Universidad Nacional Autónoma de México
Facultad de Ingeniería
Diseño Digital VLSI

Descripción de *hardware*

VHDL para circuitos combinacionales

M.I. Bryan Emmanuel Alvarez Serna

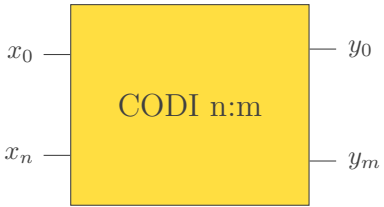
“Vive como si fueras a morir mañana; aprende como si fueras a vivir siempre.”

Mahatma Gandhi

Contenido

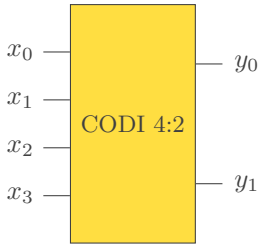
Codificador

- Circuito para mostrar una salida en binario cuando hay una entrada activa.
- Se debe cumplor $n > m$.
- La salidas m obedecen a $n = 2^m$.



Codificador

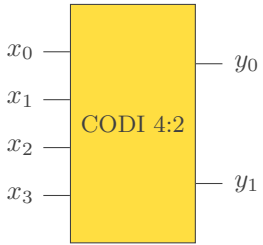
Ejemplo: Diseñar un CODI 4:2 sin prioridad.



x_0	x_1	x_2	x_3	y_0	y_1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Codificador

Ejemplo: Diseñar un CODI 4:2 sin prioridad.



x_0	x_1	x_2	x_3	y_0	y_1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY codi IS
        PORT(X: IN STD_LOGIC_VECTOR(0 TO 3);
             Y: OUT STD_LOGIC_VECTOR(0 TO 1));
END ENTITY;

ARCHITECTURE BEAS OF codi IS
BEGIN

    WITH X SELECT
         Y <= "1000" WHEN "00",
              "0100" WHEN "01",
              "0010" WHEN "10",
              "0001" WHEN "11";
END ARCHITECTURE;
```

Contenido

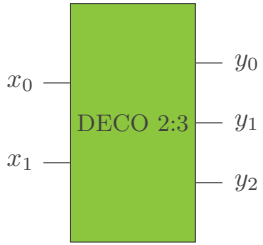
- Codificador
- **Decodificador**
- Demultiplexor
- Multiplexor

Decodificador

- Circuito para convertir de un código a otro.
- n son las entradas y m las salidas.
- Los tipos de datos y tamaño puede variar.

Decodificador

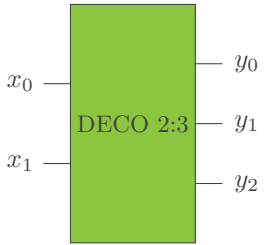
Ejemplo: Diseñar un DECO 2:3 para contar de 2 en 2 del 0 al 6.



x_0	x_1	y_0	y_1	y_2
0	0	0	0	0
0	1	0	1	0
1	0	1	0	0
1	1	1	1	0

Decodificador

Ejemplo: Diseñar un DECO 2:3 para contar de 2 en 2 del 0 al 6.



x_0	x_1	y_0	y_1	y_2
0	0	0	0	0
0	1	0	1	0
1	0	1	0	0
1	1	1	1	0

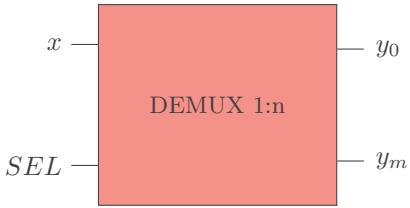
```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY deco IS
        PORT(X: IN STD_LOGIC_VECTOR(0 TO 1);
             Y: OUT STD_LOGIC_VECTOR(0 TO 2));
END ENTITY;

ARCHITECTURE BEAS OF deco IS
BEGIN

   WITH X SELECT
        Y <= "000" WHEN "00",
             "010" WHEN "01",
             "100" WHEN "10",
             "110" WHEN "11";

END ARCHITECTURE;
```

Contenido

- Codificador
- Decodificador
- **Demultiplexor**
- Multiplexor

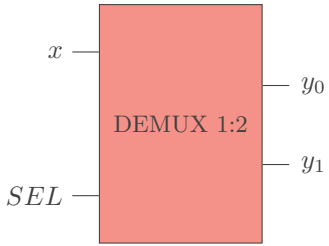
Demultiplexor

- Es similiar al decodificador pero con una señal de selección.
- El tamaño de SEL depende del número de salidas.
- $n = 2^{SEL}$.



Demultiplexor

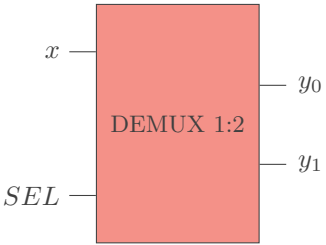
Ejemplo: Diseñar un DEMUX 1:2.



SEL	y_0	y_1
0	x	0
1	0	x

Demultiplexor

Ejemplo: Diseñar un DEMUX 1:2.

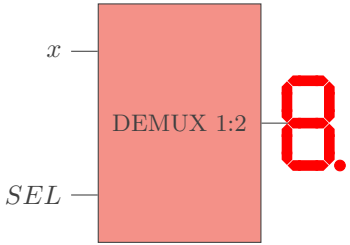


SEL	y_0	y_1
0	x	0
1	0	x

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY demux IS
   PORT(X: IN INTEGER RANGE 0 TO 15;
        Y0,Y1: OUT INTEGER RANGE 0 TO 15;
        SEL: IN STD_LOGIC);
END ENTITY;

ARCHITECTURE BEAS OF demux IS
BEGIN
    PROCESS(SEL)
    BEGIN
        IF SEL = '0' THEN
           Y0 <= X; Y1 <= 0;
        ELSE
           Y1 <= X; Y0 <= 0;
        END IF;
    END PROCESS;
END ARCHITECTURE;
```

