What is the reserved word “open” used for in VHDL?

Created by: David Rubio G.

Blog post: https://soceame.wordpress.com/2025/03/04/what-is-the-reserved-word-open-used-for-in-vhdl/

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

GitHub: https://github.com/DRubioG

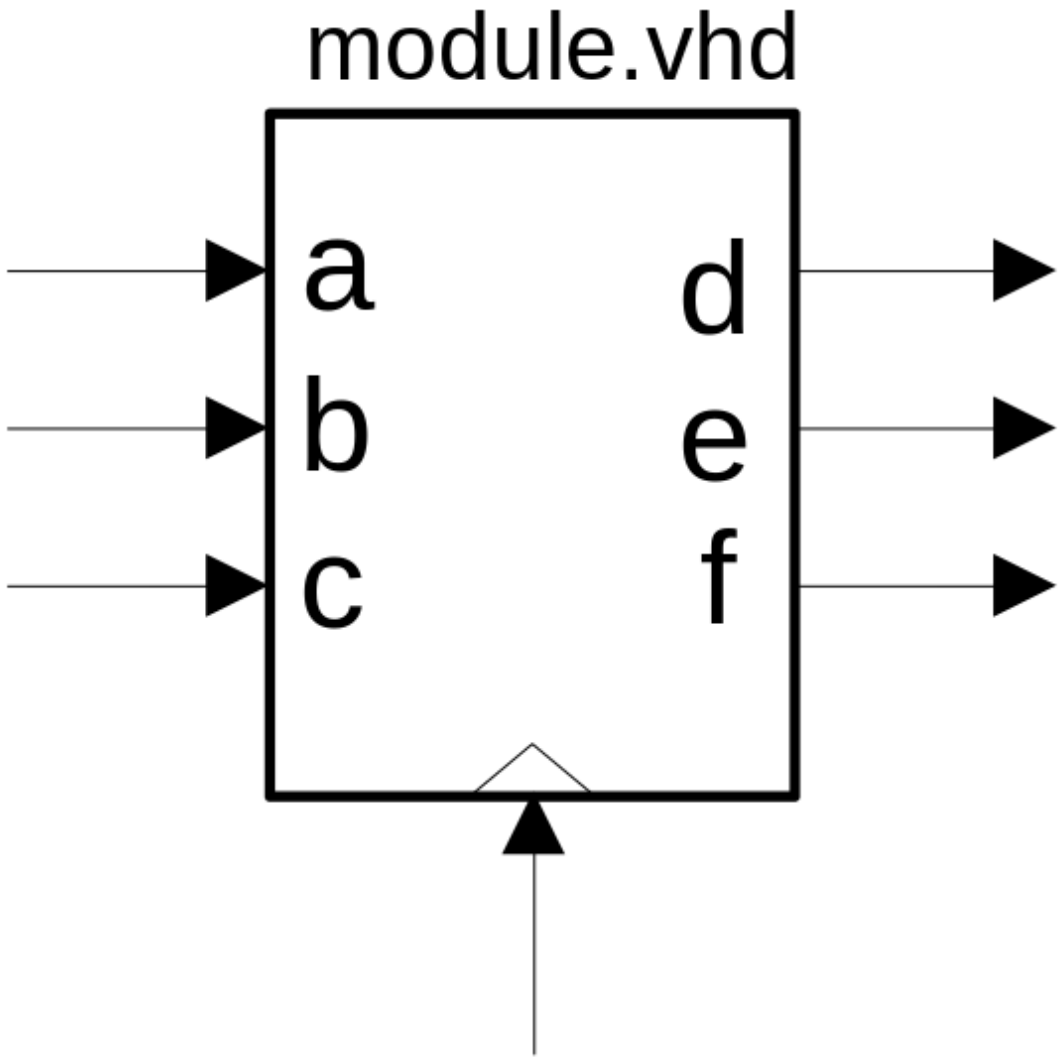
Last modification date: 04/03/25

This reserved word, until recently, I didn't even know it existed, but it's super useful to use and super easy to understand.

What is it for?

It's used to leave the entry of an implementation empty or open (hence the "open"). Here's an easy-to-understand example.

Imagine that you want to use the implementation of a module like the one in the image.



But in this block you only want the inputs "a" and "b" to be connected, but not "c". And from the output only the output "d".

Well, with the output ports you have no problem leaving them disconnected (not declaring them in the implementation). With the inputs it is different, because the synthesizer will force them to be there, and to be at a certain value or connected to something. That is where the word "*open*" comes into play, because that is where it is applied. I leave you an example of *architecture* so you can see it.

```
architecture blablabla of blabla is

-- this is a normal declaration of the module we want to
-- implement
   component module is
      port(
            clk : in std_logic;
            a, b, c : in std_logic;
            d, e, f : out std_logic
      );
   end component;
-- end declaration

signal a_i, b_i, d_i : std_logic;

begin

-- implementation
impl_module : module
      port map(
            clk => clk,
            a => a_i,
            b => b_i,
            -- so far everything is normal
            c => open,   -- this is where you declare that the
                    -- input port c is not going to be
                    -- connected
        d => d_i      -- the rest of the output ports that are
                    -- not going to be connected are not
                    -- necessary
                    -- implement them, and if you do, they
                    -- are also applies an "open"
      );
-- end implementation


end architecture;
```

*NOTE: just in case you want to compare, in Verilog the input ports that are not used must be declared, but they are left empty [*.c( )*].