

SENTENCIAS SECUENCIALES

- Se ejecutan en el orden en que aparecen en la descripción, en forma similar a lo que ocurre en un lenguaje de programación.
- Están permitidas sólo dentro de PROCESS
- Se utilizan para describir algoritmos.

Lenguajes de Descripción de Hardware

LA SENTENCIA IF

- Se utiliza normalmente para controlar el flujo de ejecución de sentencias secuenciales.
- Para ello se evalúa si una condición se cumple o no (booleana: verdadero/falso).
- La estructura general es la siguiente:

```
if CONDICIÓN then
    --sentencias secuenciales
end if;

if CONDICIÓN then
    --sentencias secuenciales
else
    --sentencias secuenciales
end if ;

if CONDICIÓN then
    --sentencias secuenciales
elsif CONDICIÓN then
    --sentencias secuenciales
...
...
else
    --sentencias secuenciales
end if;
```

Lenguajes de Descripción de Hardware

EJEMPLO

```
entity EJEMPLO is
port (A, B, C, X: in bit_vector(3 downto 0);
      Z: out bit_vector(3 downto 0));
end EJEMPLO ;
architecture ARCH_1 of EJEMPLO is
begin
    process (A, B, C, X )
    begin
        if ( X = "1111" ) then
            Z <= B ;
        elsif ( X > "1000" ) then
            Z <= C ;
        else
            Z <= A ;
        end if ;
    end process ;
end ARCH_1;
```

Lenguajes de Descripción de Hardware

SENTENCIA CASE

- La estructura general es la siguiente:

```
case EXPRESIÓN is
when VALOR1 =>
    --sentencias secuenciales
when VALOR2 |VALOR3 =>
    --sentencias secuenciales
when VALOR4 to VALOR9 =>
    --sentencias secuenciales
when OTHERS =>
    --sentencias secuenciales
end case ;
```
- Las diferentes opciones no pueden solaparse.
- Todas las opciones tienen que ser cubiertas.
- Las opciones pueden hacerse cuando EXPRESIÓN adopta un valor, un rango de valores o uno de entre dos valores.

Lenguajes de Descripción de Hardware

EJEMPLO

```
entity SENT_CASE is
port ( A, B, C, X: in integer range 0 to 15 ;
      Z: out integer range 0 to 15);
end SENT_CASE ;
architecture EJEMPLO of SENT_CASE is
begin
    process ( A, B, C, X )
    begin
        case X is
            when 0 =>
                Z <= A ;
            when 7|9 =>
                Z <= B ;
            when 1 to 5 =>
                Z <= C ;
            when others =>
                Z <= 0 ;
            end case ;
        end process ;
    end EJEMPLO ;
```

Lenguajes de Descripción de Hardware

LAZOS FOR

- Se utilizan para ejecutar el mismo código un número determinado de veces.
- El parámetro del loop está implícitamente definido, no puede ser declarado externamente, es de sólo lectura y es visible sólo dentro del lazo.
- El parámetro adopta todos los valores de la definición de rango.

Lenguajes de Descripción de Hardware

EJEMPLO

```
entity LAZO_FOR is
port ( A : in integer range 0 to 3 ;
      Z : out bit_vector(3 downto 0));
end LAZO_FOR ;
architecture EJEMPLO of LAZO_FOR is
begin
    process (A)
    begin
        Z <= "0000" ;
        for I in 0 to 3 loop
            if ( A = I ) then
                Z(I) <= '1' ;
            end if ;
        end loop ;
    end process ;
end EJEMPLO ;
```

Lenguajes de Descripción de Hardware

TRES EJEMPLOS DE LAZOS

```
entity CONV_INT is
port( DIN : in bit_vector ( 7 downto 0 ) ;
      RESULT : out integer ;
end CONV_INT ;
```

EJEMPLO 1

```
process (DIN)
variable TMP: integer;
begin
    TMP:=0 ;
    for I in 7 downto 0 loop
        if (DIN(I)='1') then
            TMP := TMP + 2**I;
        end if ;
    end loop;
    RESULT<= TMP ;
end process;
```

