



Vectorgates

← [vector2](#)  (/wiki/vector2)

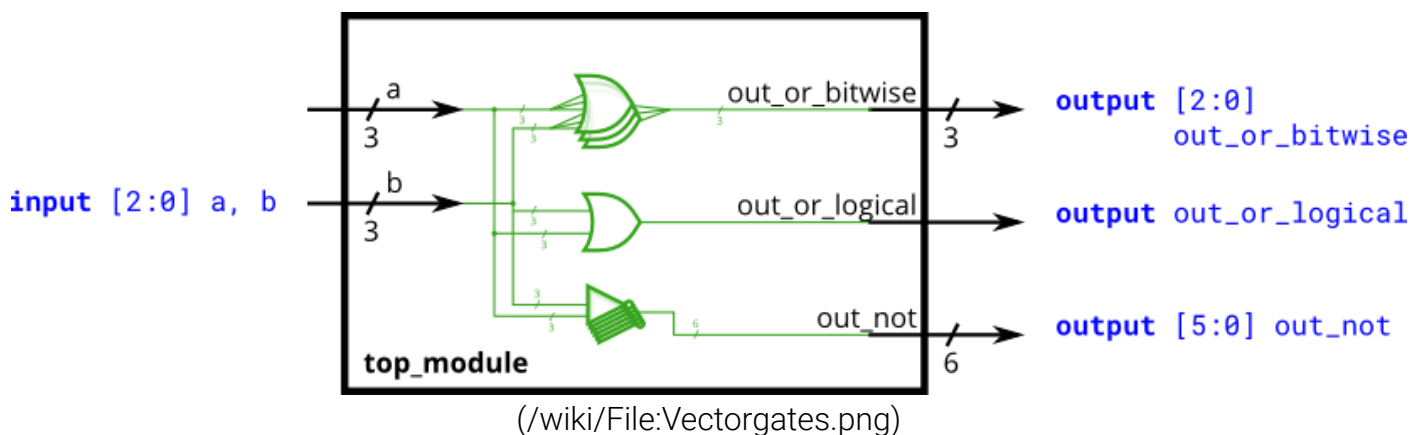
[gates4](#)  (/wiki/gates4) →

Build a circuit that has two 3-bit inputs that computes the bitwise-OR of the two vectors, the logical-OR of the two vectors, and the inverse (NOT) of both vectors. Place the inverse of b in the upper half of out_not (i.e., bits [5:3]), and the inverse of a in the lower half.

Bitwise vs. Logical Operators

Earlier, we mentioned that there are bitwise and logical versions of the various boolean operators (e.g., [norgate](#)  (/wiki/norgate)). When using vectors, the distinction between the two operator types becomes important. A bitwise operation between two N-bit vectors replicates the operation for each bit of the vector and produces a N-bit output, while a logical operation treats the entire vector as a boolean value (true = non-zero, false = zero) and produces a 1-bit output.

Look at the simulation waveforms at how the bitwise-OR and logical-OR differ.



Module Declaration

```
module top_module(
    input [2:0] a,
    input [2:0] b,
    output [2:0] out_or_bitwise,
    output out_or_logical,
    output [5:0] out_not
);
```

Hint...

Even though you cannot assign to a wire more than once, you can use a part select on the left-hand-side of an assign. You don't need to assign to the entire vector all in one statement.

Write your solution here

```
1 module top_module(  
2     input [2:0] a,  
3     input [2:0] b,  
4     output [2:0] out_or_bitwise,  
5     output out_or_logical,  
6     output [5:0] out_not  
7 );  
8  
9 endmodule  
10
```



Submit

Submit (new window)

Upload a source file... 

Solution

Show solution

 [vector2](/wiki/vector2)  (/wiki/vector2)

[gates4](/wiki/gates4)  (/wiki/gates4) 

Retrieved from "http://hdlbits.01xz.net/mw/index.php?title=Vectorgates&oldid=1182
(http://hdlbits.01xz.net/mw/index.php?title=Vectorgates&oldid=1182)"










Problem Set Contents

► Getting Started

▼ Verilog Language

► Basics

▼ Vectors

-  [Vectors \(/wiki/vector0\)](/wiki/vector0)
-  [Vectors in more detail \(/wiki/vector1\)](/wiki/vector1)
-  [Vector part select \(/wiki/vector2\)](/wiki/vector2)
-  **[Bitwise operators \(/wiki/vectorgates\)](/wiki/vectorgates)**
-  [Four-input gates \(/wiki/gates4\)](/wiki/gates4)
-  [Vector concatenation operator \(/wiki/vector3\)](/wiki/vector3)
-  [Vector reversal 1 \(/wiki/vectorr\)](/wiki/vectorr)
-  [Replication operator \(/wiki/vector4\)](/wiki/vector4)
-  [More replication \(/wiki/vector5\)](/wiki/vector5)

► Modules: Hierarchy

- ▶ Procedures
- ▶ More Verilog Features
- ▶ Circuits
- ▶ Verification: Reading Simulations
- ▶ Verification: Writing Testbenches