

Module pos ○

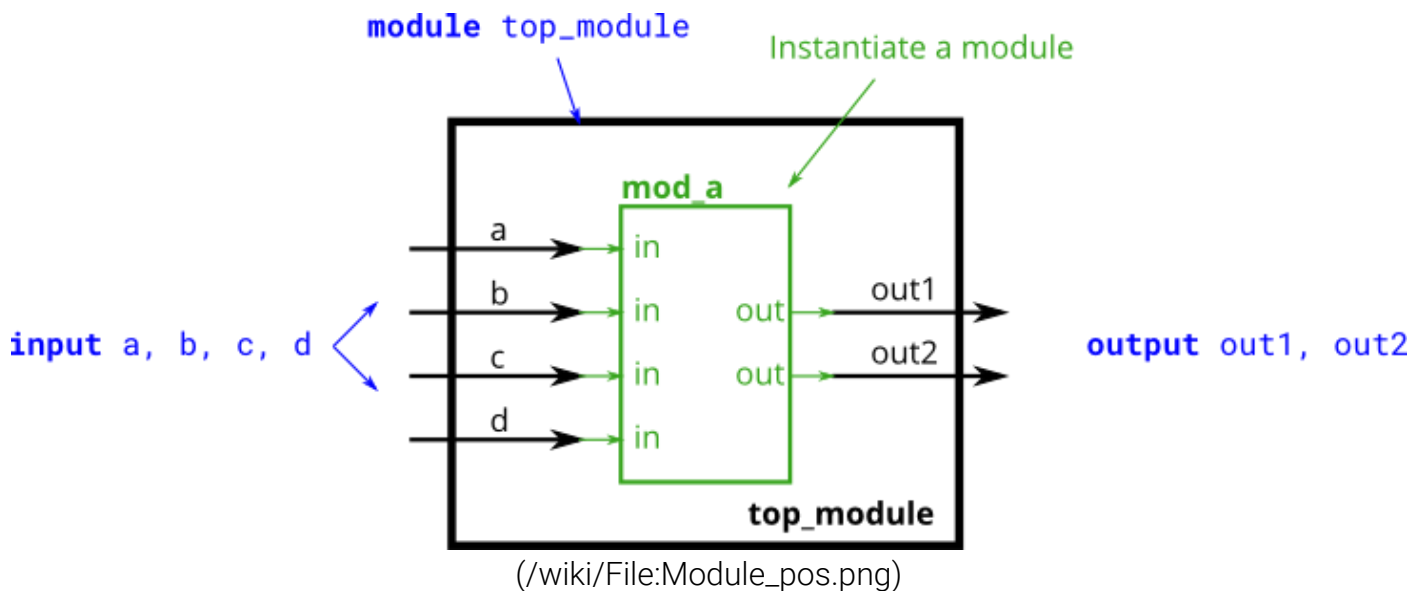
← module✔ (/wiki/module)

module_name○ (/wiki/module_name) →

This problem is similar to the previous one (module✔ (/wiki/module)). You are given a module named `mod_a` that has 2 outputs and 4 inputs, in that order. You must connect the 6 ports *by position* to your top-level module's ports `out1`, `out2`, `a`, `b`, `c`, and `d`, in that order.

You are given the following module:

```
module mod_a ( output, output, input, input, input, input );
```



Expected solution length: Around 1 line.

Module Declaration

```
module top_module (  
    input a,  
    input b,  
    input c,  
    input d,  
    output out1,  
    output out2  
);
```

Write your solution here

```
1 module top_module (  
2     input a,  
3     input b,
```

```


4     input c,
5     input d,
6     output out1,
7     output out2
8 );
9
10 endmodule
11

```

Submit

Submit (new window)

Upload a source file... 

← [module](/wiki/module)  (/wiki/module)

[module_name](/wiki/module_name)  (/wiki/module_name) →

Retrieved from "http://hdlbits.01xz.net/mw/index.php?title=Module_pos&oldid=1479
(http://hdlbits.01xz.net/mw/index.php?title=Module_pos&oldid=1479)"

Category (/wiki/Special:Categories): **Modules (/wiki/Category:Modules)**

Problem Set Contents


► Getting Started

▼ **Verilog Language**

► Basics


► Vectors

▼ **Modules: Hierarchy**

 [Modules \(/wiki/module\)](/wiki/module)

 **[Connecting ports by position \(/wiki/module_pos\)](/wiki/module_pos)**


 [Connecting ports by name \(/wiki/module_name\)](/wiki/module_name)


 [Three modules \(/wiki/module_shift\)](/wiki/module_shift)

 [Modules and vectors \(/wiki/module_shift8\)](/wiki/module_shift8)

 [Adder 1 \(/wiki/module_add\)](/wiki/module_add)

 [Adder 2 \(/wiki/module_fadd\)](/wiki/module_fadd)

 [Carry-select adder \(/wiki/module_cseladd\)](/wiki/module_cseladd)

 [Adder-subtractor \(/wiki/module_addsub\)](/wiki/module_addsub)

► Procedures

► More Verilog Features

► Circuits

► Verification: Reading Simulations

► Verification: Writing Testbenches

