



VDHL

ARELLANO GRANADOS ANGEL MARIANO 218123444

SEMINARIO DE ARQUITECTURA DE COMPUTADORAS

2021B



TIPOS DE DESCRIPCIONES

ESTRUCTURAL

Esta descripción utiliza para la creación de la arquitectura de la entidad entidades descritas y compiladas previamente.

De esta manera en VHDL podemos aprovechar diseños ya realizados, o realizar diseños sabiendo que se utilizarán en otros más complicados. Así se ahorra trabajo al diseñador-programador.

SECUENCIAL

Una de las propiedades más importantes de un process es la capacidad de la estructura para almacenar los valores de las señales que se asignan en su interior si durante el paso de simulación no se entra dentro del process o no se realiza ninguna asignación a esa señal.

Debido a esta característica se utilizarán los process para generar HW secuencial.

ESTRUCTURA DE UNA DESCRIPCION

Los modelos VHDL están formados por dos partes: la entidad (entity) y la arquitectura (architecture); es en esta última donde se escriben las sentencias que describen el comportamiento del circuito, a este modelo de programación en VHDL se le suele denominar behavioral.

```
architecture circuito of nombre is
-- señales
begin
-- sentencias concurrentes
process (lista de sensibilidad)
begin
-- sentencias secuenciales
-- sentencias condicionales
end process
end architecture circuito;
```



i)

Tipos y señales intermedias necesarios para la descripción del comportamiento.



ii)

Sentencias de asignación que deben realizarse siempre así como sentencias concurrentes.



iii)

Uno a varios process que tienen en su interior sentencias condicionales y/o asignaciones a señales que dan lugar a hardware secuencial.

PUERTOS Y SEÑALES

¡PUERTOS

Los puertos pueden ser de entrada *in*, salida *out*, entrada-salida *inout* o *buffer*. Los puertos de entrada sólo se pueden leer y no se puede modificar su valor internamente en la descripción del comportamiento del circuito (architecture), sobre los puertos de salida sólo se puede escribir pero nunca tomar decisiones dependiendo de su valor.

```
entity nombre is
    [generic (lista de parámetros);]
    [port (lista de puertos);]
    [declaraciones]
[begin
    sentencias]
end [entity] [nombre];
```

PUERTOS Y SEÑALES



SIGNAL

Las señales representan elementos de memoria o conexiones y sí pueden ser sintetizados, dicho de otra manera, a cada objeto de nuestro código VHDL que sea declarado como signal le corresponde un cable o un elemento de memoria (biestable, registro ...) en nuestro circuito.

Los puertos de una entidad son implícitamente declarados como señales en el momento de la declaración.

- ***signal identificador: tipo;***

La asignación de una señal a un valor se hace mediante el operador \leq :

nombre señal \leq valor o expresión;

A \leq 10;

PROCEDIMIENTOS

VDHL

- Pensar en puertas y biestables, no en variables ni funciones.
- Evitar bucles combinacionales y relojes condicionados.
- Saber qué parte del circuito es combinacional y cuál secuencial.
- Poder descubrir problemas en el diseño antes de su implementación física.
- La complejidad de los sistemas electrónicos crece exponencialmente, es necesaria una herramienta que trabaje con el ordenador.
- Permite que más de una persona trabaje en el mismo proyecto.



PROCEDIMIENTOS



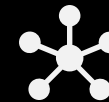
**Pequeñas
descripciones
de "circuitos".**



**Posibilidad de
intercambio
bidireccional
con el exterior**



**Posibilidad de
incluir una
sentencia wait**



**Posibilidad de
efectuar
asignaciones a
señales**



**Se define en la
zona de
declaraciones
de la
arquitectura**

COCURRENCIA

SENTENCIAS CONCURRENTES:

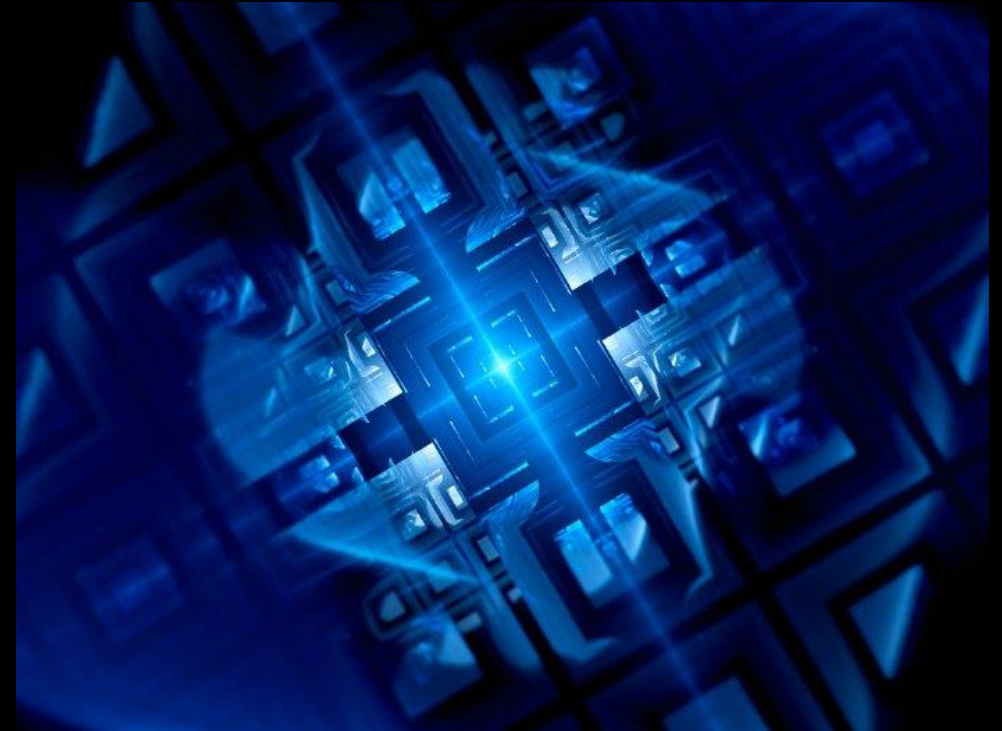
Las sentencias concurrentes son sentencias condicionales que tienen al menos un valor por defecto para cuando no se cumplen ninguna de las condiciones. Aunque podría utilizarse una sentencia común como un if con obligación de else, los desarrolladores de VHDL han preferido utilizar dos sentencias particulares:

WHEN – ELSE

```
señal_a_modificar <= valor_1 when condición_1 else
                      valor_2 when condición_2 else
                      ...
                      valor_n when condición_n else
                      valor_por defecto;
```

WITH – SELECT – WHEN

```
with señal_condición select
señal_a_modificar <= valor_1 when valor_1_señal_condición,
                      valor_2 when valor_2_señal_condición,
                      ...
                      valor_n when valor_n_señal_condición,
                      valor_por_defecto when others;
```



BIBLIOGRAFIA:

- Sánchez-Élez, M. (N/A). INTRODUCCIÓN A LA PROGRAMACIÓN EN VHDL. Facultad de Informática Universidad Complutense de Madrid: F. Informática (UCM)
- Mora Campos, A. (2016). GUÍA BÁSICA DEL VHDL. Instituto Tecnológico de Querétaro: ITQ