

Diseño Digital

2024-2

Prof. Alonso Sanchez Huapaya

Preguntas

Sobre la clase pasada...

- ¿Cuáles librerías/paquetes importamos al inicio de nuestros diseños en VHDL?
- ¿Cuáles son las 3 partes de todo código en VHDL?
- ¿Cuáles son las 2 secciones de una arquitectura y para qué sirve cada una?

Temario

- Estructura de código VHDL
- Objetos, sentencias y circuitos básicos en VHDL
- Circuitos combinacionales en VHDL

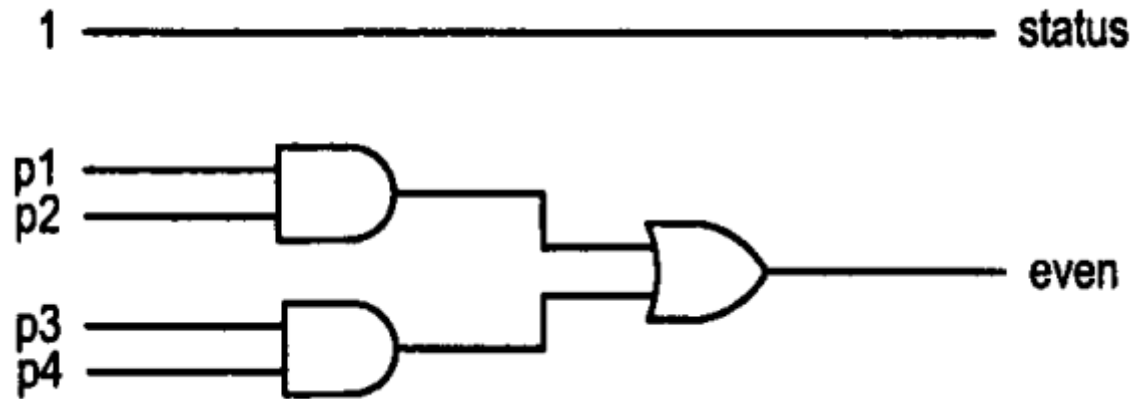
Estructura de un código VHDL

- Estructura básica:
 - Declarar librerías: std_logic_1164 , numeric_std
 - Declarar entidad
 - Declarar arquitectura
- Entidad:
 - Declarar “ports” (in ó out) con tipo de dato (std_logic → 1 bit // std_logic_vector → varios bits)
- Arquitectura:
 - Sección “concurrente”
 - Sección secuencial

Objetos, sentencias y circuitos básicos en VHDL

Sentencias booleanas

```
status <= '1';  
even <= (p1 and p2) or (p3 and p4);
```



Objetos, sentencias y circuitos básicos en VHDL

Operador de asignación: "<="

```
even <= (p1 and p2) or (p3 and p4);
```

DESTINO:

- Objeto salida ("out") o signal
- No puede ser una expresión (debe ser un único objeto).

ORIGEN:

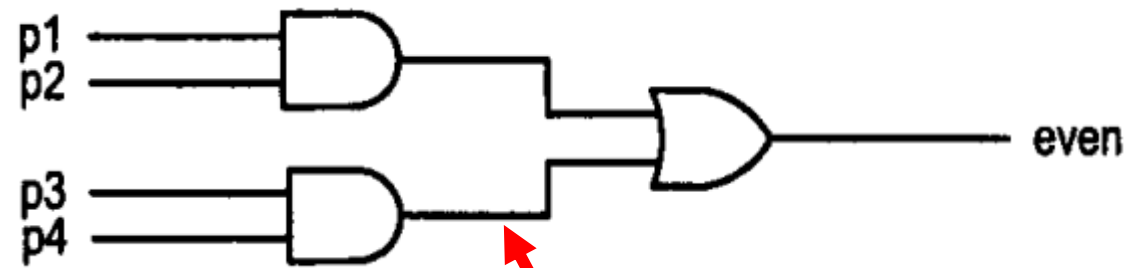
- Puede ser una expresión.
- Objetos deben ser entradas ("in") y/o signals

- Ambos lados del operador deben ser del mismo tipo de dato.
- Ambos lados del operador deben tener la misma cantidad de bits.

Objetos, sentencias y circuitos básicos en VHDL

Objeto: “signal”

```
even <= (p1 and p2) or (p3 and p4);
```



Nodo interno de circuito... signal!

Objetos, sentencias y circuitos básicos en VHDL

Sentencias condicionales (lo veremos con codificadores en breve)

```
signal_name <= value_expr_1 when boolean_expr_1 else  
               value_expr_2 when boolean_expr_2 else  
               value_expr_3 when boolean_expr_3 else  
               . . .  
               value_expr_n;
```

Objetos, sentencias y circuitos básicos en VHDL

Sentencias selectoras (lo veremos con el MUX en breve)

```
with select_expression select
    signal_name <= value_expr_1 when choice_1,
                  value_expr_2 when choice_2,
                  value_expr_3 when choice_3,
                  . . .
                  value_expr_n when choice_n;
```

Objetos, sentencias y circuitos básicos en VHDL

Objeto: vector de bits (std_logic, signed ó unsigned)

```
-- seccion declaracion de ENTIDAD (entity)
entity lab2 is
  port
  (
    -- Input ports
    A,B,C : in  std_logic; -- 3 bits de entrada
    bits_in : in std_logic_vector( 2 downto 0 ); -- 1 bus de datos de 3 bits de ancho

    -- Output ports
    X,Y,Z, o : out std_logic);
end lab2;
```

