



TEMA 6

Introducción al diseño digital con lenguajes de descripción de hardware

Electrónica Digital
Grado Ing. Tecnologías
Telecomunicación



Metodologías de diseño de ICs

- Un mismo sistema puede ser descrito utilizando diferentes dominios de descripción, incluso varios simultáneamente:
 - ✓ **descripción comportamental (*behavioral*):** especifica o describe el algoritmo que realiza el sistema
 - ✓ **descripción estructural (*structural*):** especifica los componentes necesarios para formar el sistema y la manera en que han de interconectarse
 - ✓ **descripción física:** especifica la disposición física en la que se han de situar los componentes anteriores
- Dependiendo de la tecnología objetiva y del flujo de diseño, alguno de los niveles se ven modificados o quedan ocultos.

Lenguajes de descripción de hardware

- A medida que aumenta la complejidad de los sistemas integrados, la descripción puramente estructural se hace inabordable:
 - ✓ es necesario trasladar a esquemáticos las estructuras lógicas, proceso laborioso y **propenso a errores**
 - ✓ estructuras complejas son difícilmente abordables sin herramientas de síntesis automática
 - ✓ es imprescindible verificar mediante simulación el correcto funcionamiento del sistema durante el diseño
- El aumento del nivel de **abstracción** y la **jerarquización** en la descripción de los sistemas integrados requiere el uso de medios de descripción unificados.

Electrónica Digital – Introducción a VHDL

Lenguajes de descripción de hardware

- Los lenguajes de descripción de hardware (HDL: *Hardware Description Language*) permiten:
 - ✓ describir sistemas en un nivel muy elevado de abstracción
 - ✓ facilitar la simulación y verificación del sistema durante el diseño
 - ✓ crear descripciones portables e independientes de la tecnología de implementación
- La síntesis basada en descripciones HDL produce circuitos menos densos y más lentos que los resultantes del diseño *full-custom* por diseñadores expertos:
 - ✓ el estado de desarrollo tecnológico permite que estas implementaciones basten para la mayoría de ASICs actuales

Electrónica Digital – Introducción a VHDL

Lenguajes de descripción de hardware

- A pesar de su similitud con los lenguajes de programación, existe una diferencia fundamental:
 - ✓ en *software* la ejecución del algoritmo descrito es secuencial
 - ✓ en *hardware*, un circuito está formado por bloques que funcionan simultáneamente
- Su uso, especialmente para síntesis, depende de las herramientas a utilizar, con lo que es preciso describir el sistema teniendo en mente el tipo de circuito que se va a generar:
 - ✓ pueden obtenerse resultados no deseados o incorrectos (introducción de registros y *latches*, imposibilidad de síntesis, etc.)
 - ✓ el resultado puede ocupar mucha más área de la requerida o ser excesivamente lento

Electrónica Digital – Introducción a VHDL

Lenguajes de descripción de hardware

- Algunos de los lenguajes más usados son:
 - ✓ Verilog (IEEE1364-01):
 - desarrollado en 1984 por Gateway Design Automation para simulación lógica
 - convertido en estándar abierto en 1990 tras la compra de Gateway por Cadence
 - compacto y de sintaxis similar a C
 - ✓ VHDL (IEEE1076-02, *Very high-speed integrated circuit* HDL):
 - desarrollado en 1981 por el Departamento de Defensa y estandarizado por IEEE en 1987
 - menos compacto y más verboso que Verilog
 - adecuado para grandes equipos y proyectos

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

- La descripción con VHDL se basa en la definición de unidades de código que se agrupan en diferentes ficheros.
- Los tipos de unidades posibles en VHDL son:
 - ✓ **entidad** (`entity`): describe la interfaz con el exterior de un elemento
 - equivale al símbolo de un elemento en un esquemático
 - ✓ **arquitectura** (`architecture`): describe la estructura o el comportamiento interno de una entidad
 - es posible definir diferentes arquitecturas para cada entidad
 - incluye la declaración de componentes, señales y variables internas, etc.

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

- La descripción con VHDL se basa en la definición de unidades de código que se agrupan en diferentes ficheros.
- Los tipos de unidades posibles en VHDL son:
 - ✓ **configuración** (`configuration`): especifica la arquitectura, si existen varias, que se asocia en cada momento a una entidad dependiendo de ciertas condiciones (simulación, síntesis, etc.)
 - ✓ **paquete** (`package`): incluye un conjunto de declaraciones (tipos, subtipos, etc.) compartidos por varias unidades de diseño
 - ✓ **cuerpo de paquete** (`package body`): define los elementos declarados en el paquete correspondiente
 - existen paquetes predefinidos y el usuario puede crear otros

Electrónica Digital – Introducción a VHDL

Lenguajes de descripción de hardware

- Los principales paquetes predefinidos son:
 - ✓ `std.standard`: define los tipos básicos de VHDL como `boolean`, `bit`, `bit_vector`, `character`, `string`, `integer`, `real`, `time`, etc.
 - ✓ `std.textio`: define los tipos `line` y `text` y procedimientos de lectura y escritura en ficheros
 - ✓ `IEEE.std_logic_1164`: define los tipos `std_logic` y `std_logic_vector`
 - ✓ `IEEE.std_logic_arith`: define los tipos `signed` y `unsigned` y operaciones aritméticas con estos tipos
 - ✓ `IEEE.std_logic_signed`: define operaciones aritméticas con el tipo `std_logic_vector` considerado codificado con signo