Demultiplexor

Ejemplo: Diseñar un DEMUX 1:2 para mostrar en un display de 7 segmentos las vocales mayúsculas y minúsculas.



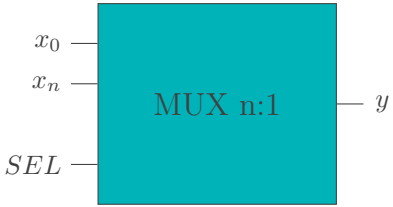

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY demuxabc IS
        PORT(X: IN INTEGER RANGE 0 TO 4;
             SEL: IN STD_LOGIC;
             SEG: OUT STD_LOGIC_VECTOR(0 TO 7));
END ENTITY;
ARCHITECTURE BEAS OF demuxabc IS
BEGIN
    PROCESS(SEL)
    BEGIN
        IF SEL = '0' THEN
            CASE X IS
                WHEN 0 => SEG <= X"11";-- A
                WHEN 1 => SEG <= X"61";-- E
                WHEN 2 => SEG <= X"3E";-- I
                WHEN 3 => SEG <= X"03";-- O
                WHEN 4 => SEG <= X"83";-- U
            END CASE;
        ELSE
            CASE X IS
                WHEN 0 => SEG <= X"05";-- a
                WHEN 1 => SEG <= X"21";-- e
                WHEN 2 => SEG <= X"BE";-- i
                WHEN 3 => SEG <= X"C5";-- o
                WHEN 4 => SEG <= X"C7";-- u
            END CASE;
        END IF;
    END PROCESS;
END ARCHITECTURE;
```

Contenido

- Codificador
- Decodificador
- Demultiplexor
- **Multiplexor**

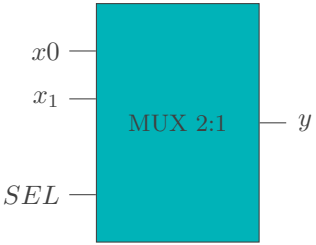
Multiplexor

- Es un circuito con n entradas y una sola salida.
- El tamaño de SEL depende del número de las entradas.
- $n = 2^{SEL}$.



Multiplexor

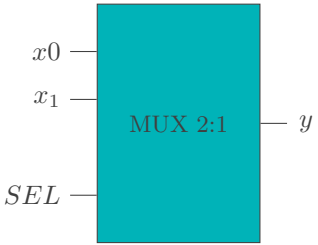
Ejemplo: Diseñar un MUX 2:1.



SEL	y_0
0	x_0
1	x_1

Multiplexor

Ejemplo: Diseñar un MUX 2:1.



SEL	y
0	x_0
1	x_1

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY mux IS
    PORT(X0, X1: IN INTEGER RANGE 0 TO 15;
             SEL: IN STD_LOGIC;
               Y: OUT INTEGER RANGE 0 TO 15);
END ENTITY;
ARCHITECTURE BEAS OF mux IS
BEGIN
    WITH SEL SELECT
        Y <= X0 WHEN '0',
             X1 WHEN OTHERS;
END ARCHITECTURE;
```

Multiplexor

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY mux IS
    PORT(X0, X1: IN INTEGER RANGE 0 TO 15;
           SEL: IN STD_LOGIC;
             Y: OUT INTEGER RANGE 0 TO 15);
END ENTITY;
ARCHITECTURE BEAS OF mux IS
BEGIN
    WITH SEL SELECT
        Y <= X0 WHEN '0',
             X1 WHEN OTHERS;
END ARCHITECTURE;
```

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY mux IS
   PORT(X0, X1: IN INTEGER RANGE 0 TO 15;
          SEL: IN STD_LOGIC;
            Y: OUT INTEGER RANGE 0 TO 15);
END ENTITY;
ARCHITECTURE BEAS OF mux IS
BEGIN
    PROCESS(SEL)
    BEGIN
        IF SEL = '0' THEN
           Y <= X0;
        ELSE
           Y <= X1;
        END IF;
    END PROCESS;
END ARCHITECTURE;
```

Multiplexor

RTL Viewer.