Objetos, sentencias y circuitos básicos en VHDL

La operación “&” (concatenar)

```
-- seccion declaracion de ENTIDAD (entity)
entity lab2 is
  port
  (
    -- Input ports
    A,B,C : in  std_logic; -- 3 bits de entrada
    bits_in : in std_logic_vector( 2 downto 0 ); -- 1 bus de datos de 3 bits de ancho

    -- Output ports
    X,Y,Z, o : out std_logic);
end lab2;
```

¿Qué pasa si quiero formar un vector de bits así?:

bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
A	A	B	B	bits_in			bits_in		

Objetos, sentencias y circuitos básicos en VHDL

La operación “&” (concatenar)

bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
A	A	B	B	bits_in			bits_in		

```
-- seccion declaracion de ENTIDAD (entity)
entity lab2 is
  port
  (
    -- Input ports
    A,B,C : in  std_logic; -- 3 bits de entrada
    bits_in : in std_logic_vector( 2 downto 0 ); -- 1 bus de datos de 3 bits de ancho

    -- Output ports
    X,Y,Z, o : out std_logic;
    concat_out : out std_logic_vector (9 downto 0)
  );
end lab2;

-- seccion declaracion de ARQUITECTURA (architecture)
architecture f_logicas of lab2 is
begin
  concat_out <= A & A & B & B & bits_in & bits_in;
```

Circuitos combinacionales en VHDL

Circuitos combinacionales en VHDL

Codificador

- Ckto combinacional, ingresan 2^n bits, salen n bits.
- $n = 3$. Entradas 'y', salidas 'b'

y7	y6	y5	y4	y3	y2	y1	y0		b2	b1	b0
0	0	0	0	0	0	0	1		0	0	0
0	0	0	0	0	0	1	0		0	0	1
0	0	0	0	0	1	0	0		0	1	0
0	0	0	0	1	0	0	0		0	1	1
0	0	0	1	0	0	0	0		1	0	0
0	0	1	0	0	0	0	0		1	0	1
0	1	0	0	0	0	0	0		1	1	0
1	0	0	0	0	0	0	0		1	1	1

¿Cómo es su diagrama RTL?

Circuitos combinacionales en VHDL

Decodificador

- Decodificador: Ckto combinacional, ingresan n bits, salen $m \leq 2^n$ bits
- Ejm: 3 a 8. Entradas 'b', salidas 'y'

b2	b1	b0	y7	y6	y5	y4	y3	y2	y1	y0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

¿Cómo es su diagrama RTL?

Circuitos combinacionales en VHDL

- ¿Cómo implementarlos en VHDL?

Ejemplo: CODIFICADOR

Una forma:

- Sentencia “when ... else”

```
architecture cond_arch of decoder4 is  
begin  
    x <= "0001" when (s="00") else  
        "0010" when (s="01") else  
        "0100" when (s="10") else  
        "1000";  
end cond_arch;
```

Circuitos combinacionales en VHDL

- ¿Cómo implementarlos en VHDL?

Ejemplo: CODIFICADOR

Otra forma:

- Sentencia “with...select”

```
architecture sel_arch of decoder4 is  
begin  
    with s select  
        x <= "0001" when "00",  
            "0010" when "01",  
            "0100" when "10",  
            "1000" when others ;  
end sel_arch;
```

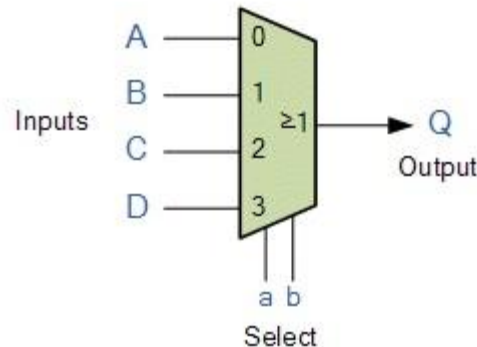
Circuitos combinacionales en VHDL

- Usos de decodificadores:
 - Conversión binario a 7 segmentos
 - Conversión BCD a 7 segmentos
 - Conversión binario a Gray
 - Implementación de funciones combinacionales arbitrarias
 - Etc...

Circuitos combinacionales en VHDL

Multiplexor

- Multiplexor: Ckto combinacional, con 2^n bits de entrada, 1 bit de salida, n bits de selección
- $n = 2$



¿Cómo es su diagrama RTL?

Circuitos combinacionales en VHDL

Multiplexor

- ¿Cómo implementarlo en VHDL?
- Usando sentencia “with...select”

```
architecture sel_arch of mux4 is
begin
    with s select
        x <= a when "00",
             b when "01",
             c when "10",
             d when others;
end sel_arch;
```

Circuitos combinacionales en VHDL

- Comparador de magnitud

¿Cómo es su diagrama RTL?

- Sumador / restador (lo veremos más adelante, requiere usar tipo de dato “numeric”)

¿Cómo es su diagrama RTL?

Circuitos combinacionales en VHDL

- Problema 4.1. del libro de Pong Chu (RTL y VHDL)

4.1 Add an enable signal, `en`, to a 2-to-4 decoder. When `en` is '1', the decoder functions as usual. When `en` is '0', the decoder is disabled and output becomes "0000". Use the conditional signal assignment statement to derive this circuit. Draw the conceptual diagram.

- Problema 4.6. del libro de Pong Chu (RTL)

4.6 We wish to design a shift-left circuit manually. The inputs include `a`, which is an 8-bit signal to be shifted, and `ctrl`, which is a 3-bit signal specifying the amount to be shifted. Both are with the `std_logic_vector` data type. The output `y` is an 8-bit signal with the `std_logic_vector` data type. Use concurrent signal assignment statements to derive the circuit and draw the conceptual diagram.