

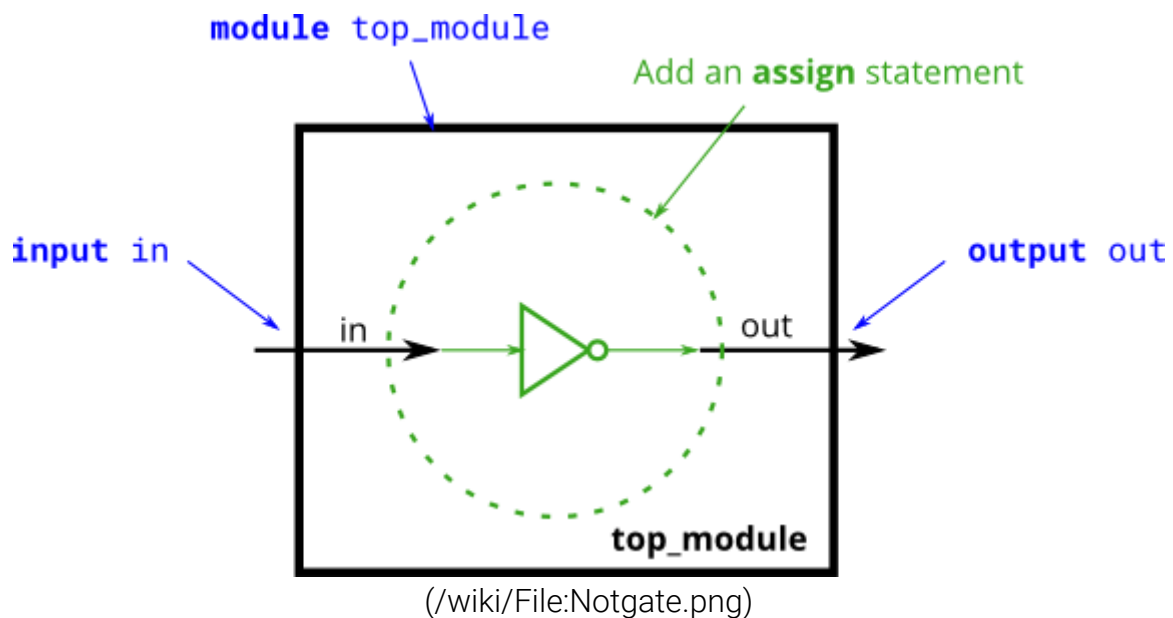
# Notgate ✓

[← wire4 ✓ \(/wiki/wire4\)](#)[andgate ✓ \(/wiki/andgate\) →](#)

Create a module that implements a NOT gate.

This circuit is similar to [wire ✓ \(/wiki/wire\)](#), but with a slight difference. When making the connection from the wire `in` to the wire `out` we're going to implement an inverter (or "NOT-gate") instead of a plain wire.

Use an assign statement. The assign statement will continuously (A change in the inputs immediately updates the output.) drive the inverse of `in` onto wire `out`.



Expected solution length: Around 1 line.

## Module Declaration

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

### Hint...

Verilog has separate bitwise-NOT (`~`) and logical-NOT (`!`) operators, like C. Since we're working with a one-bit here, it doesn't matter which we choose.

## Write your solution here

```
1 module top_module( input in, output out );
2
3 endmodule
4
```

Submit

Submit (new window)

Upload a source file... ▾

**Solution**

Show solution

← wire4 ✓ (/wiki/wire4)

andgate ✓ (/wiki/andgate) →

Retrieved from "http://hdlbits.01xz.net/mw/index.php?title=Notgate&oldid=1475  
(http://hdlbits.01xz.net/mw/index.php?title=Notgate&oldid=1475)"

## 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