

LENGUAJES DE DESCRIPCION DE HARDWARE (HDL)

- Utilizados para describir y modelar hardware

Descripción

```
if print request then
...
end if
...
```

- Descripción

- $Z \leq a$ and b

1

Lenguaje de descripción de hardware

- Un HDL es útil para:
- Característica importante:

Modelar
Simular
Sintetizar

El hardware descrito es reutilizable

2

HDL y flujo de diseño

VHDL comportamental	▪Especificación
	▪Diseño de sistema
	▪Validación
VHDL RTL	▪Diseño lógico
	▪Validación
VHDL netlist	▪Diseño de circuito
	▪Validación
	▪Layout
	▪Validación

3

Conceptos generales

- **Very High Speed Integrated Circuit Hardware
Description Language**
 - Modelado de sistemas digitales
 - Sentencias secuenciales y concurrentes
 - La documentación puede ser leída por máquinas y humanos
- **Estándares internacionales**
 - IEEE STD 1076-1987
 - IEEE STD 1076-1993
 - IEEE STD 1076-1999

4

Historia

- Comienzos de 70's: Discusiones iniciales
- Finales de 70's: Definición de requerimientos
- Mediados '82: Contrato de desarrollo
- Mediados '84: Versión 7.2
- Mediados '86: Estándar IEEE
- 1987: DoD adopta el IEEE 1076
- Mediados '88: Se incrementa el soporte en herramientas CAE
- Finales '91: Revisión
- 1993: Nuevo estándar
- 1999: Extensión VHDL-AMS

5

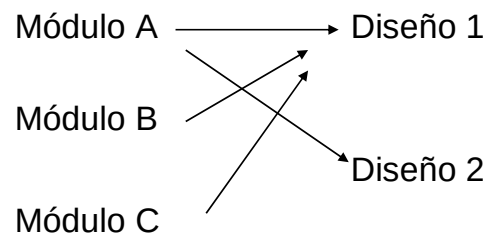
Conceptos generales

- Ejecución del código
 - Secuencial
 - Concurrente
- Metodologías
 - Abstracción
 - Modularidad
 - Jerarquía
- Abstracción:
 - Se debe diferenciar entre la información esencial y no esencial
 - En cada nivel de abstracción se toma sólo la información esencial

6

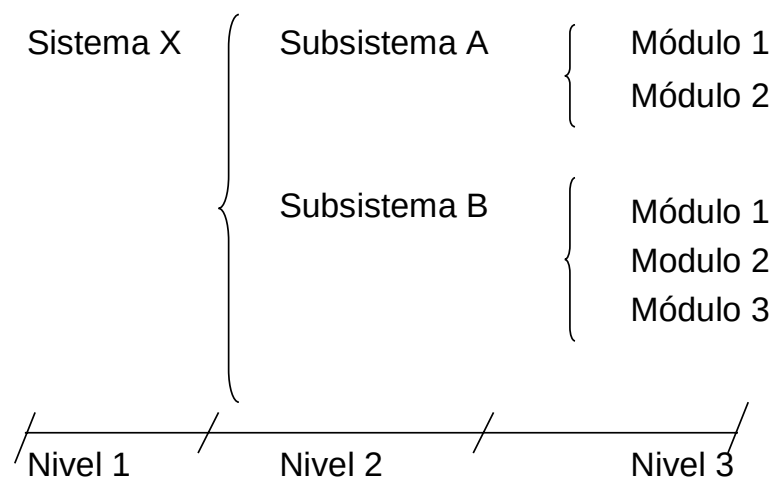
Conceptos generales

- Modularidad



7

Jerarquía



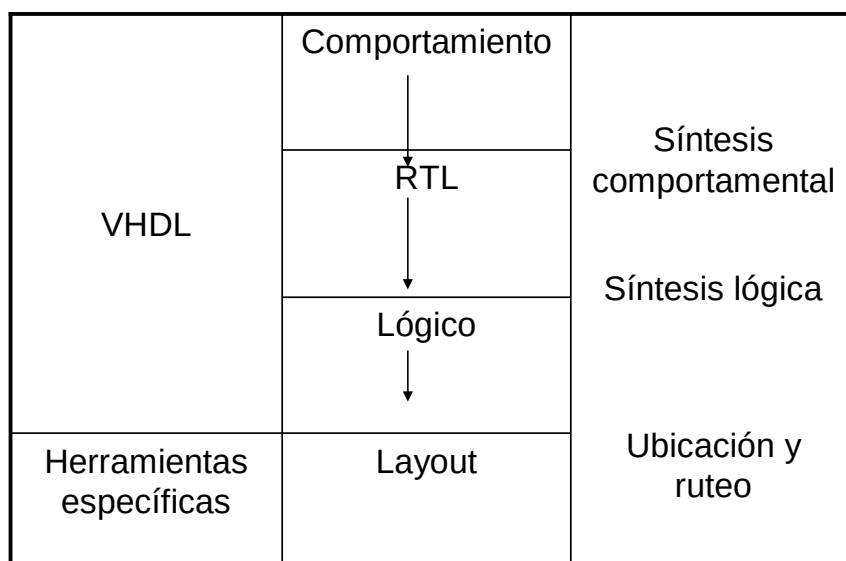
8

Niveles de abstracción

- Comportamiento
- RTL
- Lógico
- Layout

9

VHDL y niveles de abstracción



10

Elementos estructurales del VHDL

- Entity

Declara un componente o sistema

Señales de conexión con el mundo externo

No contiene definiciones de comportamiento

- Ejemplos

```
entity HALFADDER is
port ( A, B: in bit ;
      SUM, CARRY : out bit ) ;
end HALFADDER ;

entity ADDER is
port ( A, B: in integer range 0 to 3 ;
      SUM: out integer range 0 to 3;
      CARRY: out bit) ;
end ADDER ;
```