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

- Los principales paquetes predefinidos son:
 - ✓ `IEEE.std_logic_unsigned`: define operaciones aritméticas con el tipo `std_logic_vector` considerado codificado sin signo
 - ✓ `IEEE.std_logic_textio`: define procedimientos de lectura y escritura en ficheros para los tipos `std_logic` y `std_logic_vector`.
- VHDL es un lenguaje **fuertemente tipado**, a diferencia de muchos lenguajes de programación:
 - ✓ no es posible realizar asignaciones entre señales o variables de tipos diferentes
 - ✓ requiere funciones de conversión entre tipos

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

```
LIBRARY ieee; -- Declaracion librerias
USE ieee.std_logic_1164.ALL; -- Paquetes en uso de libreria
USE ieee.std_logic_arith.ALL;

ENTITY biestableD IS -- Declaracion de entidad
  PORT (
    D,clk : IN STD_LOGIC; -- Entradas
    q : OUT STD_LOGIC -- Salida
  );
END biestableD;

ARCHITECTURE proceso OF biestableD IS -- Declaracion de arquitectura
BEGIN -- Cuerpo de la arquitectura
  proceso_biestableD : PROCESS (clk) -- Lista de sensibilidad
  BEGIN
    IF clk'event AND CLK='1' THEN -- Flanco de subida de clk
      q <= d; -- Asignación concurrente
    END IF;
  END PROCESS proceso_biestableD; -- Fin del proceso
END proceso;
```

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

- Los puertos de las entidades pueden ser de cuatro tipos, atendiendo al tipo de E/S salida físicas:
 - ✓ **in**: puerto de entrada
 - sólo puede ser leído en el interior de la entidad
 - ✓ **out**: puerto de salida
 - la entidad sólo puede actualizar su valor, nunca leerlo
 - ✓ **buffer**: puerto de salida
 - la entidad puede actualizar su valor y leerlo internamente
 - puede cambiarse a **out** añadiendo una señal intermedia
 - ✓ **inout**: puerto de entrada/salida
 - la entidad puede actualizar su valor y leerlo (fijado externamente)
 - normalmente corresponde a una E/S triestado

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

- En una arquitectura se puede definir:
 - ✓ tipos (`type`) y constantes (`constant`): enumerados, físicos, compuestos, etc.
 - ✓ señales (`signal`): corresponden a las interconexiones en el interior de la entidad
- El cuerpo de la arquitectura de una entidad está compuesto por:
 - ✓ asignaciones concurrentes: se sintetizan como lógica combinacional
 - ✓ procesos: a través de la lista de sensibilidad permiten describir elementos secuenciales de un sistema
 - ✓ instanciación de componentes: permite estructurar jerárquicamente el diseño a través de una descripción estructural

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

- Asignaciones concurrentes:

```
comp <= '1' WHEN a=b ELSE '0';           -- Comparador
addsub <= a+b WHEN sel='1' ELSE a-b;     -- Sumador/restador
z <= x WHEN tri='0' ELSE 'Z';           -- Buffer triestado
```

```
SIGNAL code   : STD_LOGIC_VECTOR(1 DOWNTO 0);
SIGNAL decode : STD_LOGIC_VECTOR(3 DOWNTO 0);
...
WITH code SELECT
    decode <=
        "0001" WHEN "00",
        "0010" WHEN "01",
        "0100" WHEN "10",
        "1000" WHEN "11",
        "0000" WHEN OTHERS;
```

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

- Los procesos requieren:
 - ✓ lista de sensibilidad: señales que lo activan
 - ✓ sentencia `wait`: suspende la ejecución del proceso
 - `wait on` lista de sensibilidad
 - `wait until` expresión booleana
 - `wait for` expresión temporal
- Dentro de los procesos se pueden definir variables (`variable`):
 - ✓ se declaran y se asignan dentro del proceso
 - ✓ se evalúan en el momento en el que la sentencia de asignación se ejecuta y el resultado se asigna inmediatamente a la variable

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

- Los procesos permiten diferentes estructuras secuenciales:

```
WHILE condicion booleana LOOP
...                               -- Sentencias secuenciales
END LOOP;

FOR identificador IN rango LOOP
...                               -- Sentencias secuenciales
END LOOP;

LOOP
...                               -- Sentencias secuenciales
END LOOP;
```

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

- Los procesos permiten diferentes estructuras secuenciales:

```

IF condicion booleana THEN
...           -- Sentencias secuenciales
ELSIF condicion booleana THEN
...           -- Sentencias secuenciales
END IF;

CASE (señal,variable) IS
  WHEN valor1=>
...           -- Sentencias secuenciales
  WHEN valor2=>
...           -- Sentencias secuenciales
...
  WHEN OTHERS=>
...           -- Especificar todas las opciones
...           -- Sentencias secuenciales
END CASE;
  
