



Los archivos generados deben respetar el siguiente formato de nombre: `guiaDeClase00_ejercicioNumero.vhd`. Si el archivo es un testbench, el formato del nombre es el siguiente: `guiaDeClase00_ejercicioNumero_tb.vhd`. Por ejemplo, el archivo del ejercicio 1 debe llevar el nombre `guiaDeClase00_01.vhd`. Si el ejercicio tuviera ítems a y b, por ejemplo, el nombre deberá ser `guiaDeClase00_01_A.vhd` para el punto A. Todos los archivos deberán ser subidos al repositorio dentro de una carpeta con el nombre `guiaDeClase00` junto con sus simulaciones (solo suba los archivos `.vhd`).

1. Implemente en VHDL una compuerta AND de dos entradas.

```
entity myAnd2 is
  Port ( a, b : in std_logic;
         c : out std_logic);
end myAnd2;
```

- a. Observe la vista RTL y tecnológica.
- b. Verifique su tabla de la verdad en simulación.

2. Implemente en VHDL una compuerta AND de cuatro entradas.

```
entity myAnd4 is
  Port ( a : in std_logic_vector (3 downto 0);
         c : out std_logic);
end myAnd4;
```

3. Implemente un sumador completo de 1 bit.

```
entity fullAdder is
  Port ( a, b, ci : in std_logic;
         co, s : out std_logic);
end fullAdder;
```

4. Implemente un generador de paridad par de 4 bits. (Use XOR)

```
entity paridadPar4 is
  Port ( a : in std_logic_vector (3 downto 0);
         p : out std_logic);
end paridadPar4;
```

La salida `p` será 1 cuando la palabra de entrada tenga cantidad impar de unos.

Por ejemplo

- Con una entrada "0101", la salida `p` valdrá cero.
- Con una entrada "0111", la salida `p` valdrá uno.

5. Implemente un multiplexor de 2 vías. (Use when - else)

```
entity myMux is
  Port ( a : in std_logic_vector (1 downto 0);
         c : in std_logic;
         s : out std_logic);
end myMux;
```



6. Implemente un buffer tri-State. (Use when - else)

```
entity myTriState is
  Port ( entrada : in std_logic;
        control : in std_logic;
        salida : out std_logic);
end myTriState;
```

control	entrada	salida
0	0	Z
0	1	Z
1	0	0
1	1	1

7. Implemente ocho buffers tri-State con una única señal de control para todos. (Use when - else)

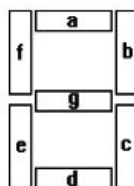
```
entity myTriState8 is
  Port ( entrada : in std_logic_vector (7 downto 0);
        control : in std_logic;
        salida : out std_logic_vector (7 downto 0));
end myTriState8;
```

8. Implemente un comparador de magnitudes de 8 bits. (Use when - else)

```
entity myMagCmp8 is
  Port ( a, b : in std_logic_vector (7 downto 0);
        igual, aMayorB, bMayorA : out std_logic);
end myMagCmp8;
```

9. Implemente un decodificador BCD a 7 segmentos. (Use with - select)

```
entity myDeco7Seg is
  Port ( entrada : in std_logic_vector (3 downto 0);
        a, b, c, d, e, f, g : out std_logic);
end myDeco7Seg;
```





Entrada	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	0	1	1	1	0	0	1
0100	1	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	1	1	1
1010	1	1	1	0	1	1	1
1011	0	0	1	1	1	1	1
1100	1	0	0	1	1	1	0
1101	0	1	1	1	1	0	1
1110	1	0	0	1	1	1	1
1111	1	0	0	0	1	1	1

10. Implemente un decodificador de 3 a 8. (use case - when)

```
entity myDeco3_8 is
  Port ( entrada : in std_logic_vector (2 downto 0);
        salida : out std_logic_vector (7 downto 0));
end myDeco3_8;
```

11. Implemente un codificador de 4 a 2. (use case - when)

```
entity myCoder4_2 is
  Port ( entrada : in std_logic_vector (3 downto 0);
        salida : out std_logic_vector (1 downto 0));
end myCoder4_2;
```



12. Implemente un circuito que permite dividir una magnitud de 8 bits por 1, 2, 4, 8 según se seleccione.

```
entity myDivMag8 is
  Port ( entrada : in  std_logic_vector (7 downto 0);
        div: in  std_logic_vector (1 downto 0);
        salida : out std_logic_vector (7 downto 0));
end myDivMag8;
```

13. Implemente un circuito que permite dividir un número signado en convenio CA2 de 8 bits por 1, 2, 4, 8 según se seleccione.

```
entity myDiv8 is
  Port ( entrada : in  std_logic_vector (7 downto 0);
        div: in  std_logic_vector (1 downto 0);
        salida : out std_logic_vector (7 downto 0));
end myDiv8;
```