



# **UNIVERSIDAD DE COLIMA**

## **FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA**

### **CURSO: IMPLEMENTACION DE CIRCUITOS DIGITALES**

#### **ACTIVIDAD 4: Asignaciones Condicionales Concurrentes en VHDL**

**Instructor: Dr. Alberto Manuel Ochoa Brust**

---

Esta actividad está planeada con un enfoque de aprendizaje colaborativo, en el cual lo más importante es el trabajo en equipo para lograr el objetivo de aprender significativamente. Para ello se formarán grupos de 3 personas, que tendrán que leer este documento. Uno de los integrantes deberá operar la PC siguiendo paso a paso las instrucciones que se indican aquí. La calificación de esta actividad va de 0 a 100 y se asignará proporcionalmente al avance logrado al final de la sesión

---

El objetivo de esta actividad es conocer el uso de las asignaciones condicionales y concurrentes de VHDL

### Asignación condicional WHEN-ELSE

Le asigna a una señal o puerto de salida alguno de los varios valores, según se cumpla alguna de las condiciones propuestas. A continuación se presenta la sintaxis con una explicación en lenguaje natural (coloquial) escrita como comentarios:

```
salida <= valor1 WHEN condicion1 ELSE -- Se asigna a salida el valor1 cuando condicion1 sea
                                     -- verdadera, si no (ELSE) ...
      valor2 WHEN condicion2 ELSE    -- se asigna a salida el valor2 cuando condicion2 sea
                                     -- verdadera, si no (ELSE) ...
      ⋮                               ⋮
      valorN WHEN condicionN ELSE    -- se asigna a salida el valorN cuando condicionN sea
                                     -- verdadera, si no (ELSE) ...
      valorM;                        -- se cumplen ninguna de las condiciones
                                     -- anteriores, se le asigna a salida el valorM.
```

La asignación WHEN-ELSE consiste de una salida (*salida*), que puede ser una señal interna o puerto de salida, de cualquier tipo de dato, dimensión y rango, y de una serie de sentencias WHEN, cada una ellas relacionadas con un valor (*valor1*, *valor2*, ..., *valorN* y *valorM*), del mismo tipo que la salida, y con una condición en forma de expresión de tipo BOOLEAN (*condicion1*, *condicion2*, ..., y *condicionN*).

La salida será asignada con alguno de los valores listados, dependiendo del valores de las expresiones booleanas (TRUE ó FALSE), por ejemplo si, *condicion2*=TRUE, entonces a *salida* se le asignará *valor2*; si, *condicionN*=TRUE, entonces a *salida* se le asignará *valorN*. Cuando más de una condición sea verdadera, la *salida* tomará el valor de la condición que se escribió antes, esto permite darle a la instrucción una prioridad. Por ejemplo, si *condicion1* y *condicion2* son verdaderas, entonces a *salida* se le asignará *valor1*, ya que fue definido primero en la sentencia.

En toda expresión condicional, que describa hardware de forma concurrente, es importante incluir todas las opciones posibles y contemplar todos los casos posibles de una variable. En este sentido es obligatorio siempre finalizar esta expresión condicional con un ELSE antes de definir el último valor (*valorM*).

A continuación se presenta el diseño de un multiplexor de 2 a 1, usando la asignación condicional WHEN-ELSE.

```
ENTITY mux2a1 IS
  PORT (A, B, SEL: IN STD_LOGIC;
        Z: OUT STD_LOGIC);
END mux2a1;

ARCHITECTURE Behavioral OF mux2a1 IS
BEGIN
  Z <= A WHEN SEL= '0' ELSE B;    -- Al puerto Z se le asigna el valor del puerto de entrada
                                  -- A cuando SEL='0', si no (SEL='1') a Z se le asigna el
                                  -- valor del puerto de entrada B
END Behavioral;
```

El siguiente es un decodificador en VHDL de 2 a 4 líneas, usando la asignación concurrente WHEN-ELSE. Para los puertos de entrada, las líneas de selección es un puerto (de 2 bits) de tipo STD\_LOGIC\_VECTOR (1 DOWNT0 0), con los elementos ordenados de derecha a izquierda; las salidas también están agrupadas, pero en un puerto (de 4 bits) de tipo STD\_LOGIC\_VECTOR (3 DOWNT0 0).

