

Diseño Automático de sistemas

Repaso VHDL

Prof. Pablo Sarabia Ortiz



UNIVERSIDAD
NEBRIJA

Contenidos

1. Generalidades
2. Estructura de VHDL
3. Bloques combinacionales
4. Bloques secuenciales: Process



Bibliografía

- Chapter 2: Digital Systems Design Using VHDL (Second Edition), Charles H. Roth, Jr and Lizy Kurian John.



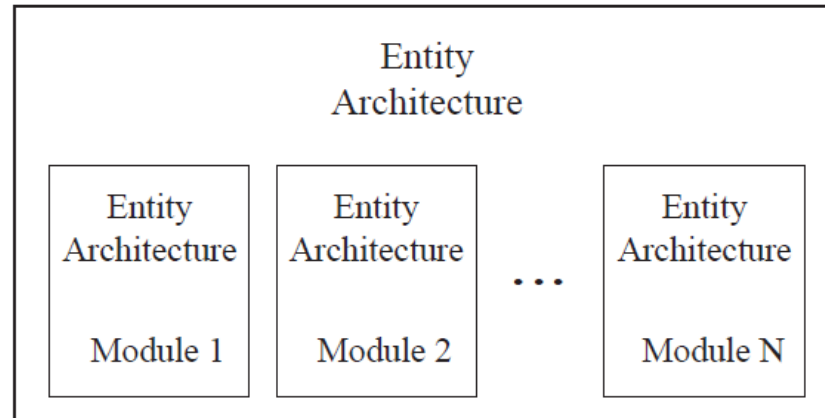
Generalidades

- VHDL: VHSIC (Very High Speed Integrated Circuits) Hardware Description Language
- Surge como herramienta de documentación de circuitos digitales.
- Se puede utilizar para diseño y para testbench
- Existe código sintetizable y no sintetizable



Estructura de VHDL

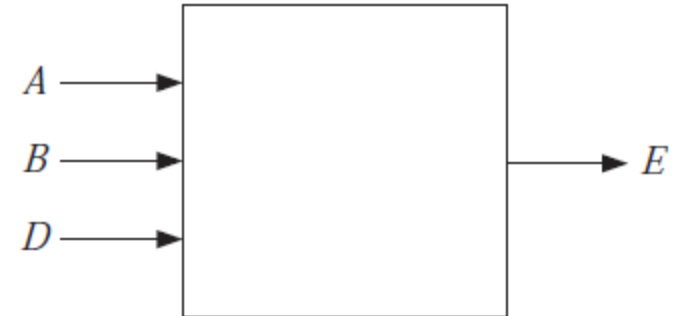
- Compuesto de *entity* y *architecture*:
 - *Entity*: Da nombre al módulo y describe las entradas y salidas mediante *port*.
 - *Architecture*: Especifica el funcionamiento del módulo



Estructura de VHDL: Entity

```
entity entity-name is  
    [port(interface-signal-declaration);]  
end [entity] [entity-name];
```

```
entity two_gates is  
    port(A, B, D: in bit; E: out bit);  
end two_gates;
```



Estructura de VHDL: Declaración de interfaz

```
list-of-interface-signals: mode type [:= initial-value]  
{; list-of-interface-signals: mode type [:= initial-value]};
```

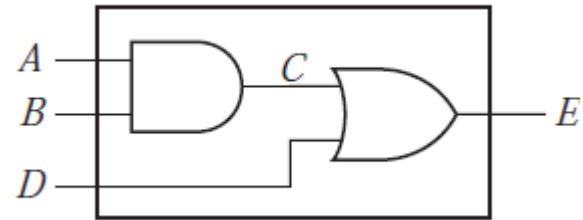
```
12 entity RS232_TX is  
13  
14     port(  
15         clk           :in    std_logic;           -- Reloj del sistema  
16         reset_n       :in    std_logic;           -- Reset asíncrono  
17         Start         :in    std_logic;           -- Comienzo de la transmisión  
18         Data          :in    std_logic_vector(7 downto 0); -- Datos a transmitir  
19         EOT           :out    std_logic;           -- Emisor disponible  
20         TX            :out    std_logic;           -- Salida  
21     );  
22  
23 end entity RS232_TX;
```



