Wire

← zero (/wiki/zero)

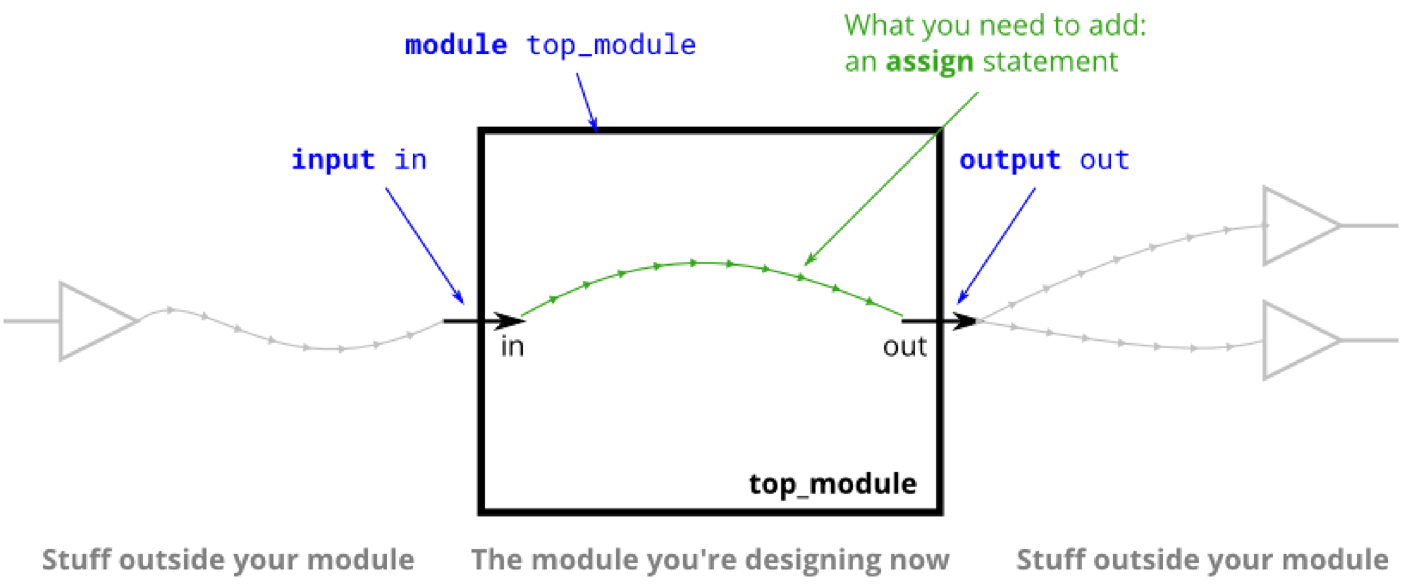
wire4 (/wiki/wire4) →

Create a module with one input and one output that behaves like a wire.

Unlike physical wires, wires (and other signals) in Verilog are *directional*. This means information flows in only one direction, from (usually one) *source* to the *sinks* (The source is also often called a *driver* that *drives* a value onto a wire). You must be aware of which direction information is flowing when writing `assign` statements.

Input and output ports also have a direction (that's why there are two types). An input port is *driven by* something from outside the module, while an output port *drives* something outside. From a perspective inside the module, an input port is a driver or source, while an output port is a sink.

The diagram below illustrates how each part of the circuit corresponds to each bit of Verilog code. The module and port declarations create the black portions of the circuit. Your task is to create the green (unidirectional) wire by adding in an `assign` statement to connect `in` to `out`. The parts outside the box are not your concern, but you should know that your circuit is tested by connecting signals from our test harness to the ports on your `top_module`.



(/wiki/File:Wire.png)

Expected solution length: Around 1 line.

Module Declaration

```
module top_module( input in, output out );
```

Hint...

A *continuous* assignment assigns the right side to the left side *continuously*, so any change to the RHS is immediately seen in the LHS.

Write your solution here

```
module top_module( input in, output out );

endmodule

```

Submit   Submit (new window)

Upload a source file...

Solution

Show solution

← zero✓ (/wiki/zero)

wire4✓ (/wiki/wire4) →

Retrieved from "http://hdlbits.01xz.net/mw/index.php?title=Wire&oldid=1594 (http://hdlbits.01xz.net/mw/index.php?title=Wire&oldid=1594)"

Problem Set Contents