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY dec2a4 IS
PORT(sel: IN STD_LOGIC_VECTOR (1 DOWNT0 0);
      salida: OUT STD_LOGIC_VECTOR (3 DOWNT0 0));
END dec2a4;

ARCHITECTURE behavioral OF dec2a4 IS
BEGIN
    salida <= "0001" WHEN sel="00" ELSE --salida es igual a "0001" cuando sel="00", si no
        "0010" WHEN sel="01" ELSE --salida es igual a "0010" cuando sel="01", si no
        "0100" WHEN sel="10" ELSE --salida es igual a "0100" cuando sel="10", si no
        "1000"; --salida es igual a "1000" para los otros casos (sel="11").
end Behavioral;

```

En la Figura 4.1 se muestra una simulación del código anterior.

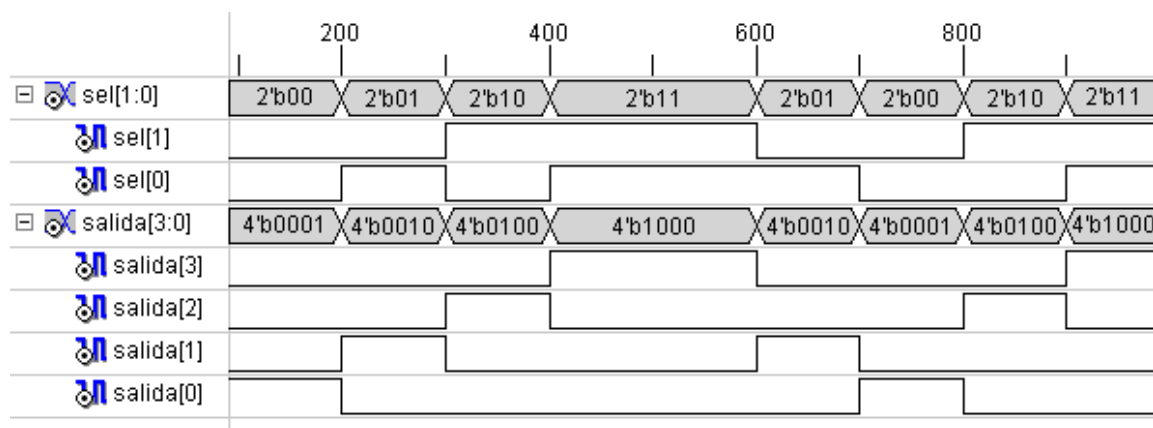


Figura 4.1: Simulación de un decodificador de 2 a 4.

#### PROBLEMA 4.1

a) Diseñe un multiplexor de 4 a 1 en VHDL usando la instrucción WHEN-ELSE. dicho diseño deberá tener una "Entity" con un puerto de entrada (SEL) de 2 bits, para las líneas de selección, un puerto de entrada (X) de 4 bits para los datos de entrada del mux y un puerto de salida (Z) de un bit. Como es de esperarse las entradas y la salida se relacionarán como lo indica la siguiente tabla.

Tabla 4.1. Tabla de verdad de un multiplexor de 4 a 1.

SEL(1)	SEL(0)	Z
0	0	X(0)
0	1	X(1)
1	0	X(2)
1	1	X(3)

b) Escriba en lenguaje natural el funcionamiento la instrucción WHEN-ELSE del programa diseñado.

c) Complete las formas de onda el puerto de entrada SEL en la Figura 4.2.

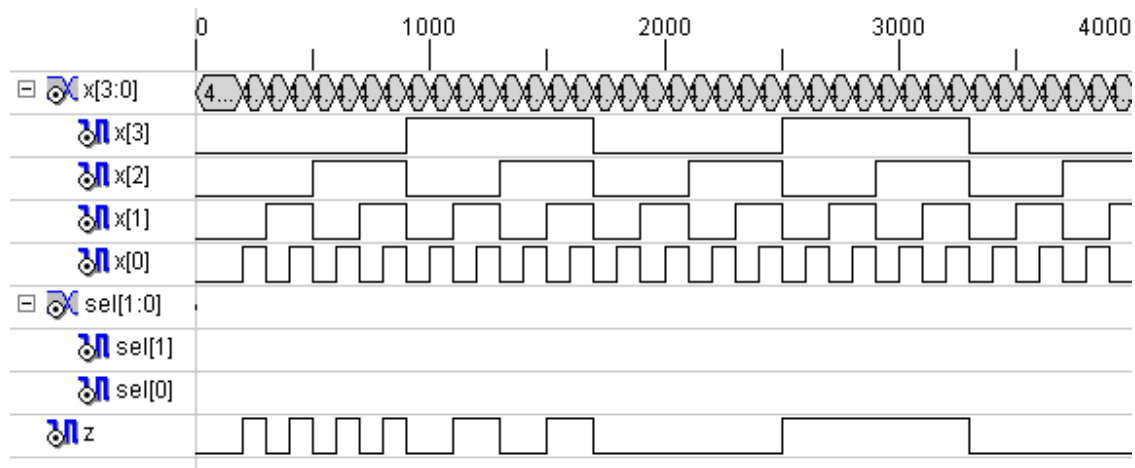


Figura 4.2: Simulación de un multiplexor de 4 a 1.

**Llame al instructor para registrar el avance. (10 puntos).**

#### PROBLEMA 4.2

Considere el siguiente código en VHDL:

```
ENTITY encoder3a2p IS
    Port ( entrada : in  STD_LOGIC_VECTOR (2 DOWNTO 0);
          salida : OUT STD_LOGIC_VECTOR (1 DOWNTO 0));
END encoder3a2p;

ARCHITECTURE Behavioral OF encoder3a2p IS
BEGIN
    salida <= "01" WHEN entrada(0)='1' ELSE
              "10" WHEN entrada(1)='1' ELSE
              "11" WHEN entrada(2)='1' ELSE
              "00";
END Behavioral;
```

a) Complete las formas de onda del puerto de salida en la Figura 4.3.

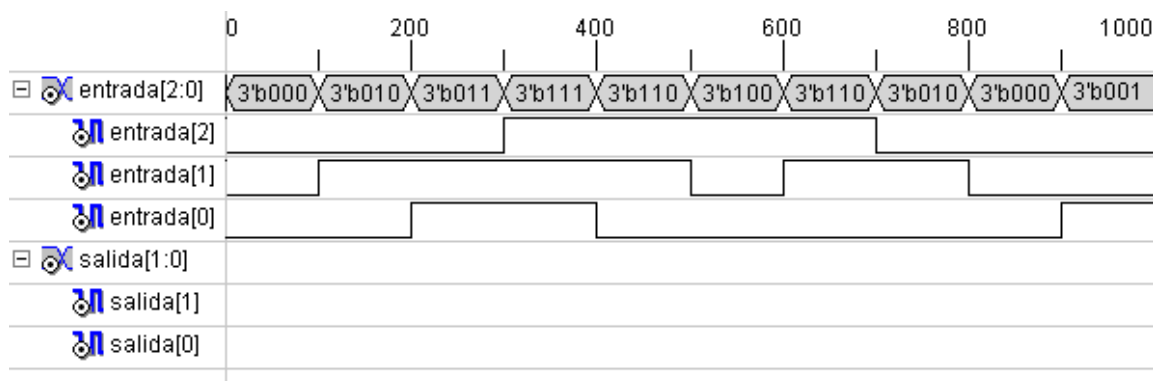


Figura 4.3: Simulación de un encoder 3 a 2 con prioridad.

- Escriba en lenguaje natural el funcionamiento la instrucción WHEN-ELSE.
- Escriba la tabla de verdad que relaciona las entradas con las salidas de este diseño.

### **Llame al instructor para registrar el avance. (10 puntos).**

#### **Asignación condicional: WITH-SELECT-WHEN**

Es una ampliación del condicional y es similar a las construcciones SWITCH-CASE del lenguaje C. Se utiliza para asignar un valor a una señal o puerto de salida (*salida*), dependiendo del contenido de cierto objeto (*expresion*). La sintaxis es:

WITH <i>expresion</i> SELECT	-- Dependiendo del valor de <i>expresion</i>
<i>salida</i> <= <i>valor1</i> WHEN <i>opcion1</i> ,	-- A <i>salida</i> se le asigna <i>valor1</i> cuando
	-- <i>expresion</i> = <i>opcion1</i> .
<i>valor2</i> WHEN <i>opcion2</i>   <i>opcion3</i>	-- A <i>salida</i> se le asigna <i>valor2</i> cuando
	-- <i>expresion</i> sea igual a <i>opcion2</i> ó a <i>opcion3</i> .
⋮	⋮
<i>valorK</i> WHEN <i>opcionL</i> TO <i>opcionM</i>	-- A <i>salida</i> se le asigna el valor <i>valorK</i> cuando
	-- <i>expresion</i> tenga un valor en el rango entre
	-- <i>opcionL</i> TO <i>opcionM</i> .
⋮	⋮
<i>valorN</i> WHEN OTHERS;	-- A <i>salida</i> se le asigna <i>valorN</i> cuando
