurarse para que cada componente se comunique con los demás de acuerdo a las necesidades del circuito.

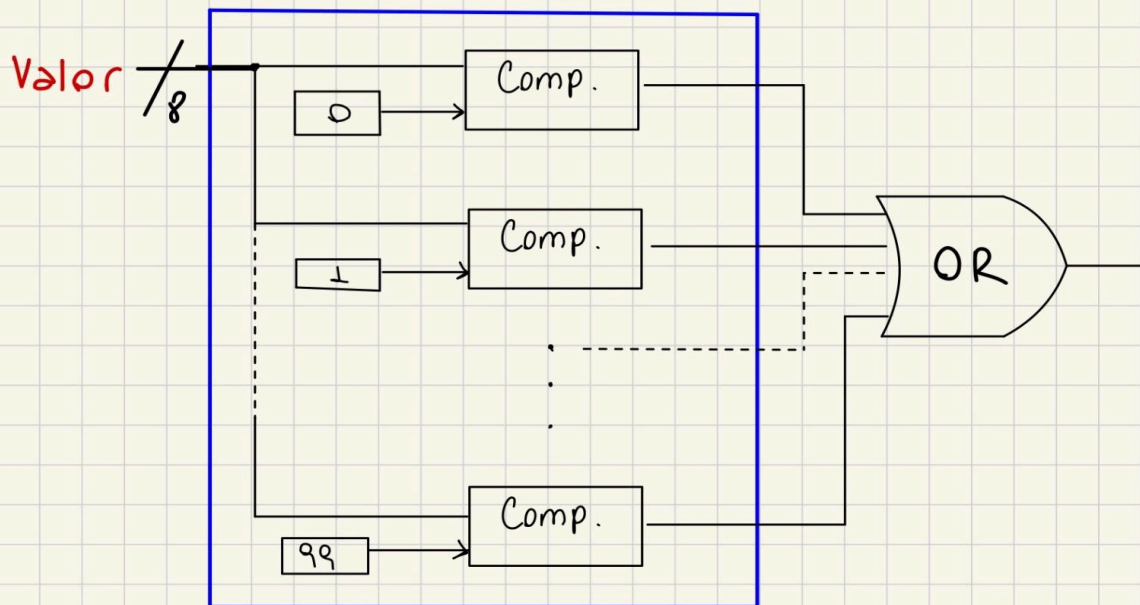
6) Un lenguaje de descripción de hardware (HDL), como VHDL o Verilog, permite describir el comportamiento y la estructura de un circuito digital, generando hardware físico (en FPGAs o ASICs) mediante síntesis. A diferencia de los lenguajes de programación tradicionales, un HDL no es secuencial, describe eventos en paralelo y controla la temporización de señales. Por otro lado, los lenguajes de programación ejecutan instrucciones en forma secuencial.

7) La entidad tendrá 3 entradas y 8 salidas. Se sintetiza un Decodificador 3 a 8.

8)

a)

a) Esquema de la entidad



b) El circuito es puramente combinacional. Esto se debe a que el proceso de comparación se realiza en forma instantánea y en paralelo, sin la participación de elementos de memoria.

c) La frecuencia máxima de operación quedará determinada por los retardos de la lógica combinacional y por el tiempo que tarde en hacerse la lectura de datos de la RAM.

Una forma de incrementar la frecuencia sería cargar los datos previamente, reduciendo los retardos generados por la lectura de la memoria. No obstante, esto significaría que deberá programarse la placa nuevamente cuando cambien los datos de memoria.

d) Se tienen 100 vectores (0 a 99) y de 8 bits cada uno. Por lo tanto, el arreglo tendrá un tamaño de $8 \times 100 = 800$ bits.