

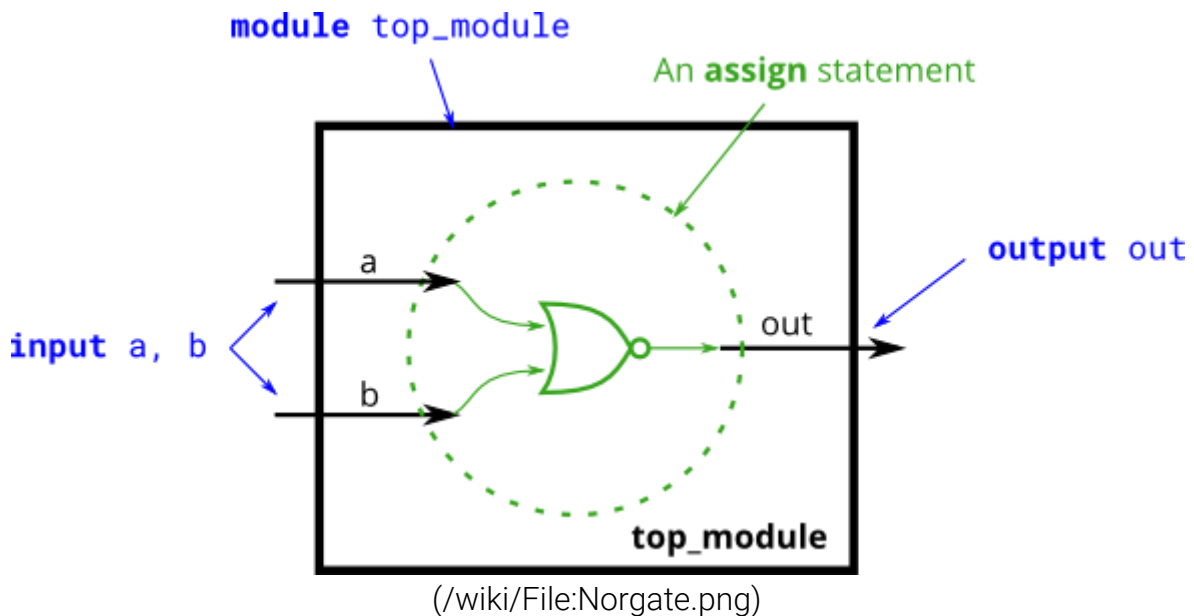
Norgate ✓

← andgate ✓ (/wiki/andgate)

xnorgate ✓ (/wiki/xnorgate) →

Create a module that implements a NOR gate. A NOR gate is an OR gate with its output inverted. A NOR function needs two operators when written in Verilog.

An assign statement drives a wire (or "net", as it's more formally called) with a value. This value can be as complex a function as you want, as long as it's a *combinational* (i.e., memory-less, with no hidden state) function. An assign statement is a *continuous assignment* because the output is "recomputed" whenever any of its inputs change, forever, much like a simple logic gate.



Expected solution length: Around 1 line.

Module Declaration

```
module top_module(  
    input a,  
    input b,  
    output out );
```

Hint...

Verilog has separate bitwise-OR (`|`) and logical-OR (`||`) 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 a,  
    input b,  
    output out );
```

```
2   input a,  
3   input b,  
4   output out );  
5  
6 endmodule  
7
```

Submit

Submit (new window)

Upload a source file... 

[← andgate](#)  (/wiki/andgate)

xnorgate  (/wiki/xnorgate) [→](#)









Retrieved from "http://hdlbits.01xz.net/mw/index.php?title=Norgate&oldid=1489
(http://hdlbits.01xz.net/mw/index.php?title=Norgate&oldid=1489)"

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