Wire4

wire (/wiki/wire)

notgate (/wiki/notgate)

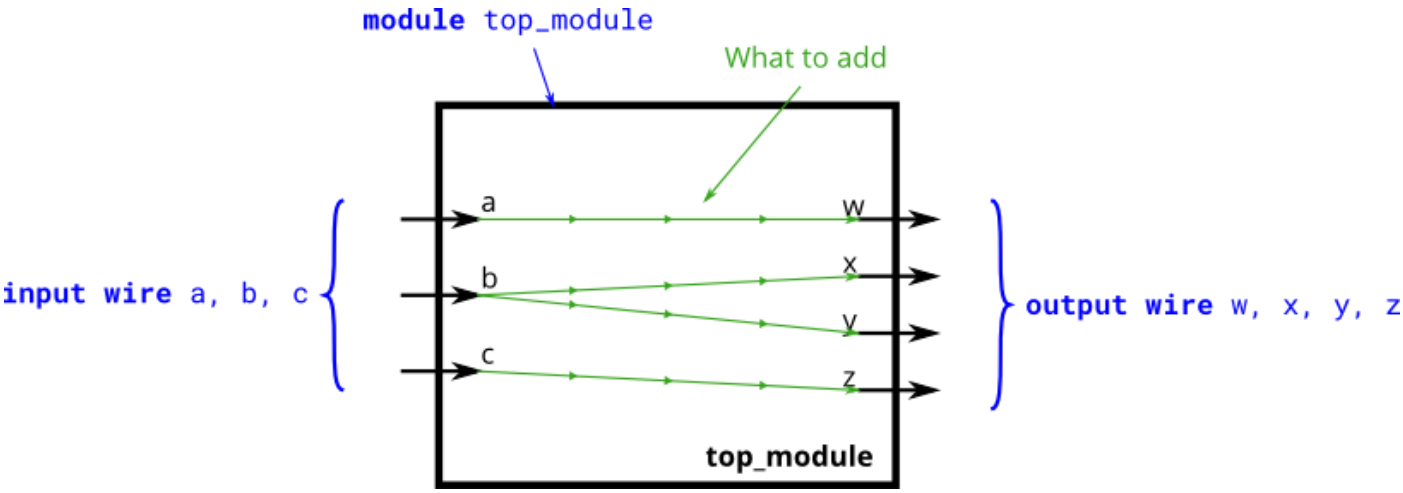
Create a module with 3 inputs and 4 outputs that behaves like wires that makes these connections:

```
a -> w
b -> x
b -> y
c -> z
```

The diagram below illustrates how each part of the circuit corresponds to each bit of Verilog code. From outside the module, there are three input ports and four output ports.

When you have multiple `assign` statements, the **order** in which they appear in the code **does not matter**. Unlike a programming language, `assign` statements ("continuous assignments") describe *connections* between things, not the *action* of copying a value from one thing to another.

One potential source of confusion that should perhaps be clarified now: The green arrows here represent connections *between* wires, but are not wires in themselves. The module itself *already* has 7 wires declared (named a, b, c, w, x, y, and z). This is because `input` and `output` declarations actually declare a wire unless otherwise specified. Writing `input wire a` is the same as `input a`. Thus, the `assign` statements are not creating wires, they are creating the connections between the 7 wires that already exist.



(/wiki/File:Wire4.png)

Expected solution length: *Around 4 lines.*

Module Declaration

```
module top_module(
    input a,b,c,
    output w,x,y,z );
```

Hint...

Write your solution here

```
1 module top_module(
2     input a,b,c,
3     output w,x,y,z );
4
5 endmodule
6
```

Submit | Submit (new window)

Upload a source file...

Solution

Show solution

← wire (/wiki/wire)

notgate (/wiki/notgate) →

Retrieved from "http://hdlbits.01xz.net/mw/index.php?title=Wire4&oldid=1472 (http://hdlbits.01xz.net/mw/index.php?title=Wire4&oldid=1472)"

Problem Set Contents

- Getting Started
- **Verilog Language**
  - **Basics**
    - Simple wire (/wiki/wire)
    - **Four wires (/wiki/wire4)**
    - Inverter (/wiki/notgate)
    - AND gate (/wiki/andgate)
    - NOR gate (/wiki/norgate)
    - XNOR gate (/wiki/xnorgate)
    - Declaring wires (/wiki/wire_decl)
    - 7458 chip (/wiki/7458)
  - Vectors

- Modules: Hierarchy
- Procedures
- More Verilog Features

- Circuits
- Verification: Reading Simulations
- Verification: Writing Testbenches