

Vector4 ○

← vectorr (/wiki/vectorr)

vector5 ○ (/wiki/vector5) →

The concatenation operator (/wiki/Vector3) allowed concatenating together vectors to form a larger vector. But sometimes you want the same thing concatenated together many times, and it is still tedious to do something like `assign a = {b, b, b, b, b, b};`. The replication operator allows repeating a vector and concatenating them together:

```
{num{vector}}
```

This replicates *vector* by *num* times. *num* must be a constant. Both sets of braces are required.

Examples:

```
{5{1'b1}}           // 5'b11111 (or 5'd31 or 5'h1f)
{2{a,b,c}}          // The same as {a,b,c,a,b,c}
{3'd5, {2{3'b6}}}}  // 9'b101_110_110. It's a concatenation of 101 with
                    // the second vector, which is two copies of 3'b110.
```

A Bit of Practice

One common place to see a replication operator is when sign-extending a smaller number to a larger one, while preserving its signed value. This is done by replicating the sign bit (the most significant bit) of the smaller number to the left. For example, sign-extending 4' b**0**101 (5) to 8 bits results in 8' b**0000**0101 (5), while sign-extending 4' b**1**101 (-3) to 8 bits results in 8' b**1111**1101 (-3).

Build a circuit that sign-extends an 8-bit number to 32 bits. This requires a concatenation of 24 copies of the sign bit (i.e., replicate bit[7] 24 times) followed by the 8-bit number itself.

Module Declaration

```
module top_module (
    input [7:0] in,
    output [31:0] out );
```

Write your solution here

```
1 module top_module (
2     input [7:0] in,
3     output [31:0] out );//
4
5     // assign out = { replicate-sign-bit , the-input };
6
7 endmodule
8
```


Submit

Submit (new window)

Upload a source file... 

Solution

Complete problem first to see solution

← vectorr  (/wiki/vectorr)

vector5  (/wiki/vector5) →

Retrieved from "http://hdlbits.01xz.net/mw/index.php?title=Vector4&oldid=1429"
(http://hdlbits.01xz.net/mw/index.php?title=Vector4&oldid=1429)"










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