

# Operadores y atributos en VHDL [1]

Sistemas Digitales Avanzados

Universidad Técnica Particular de Loja

*Prof: Diego Barragán Guerrero*

Oct. 2014 - Feb. 2015

Operadores pre definidos:

- Operadores de asignación.
- Operadores lógicos.
- Operadores aritméticos.
- Operadores de relación.
- Operadores de desplazamiento.
- Operadores de concatenación.

# Operadores de asignación

- $\leq$  Asignar valor a *Signal*.
- $:=$  Asignar valor a *Variable*, *Constant* o *Generic*. Establece también el valor inicial.
- $=>$  Asignar valores a elementos de vectores con *Others*.

## Código VHDL

```
SIGNAL x: STD_LOGIC;  
VARIABLE y: STD_LOGIC_VECTOR(3 DOWNTD 0);  
SIGNAL w: STD_LOGIC_VECTOR(0 TO 7);  
  
x<='1'      -- Asignación a Signal.  
y:="0000"; -- Asignación a variable  
w<=(0=>'1', OTHERS =>'0'); -- Asignación a elementos de vectores
```

# Operadores lógicos

- Datos de entrada:
  - BIT, STD\_LOGIC o STD\_ULOGIC.
  - BIT\_VECTOR, STD\_LOGIC\_VECTOR o STD\_LOGIC\_VECTOR.
- Lista de operadores:
  - NOT, AND, OR, NAND, NOR, XOR, XNOR.

## Código VHDL

```
y<= NOT a AND b;    --(a'.b)
y<= NOT (a AND b);--(a.b) '
y<= a NAND b;      --(a.b) '

```

# Operadores aritméticos

- Datos de entrada: INTEGER, SIGNED o UNSIGNED.
- Con paquetes *std\_logic\_signed* *std\_logic\_unsigned* el dato *std\_logic\_vector* puede usarse para suma o resta.

Cuadro: Operadores aritméticos

+	Suma
-	Resta
*	Multiplicación
/	División
**	Potencia
MOD	Módulo
REM	Resto
ABS	Valor absoluto

# Operadores de comparación

- Datos de entrada: cualquier tipo de dato.

Cuadro: Operadores de comparación

=	Igual a
/=	Diferente de
<	Menor que
>	Mayor que
<=	Menor o igual
>=	Mayor o igual

# Operadores de desplazamiento

- Sintaxis: <Operador izquierdo><Operación><Operador derecho>
- Operador izquierdo: del tipo BIT\_VECTOR.
- Operador derecho: del tipo INTEGER.

Cuadro: Operadores de desplazamiento

sll	desplazamiento a la izquierda
srl	desplazamiento a la derecha

# Atributos de datos

- Todos estos atributos son sintetizables.

Cuadro: Atributos de datos

d'LOW	índice menor del arreglo
d'HIGH	índice superior del arreglo
d'LEFT	índice más a la izquierda
d'RIGHT	índice más a la derecha
d'LENGTH	dimensión del vector
d'RANGE	rango del vector
d'REVERSE_RANGE	rango del vector invertido

# Atributos de datos (ejemplos)

## Código VHDL

```
SIGNAL d: STD_LOGIC_VECTOR (7 DOWNTO 0);
```

Cuadro: Atributo de la señal *d*

d'LOW	0
d'HIGH	7
d'LEFT	7
d'RIGHT	0
d'LENGTH	8
d'RANGE	7 downto 0
d'REVERSE_RANGE	0 to 7

# Atributos de señales

- Considerando una señal  $s$ .

Cuadro: Atributos de señales

$s'$ EVENT	verdadero si un evento ocurre en $s$
$s'$ STABLE	verdadero si no hay eventos en $s$
$s'$ ACTIVE	verdadero si $s=1$
$s'$ QUIET	verdadero si no hay eventos en un tiempo determinado
$s'$ LAST_EVENT	tiempo desde del último evento
$s'$ LAST_ACIVE	valor de $s$ antes del último evento

- Solo las dos primeras son sintetizables.

# Atributos de señales (ejemplo)

- Todas las asignaciones a seguir son sintetizables y retornan un valor verdadero (true) cuando un evento ocurre en el reloj (clk) Y ese evento es hacia arriba.

## Código VHDL

```
IF (clk'EVENT AND clk='1')...
```

```
IF (NOT clk'STABLE AND clk='1')...
```

```
WAIT UNTIL (clk'EVENT AND clk='1');
```

```
IF RISING_EDGE(clk)... -- Llamado a una función
```

# Generic

- Parámetro estático definido para facilidad de modificación del código.
- Implementa flexibilidad al código.
- Es declarado en la entidad.
- Es global para todo el programa.

## Código VHDL

```
ENTITY my_entity IS
    GENERIC (n : INTEGER := 8);
    PORT (...);
END my_entity;
ARCHITECTURE my_architecture OF my_entity IS
    ...
END my_architecture;
-- Se puede especificar más de un Generic
GENERIC (n: INTEGER := 8; vector: BIT_VECTOR := "00001111");
```

- [1] Volnei A. Pedroni.  
*Circuit Design with VHDL.*  
MIT Press, Cambridge, MA, USA, 2004.