

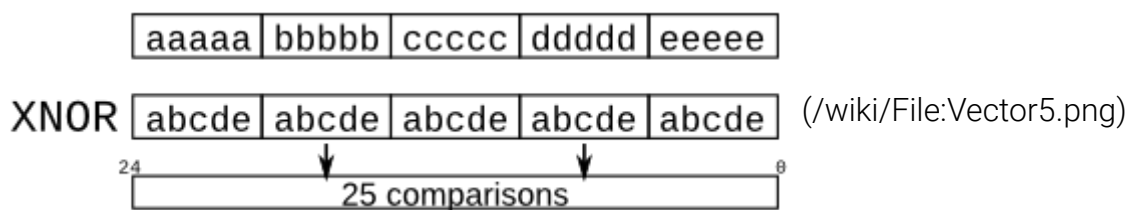
# Vector5 ○

← vector4 ✓ (/wiki/vector4)

module ○ (/wiki/module) →

Given five 1-bit signals (a, b, c, d, and e), compute all 25 pairwise one-bit comparisons in the 25-bit output vector. The output should be 1 if the two bits being compared are equal.

```
out[24] = ~a ^ a;    // a == a, so out[24] is always 1.
out[23] = ~a ^ b;
out[22] = ~a ^ c;
...
out[ 1] = ~e ^ d;
out[ 0] = ~e ^ e;
```



As the diagram shows, this can be done more easily using the replication (/wiki/Vector4) and concatenation operators.

- The top vector is a concatenation of 5 repeats of each input
- The bottom vector is 5 repeats of a concatenation of the 5 inputs

## Module Declaration

```
module top_module (  
    input a, b, c, d, e,  
    output [24:0] out );
```

## Write your solution here


```
1 module top_module (  
2     input a, b, c, d, e,  
3     output [24:0] out );//  
4  
5     // The output is XNOR of two vectors created by  
6     // concatenating and replicating the five inputs.  
7     // assign out = ~{ ... } ^ { ... };  
8  
9     endmodule  
10
```

Submit

Submit (new window)

## Solution

Complete problem first to see solution

[← vector4](#)  (/wiki/vector4)[module](#)  (/wiki/module) [→](#)

Retrieved from "http://hdlbits.01xz.net/mw/index.php?title=Vector5&oldid=1038  
(http://hdlbits.01xz.net/mw/index.php?title=Vector5&oldid=1038)"










## Problem Set Contents

▶ Getting Started

### ▼ Verilog Language

▶ Basics

#### ▼ Vectors

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

▶ Modules: Hierarchy

▶ Procedures

▶ More Verilog Features

▶ Circuits

▶ Verification: Reading Simulations

▶ Verification: Writing Testbenches