

¿Para qué sirve la palabra reservada «open» en VHDL?

Creador: David Rubio G.

Entrada: <https://soceame.wordpress.com/2024/02/13/para-que-sirve-la-palabra-reservada-open-en-vhdl/>

Blog: <https://soceame.wordpress.com/>

GitHub: <https://github.com/DRubioG>

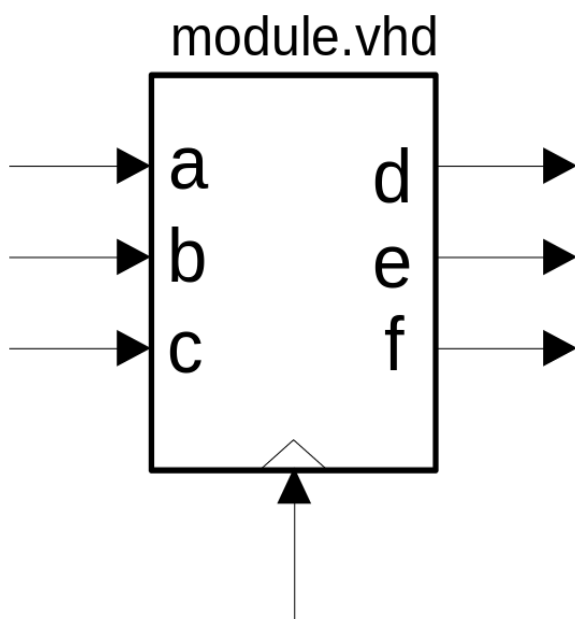
Fecha última modificación: 22/02/2025

Esta palabra reservada hasta hace poco no sabía ni que existía, pero es superútil utilizarla, y supersimple entenderla.

¿Para qué sirve?

Sirve para dejar la entrada de una implementación como vacía o abierta(de ahí lo de «open»). Te pongo un ejemplo fácil de entender.

Imagina que quieres usar la implementación de un modulo como el de la imagen.



Pero de este bloque solo quieres que esté conectadas las entradas «a» y «b», pero no «c». Y de la salida solo la salida «d».

Bien pues con los puertos de salida no tienes ningún problema con dejarlos desconectados (no declarándolos en la implementación). Con las entradas es diferente, porque el sintetizador te va a obligar a que estén, y que estén a un determinado valor o conectado a algo. Ahí, es donde entra en funcionamiento la palabra «*open*», porque es ahí donde se aplica. Te dejo un ejemplo de *architecture* para que lo veas.

```
architecture blabla of blabla is
-- esta es una declaración normal del module que queremos
-- implementar
  component module is
    port(
      clk : in std_logic;
      a, b, c : in std_logic;
      d, e, f : out std_logic
    );
  end component;
```

```
-- fin declaracion
signal a_i, b_i, d_i : std_logic;
begin
-- implementacion
impl_module : module
    port map(
        clk => clk,
        a => a_i,
        b => b_i,
        -- hasta aquí todo normal
        c => open, -- aquí es donde se declara que el puerto
                -- de entrada c no va a estar conectado
        d => d_i -- el resto de puertos de salida que no
                -- van a estar conectados no es necesario
                -- implementarlos, y si se hace se les
                -- aplica también un "open"
    );
-- fin implementacion
end architecture;
```

*NOTA: por si os da por comparar, en Verilog los puertos de entrada que no se utilizan, es necesario que estén declarados, pero se dejan vacíos [.c()].