	-- <i>expresion</i> no sea igual a las anteriores
	-- opciones.

Se tiene la opción de asignar el mismo valor (*valor2*), cuando el objeto que se evalúa (*expresion*), tiene alguno de varios valores (*opcion2* u *opcion3*), usando el operador “ | ” para delimitar las opciones para la expresión. También es posible especificar un rango de valores después de la cláusula WHEN, con lo cual lograremos que *salida* tenga un mismo valor cuando *expresion* esté dentro del rango especificado. Es obligatorio, incluir todos los posibles valores que pueda tomar la expresión. Por lo tanto, si no se especifican todos los valores en las cláusulas WHEN, entonces hay que incluir la cláusula WHEN OTHERS al final de la sentencia. A diferencia de la estructura WHEN-ELSE, la asignación WITH-SELECT-WHEN no establece ningún tipo de prioridad, en el orden que se introducen los valores, debido a que la salida es asignada dependiendo únicamente del objeto *expresion*.

A continuación se presenta el diseño de un multiplexor de 2 a 1, usando la asignación condicional WITH-SELECT-WHEN:

```

ENTITY mux2a1 IS
  PORT ( A, B, SEL: IN STD_LOGIC;
         Z: OUT STD_LOGIC);
END mux2a1;

ARCHITECTURE Behavioral OF mux2a1 IS
BEGIN
  WITH SEL SELECT
    Z <= A WHEN '0',          -- Dependiendo del valor de SEL
        B WHEN OTHERS;       -- Al puerto Z se le asigna el valor del puerto A, cuando SEL='0'
                                -- Al puerto Z se le asigna el valor del puerto B, si no se cumple
                                -- lo anterior (SEL='1').
END Behavioral;

```

El siguiente código es un decodificador de 2 a 4 líneas, usando la asignación concurrente WITH-SELECT-WHEN, su funcionamiento es idéntico al presentado con la instrucción WHEN-ELSE, en esta misma actividad.

