Especificación, compilación y síntesis de una compuerta AND en AHDL, VHDL y Verilog.

Compuerta AND

Una compuerta lógica AND es un tipo de circuito que se utiliza en electrónica digital y que realiza una operación booleana de conjunción lógica. Es decir, la salida de la compuerta AND es "verdadera" (1) solamente si todas sus entradas también son "verdaderas", de lo contrario la salida es "falsa" (0).

La compuerta AND tiene dos o más entradas y una única salida. Las entradas pueden ser señales eléctricas o digitales, y la salida también es una señal eléctrica o digital. La operación de la compuerta AND se puede expresar mediante una tabla de verdad:

Entrada A	Entrada B	Salida
0	0	0
0	1	0
1	0	0
1	1	1

La tabla de verdad indica que la salida de la compuerta AND es "verdadera" (1) solamente si ambas entradas son "verdaderas" (1). Si una o ambas entradas son "falsas" (0), la salida será "falsa" (0).

La compuerta AND se utiliza en circuitos electrónicos para realizar operaciones lógicas, como la comparación de valores binarios o la toma de decisiones en base a varias condiciones.

Implementación en AHDL

```
1
2   -- Declaramos el SUBDESING con el nombre
3   -- MyAND
4   SUBDESIGN MyAND{
5      -- Le damos dos entradas y una salida.
6      A,B:INPUT;
7      C:OUTPUT;
8   }
9   -- Declaramos la logica de nuestro and;
10  BEGIN
11     C=A and B;
12  END;
```

Implementación en VHDL

```vhdl
-- SIEMPRE hay que indicar la libreria de STD_LOGIC
library IEEE;
   use IEEE.STD_LOGIC_1164.ALL;

-- La entidad funciona como la caja negra
-- solo nos sirve para declarar las variables que ocupara
-- nuestro ciruito.
entity AndVHDL is

--declaramos nuestras entradas y salidas
   port (

   -- las entradas se declaran con "in" haciendo uso de std_logic.
      A: in std_logic;
      B: in std_logic;
   -- las salidas se declaran on "out" haciendo uso de std_logic.
      C : out std_logic

      --std_logic sirve para declarar valores binarios.
      );

end AndVHDL;

-- iniciamos con la arquitectura de nuestro circuito
-- aqui declaramos toda la logica.
architecture gate of AndVHDL is -- gate es el nombre de la arquitectura
begin
   -- Si A y B son 1 entonces C vale 1.
   C<= A and B;

end gate;
```

Implementación en Verilog

```verilog
//verilog esta medio raro

// Iniciamos un modulo, le decimos que variables usara
module AndVerilog(a1,b1,y);

   // Delaramos esas variables como entreadas
   input a1,b1;
   // o salidas
   output y;

// En assign declaramos la logica de nuestro modulo
assign y = a1 & b1;

// Cerramos el modulo
endmodule
//no hace falta un punto y como al final
```