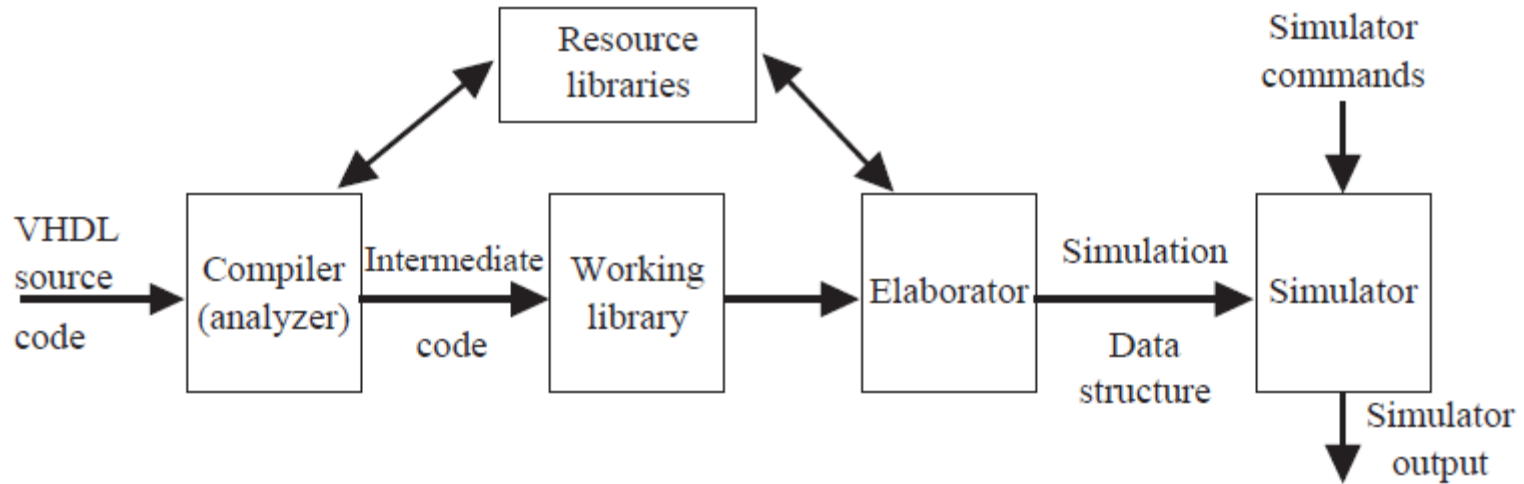
Estructura de VHDL: Architecture

```
architecture architecture-name of entity-name is
    [declarations]
begin
    architecture body
end [architecture] [architecture-name];
```

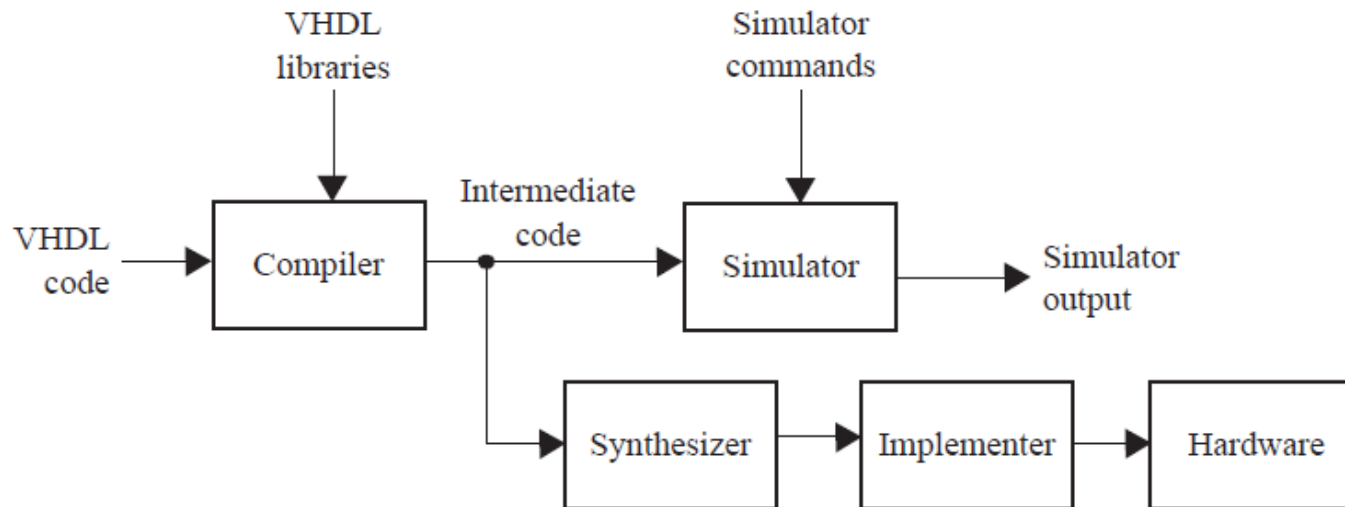
```
architecture gates of two_gates is
    signal C: bit;
begin
    C <= A and B; -- concurrent
    E <= C or D;  -- statements
end gates;
```



Proceso de compilación, simulación y síntesis



Proceso de compilación, simulación y síntesis



Bloques combinacionales

- Por defecto cualquier código escrito en VHDL es combinacional.
- Es concurrente.

```
66 architecture Equations of FullAdder is
67 begin -- concurrent assignment statements
68     Sum <= X xor Y xor Cin;
69     Cout <= (X and Y) or (X and Cin) or (Y and Cin);
70 end Equations;
```



Bloques secuenciales

- Todo lo incluido en un process.
- Es secuencial.
- Nos permite registrar señales.
- En la sensitivity list solo debe estar el reloj y el reset asíncrono

```
process(sensitivity-list)
begin
    sequential-statements
end process;
```



Bloques secuenciales

- En este caso la señal q está registrada y se inicializa a 0 cuando se activa el reset asíncrono.

```
92 process (clk, reset)
93 begin
94     if reset = '1' then
95         q <= '0' --Default value
96     elsif rising_edge(clk) then
97         q <= a or b;
98     end if;
99 end process;
```



¿Preguntas?



UNIVERSIDAD
NEBRIJA