11

Elementos estructurales del VHDL

- Architecture

La implementación del diseño se realiza dentro de la arquitectura

Debe estar siempre conectada a una entidad específica

La implementación puede hacerse en cualquier nivel de abstracción (comportamental, RTL o netlist)

```
ENTITY halfadder IS
PORT (A, B : IN bit;
      SUM,CARRY : OUT BIT);
END halfadder;
```

```
ARCHITECTURE rtl OF halfadder IS
BEGIN
  SUM <= A xor B;
  CARRY <= A and B;
END rtl;
```

12

Elementos estructurales: ARCHITECTURE

```
architecture EXAMPLE of STRUCTURE is
  subtype DIGIT is integer range 0 to 9;
  constant BASE: integer:=10;
  signal DIGIT_A, DIGIT_B: DIGIT;
  signal CARRY: DIGIT;
begin
  DIGIT_A<=3;
  SUM<=DIGIT_A+DIGIT_B;
  DIGIT_B<=7;
  CARRY <= 0   when SUM<BASE else 1;
end EXAMPLE;
```

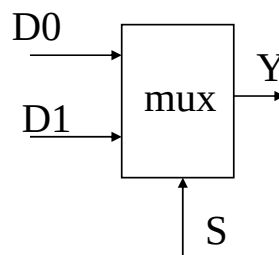
13

EJEMPLO:

- Multiplexor 2 a 1

```
ENTITY mux IS
  PORT (D0,D1,S : IN BIT;
        Y : OUT BIT);
END mux;
```

```
ARCHITECTURE arch1 OF mux IS
  BEGIN
    Y<= D0 WHEN S='0';
    ELSE D1;
  END arch1;
```



Y	S
D0	0
D1	1

$$Y = \bar{S}.D0 + S.D1$$

14

PAQUETES

- Permite agrupar una serie de declaraciones para que puedan ser utilizadas por varios dispositivos sin ser declaradas nuevamente para cada uno de ellos.
- Normalmente se declaran: constantes, tipos y subtipos de datos, subprogramas y componentes.
- Posee dos unidades diferenciadas: declaración y cuerpo.
- Los resultados de compilación son almacenados en una biblioteca para ser utilizados posteriormente.

15

Ejemplo:

- Se definen en este paquete los retardos de tiempo para los operadores not, and y or

```
package retardosop is
  constant retnot: time ;
  constant retand: time ;
  constant retor : time ;
end retardosop ;

package body retardosop is
  constant retnot: time := 1 ns ;
  constant retand: time := 2 ns ;
  constant retor : time := 2 ns ;
end retardosop ;
```

16

EJEMPLO

- Si se supone que el resultado de compilación es guardado en la biblioteca “biblio” y se pretende utilizarlo en una descripción VHDL, se podrá hacer:

```
architecture flujodatos of mux2 is
signal ctrl_n, n1 , n2 : bit ;
begin
    signal ctrl_n <= not (ctrl) after
biblio.retardosop.retnot;
    n1 <= ctrl_n and a after biblio.retardosop.retand ;
    n2 <= ctrl_n and b after biblio.retardosop.retand ;
    z <= n1 or n2 after biblio.retardosop.retor;
end flujodatos ;
```

17

EJEMPLO

- Para no complicar tanto la descripción puede hacerse uso de las siguientes cláusulas:

```
use biblio.retardosop.retnot
```

- si se quiere utilizar solo retnot.

```
use biblio.retardosop.all
```

- si se quieren utilizar todos.

18

Ejemplo:

```
architecture flujodatos of mux2 is
signal ctrl_n, n1 , n2 : bit ;
begin
    signal ctrl_n <= not (ctrl) after
    retnot ;
    n1 <= ctrl_n and a after retand ;
    n2 <= ctrl_n and b after retand ;
    z <= n1 or n2 after retor ;
end flujodatos ;
```

19

BIBLIOTECAS

- Sirven para almacenar el resultado de la compilación de unidades de diseño para su utilización posterior.
- Facilitan la reutilización del código en los diferentes diseños.
- La biblioteca “work” es la biblioteca por defecto. El diseñador puede crear otras.
- Desde un modelo almacenado en una biblioteca no se puede acceder a las unidades de diseño de otras bibliotecas ya que se tiene visibilidad sólo de la biblioteca en la cual está almacenado el modelo.
- Para dar visibilidad a una biblioteca se utiliza la sentencia library. Las bibliotecas work y std son excepciones y están siempre visibles.

20

EJEMPLO

```
library bibliotecaejemplo ;  
use bibliotecaejemplo.paqueteejemplo.all
```

- En este caso se pueden utilizar los elementos de un paquete denominado “paqueteejemplo” que está almacenado en una biblioteca llamada “bibliotecaejemplo”.

21

PARA DESTACAR:

- La definición de biblioteca es lógica. Cada herramienta puede implementarla como quiera sobre el sistema de archivos. En algunos casos una biblioteca será un archivo, en otros un directorio o una estructura jerárquica de directorios.

22