```

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

- Palabras reservadas en VHDL-87:

ABS	CASE	GENERIC	NOR	REGISTER	VARIABLE
ACCESS	COMPONENT	GUARDED	NOT	REM	WAIT
AFTER	CONFIGURATION	IF	NULL	REPORT	WHEN
ALIAS	CONSTANT	IN	OF	RETURN	WHILE
ALL	DISCONNECT	INOUT	ON	SELECT	WITH
AND	DOWNTO	IS	OPEN	SEVERITY	XOR
ARCHITECTURE	ELSE	LABEL	OR	SIGNAL	
ARRAY	ELSIF	LIBRARY	OTHERS	SUBTYPE	
ASSERT	END	LINKAGE	OUT	THEN	
ATTRIBUTE	ENTITY	LOOP	PACKAGE	TO	
BEGIN	EXIT	MAP	PORT	TRANSPORT	
BLOCK	FILE	MOD	PROCEDURE	TYPE	
BODY	FOR	NAND	PROCESS	UNITS	
BUFFER	FUNCTION	NEW	RANGE	UNTIL	
BUS	GENERATE	NEXT	RECORD	USE	

- Palabras reservadas añadidas en VHDL-93:

GROUP	LITERAL	REJECT	SHARED	SRA	XNOR
IMPURE	POSTPONED	ROL	SLA	SRL	
INERTIAL	PURE	ROR	SLL	UNAFFECTED	

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

➤ Operadores lógicos:

and or nand nor xor xnor not

➤ Operadores de relación:

= /= < <= > >=

➤ Operadores de desplazamiento:

sll srl sla sra rol ror

➤ Operadores aditivos:

+ - &

➤ Operadores multiplicativos:

* / mod rem

➤ Otros operadores:

abs **

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

➤ **Ejemplo:** descripción estructural

```
ENTITY half_adder IS -- Declaracion de entidad
    PORT (
        a,b : IN BIT; -- Sumandos
        c,s : OUT BIT -- Suma y acarreo
    );
END half_adder;
ARCHITECTURE ed OF half_adder IS
    COMPONENT xor2 -- Puerta EXOR
        PORT ( x,y : IN BIT;
              z : OUT BIT);
    END COMPONENT;
    COMPONENT and2 -- Puerta AND
        PORT ( x,y : IN BIT;
              z : OUT BIT);
    END COMPONENT;
BEGIN
    C1: xor2 PORT MAP (x => a, y => b, z => s);-- s <= a XOR b
    C2: and2 PORT MAP (x => a, y => b, z => c);-- c <= a AND b
END ed;
```

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

> Ejemplo: *latch* y biestable con reset síncrono

```

ENTITY latch IS
  PORT (
    d,clk : IN BIT;
    q      : OUT BIT );
END latch;

ARCHITECTURE ed OF latch IS
BEGIN
  PROCESS (d,clk)
  BEGIN
    IF clk= '1' THEN
      q <= d;
    END IF;
  END PROCESS;
END ed;

ENTITY flip-flop IS
  PORT (
    d,rst,clk : IN BIT;
    q          : OUT BIT );
END flip-flop;

ARCHITECTURE ed OF flip-flop IS
BEGIN
  PROCESS (clk)
  BEGIN
    IF clk'event AND clk= '1' THEN
      IF rst= '0' THEN
        q <= '0';
      ELSE
        q <= d;
      END IF;
    END IF;
  END PROCESS;
END ed;
  
```

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

> Ejemplo: registro de 32 bits

```

ENTITY reg_32 IS
  PORT (
    d      : IN      STD_LOGIC_VECTOR(31 DOWNTO 0);
    q      : OUT     STD_LOGIC_VECTOR(31 DOWNTO 0);
    clk    : IN      STD_LOGIC;
  );
END reg_32;

ARCHITECTURE ed OF reg_32 IS
BEGIN
  PROCESS (clk)
  BEGIN
    IF clk'event AND clk= '1' THEN
      q <= d;
    END IF;
  END PROCESS;
END ed;
  
```

Electrónica Digital – Introducción a VHDL

Descripción de sistemas digitales: VHDL

> Ejemplo: máquina de estados finitos con reset asíncrono

```
ENTITY fsm IS
PORT (
    clk      : IN STD_LOGIC;
    rst      : IN STD_LOGIC;
    entrada  : IN STD_LOGIC;
    salida   : OUT STD_LOGIC
);
END fsm;

ARCHITECTURE ed OF fsm IS

TYPE estado IS (init,exec,idle);
SIGNAL estado_actual : estado;

BEGIN
    PROCESS (clk, rst)
    BEGIN
        IF rst = '1' THEN
            estado_actual <= init;
        ELSIF clk'event AND clk = '1' THEN
            CASE estado_actual IS
            WHEN init =>
                IF condicion THEN
                    estado_actual <= exec;
                END IF;
            WHEN exec =>
                IF condicion THEN
                    estado_actual <= idle;
                END IF;
            WHEN idle =>
                IF condicion THEN
                    estado_actual <= init;
                END IF;
            END CASE;
        END IF;
    END PROCESS;
END ed;
```