```

ENTITY dec2a4wsw IS
  PORT(sel: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
        salida: OUT STD_LOGIC_VECTOR (3 DOWNTO 0));
END dec2a4wsw;

ARCHITECTURE behavioral OF dec2a4wsw IS
BEGIN
  WITH SEL SELECT
    salida <= "0001" WHEN "00",      -- salida es igual a "0001" cuando sel="00"
        "0010" WHEN "01",          -- salida es igual a "0010" cuando sel="01"
        "0100" WHEN "10",          -- salida es igual a "0100" cuando sel="10"
        "1000" WHEN OTHERS;        -- salida es igual a "1000" cuando sel tenga un valor
                                    -- diferente a los anteriores (sel="11").
END behavioral;

```

### PROBLEMA 4.3

a) Reescriba el diseño del problema 4.1, pero usando la instrucción WITH-SELECT-WHEN.

b) Escriba en lenguaje natural el funcionamiento la instrucción WITH-SELECT-WHEN del programa diseñado.

**Llame al instructor para registrar el avance. (10 puntos).**

#### PROBLEMA 4.4

Considere el siguiente código en VHDL:

```
ENTITY encoder3a2 IS
  Port ( entrada : in  STD_LOGIC_VECTOR (2 DOWNTO 0);
        salida : OUT STD_LOGIC_VECTOR (1 DOWNTO 0));
END encoder3a2;

ARCHITECTURE Behavioral OF encoder3a2 IS
BEGIN
  WITH entrada SELECT
    salida <= "01" WHEN "001",
              "10" WHEN "010",
              "11" WHEN "100",
              "00" WHEN OTHERS;
END Behavioral;
```

a) Complete las formas de onda del puerto de salida en la Figura 4.4.

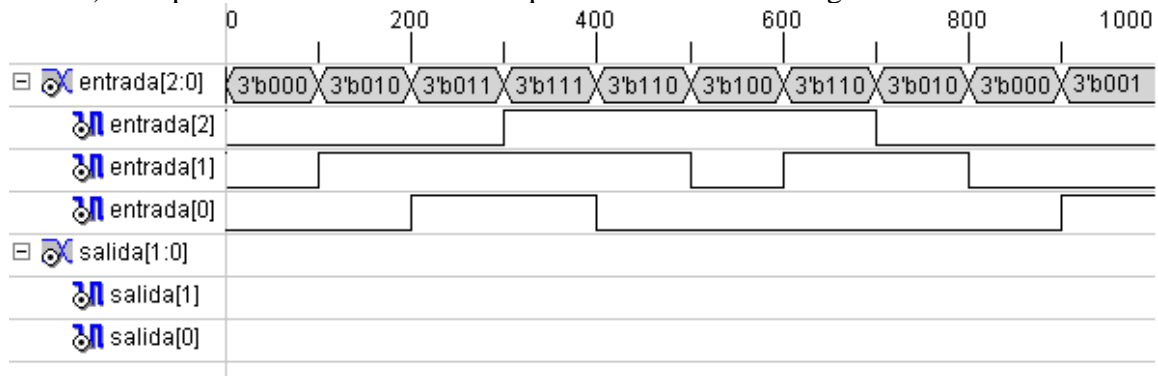


Figura 4.4: Simulación de un encoder 3 a 2 sin prioridad.

- b) Escriba en lenguaje natural el funcionamiento la instrucción WITH-SELECT-WHEN.
- c) Escriba la tabla de verdad que relaciona las entradas con las salidas de este diseño.  
**Llame al instructor para registrar el avance. (20 puntos).**

#### PROBLEMA 4.5

- b) Simule los códigos de los problemas 4.2 y 4.4 con el software ISE. Compare los resultados de ambas simulaciones y escriba las diferencias y similitudes.
- c) Implemente los diseños de los problemas 4.2 y 4.4 en el módulo BASYS 2, conectando las salidas a los LED, pero considerando al LED encendido como un '1' lógico y al LED apagado como un '0' lógico.

**Llame al instructor para registrar el avance. (20 puntos).**