EJEMPLO 2

```
process (DIN)
variable TMP: integer;
begin
    TMP:=0 ;
    for I in DIN'range loop
        if (DIN(I)='1') then
            TMP := TMP + 2**I;
        end if ;
    end loop;
    RESULT<= TMP ;
end process;
```

EJEMPLO 3

```
...
process (DIN)
variable TMP: integer;
variable I : integer
begin
    TMP:=0 ;
    I:= DIN'high;
    while(I>= DIN'low) loop
        if (DIN(I)='1') then
            TMP := TMP + 2**I;
        end if ;
        I=I-1
    end loop;
    RESULT<= TMP ;
end process;
```

Lenguajes de Descripción de Hardware

LA SENTENCIA WAIT

Esta sentencia detiene el proceso de ejecución hasta que se cumplan los requisitos planteados (que transcurra un cierto tiempo, que suceda un evento, que se cumpla una condición).

```
entity FF is
port (D, CLK : in bit;
      Q : out bit);
end FF;
architecture BEHAV of FF is
begin
    process
    begin
        wait on CLK;
        if ( CLK = '1') then
            Q <= D ;
        end if;
    end process;
end BEHAV ;
```

Lenguajes de Descripción de Hardware

SENTENCIAS CONCURRENTES

- Las sentencias concurrentes se ejecutan en paralelo sin importar el orden en que aparecen en el modelo.

Lenguajes de Descripción de Hardware

ASIGNACIÓN DE SEÑAL CONDICIONADA

- Ejemplo

```
entity ASIG_COND is
port ( A,B,C,X:in bit_vector(3 downto 0);
      Z_CONC:out bit_vector(3 downto 0);
      Z_SEC :out bit_vector(3 downto 0));
end ASIG_COND ;
architecture EJEMPLO of ASIG_COND is
-- VERSIÓN CONCURRENTES
  Z_CONC <= B when X = "1111"
  else
  C when X = "1000"
  else A ;
---VERSIÓN SECUENCIAL
  process ( A, B, C, X )
  begin
    if ( X = "1111" ) then
      Z_SEC <= B;
    elsif ( X > "1000" ) then
      Z_SEC <= C;
    else
      Z_SEC <= A ;
    end if ;
  end process;
end EJEMPLO ;
```

Lenguajes de Descripción de Hardware

ASIGNACIÓN DE SEÑAL SELECCIONADA

- La estructura general es la siguiente:

```
with EXPRESIÓN select
TARGET <= VALOR1 when CHOICE1;
        VALOR2 when CHOICE2|CHOICE3;
        VALOR3 when CHOICE4 TO CHOICE5;
        VALORN when others;
```

- Las diferentes opciones no pueden solaparse.
- Todas las opciones tienen que ser cubiertas.
- Las opciones pueden hacerse cuando EXPRESIÓN adopta un valor, un rango de valores o uno de entre dos valores.

Lenguajes de Descripción de Hardware

EJEMPLO

```
entity ASIG_SELEC is
port ( A,B,C,X: in integer range 0 to 15 ;
      Z_CONC : out integer range 0 to 15 ;
      Z_SEC  : out integer range 0 to 15) ;
end ASIG_SELECT ;

architecture EJEMPLO of ASIG_SELEC is
begin
-- VERSIÓN CONCURRENTES
  with X select
    Z_CONC <= A when 0,
    B when 7|9,
    C when 1 to 5
    0 when others ;
--VERSIÓN SECUENCIAL
  process (A,B,C,X)
  begin
    case X is
      when 0 => Z_SEC <= A ;
      when 7|9 => Z_SEC <= B ;
      when 1 to 5 => Z_SEC <= C ;
      when others => Z_SEC <= 0 ;
    end case;
  end process ;
end EJEMPLO ;
```

Lenguajes de Descripción de Hardware