

# Module fadd ○

← module\_add ● (/wiki/module\_add)

module\_cseladd ○ (/wiki/module\_cseladd) →

In this exercise, you will create a circuit with two levels of hierarchy. Your `top_module` will instantiate two copies of `add16` (provided), each of which will instantiate 16 copies of `add1` (which you must write). Thus, you must write two modules: `top_module` and `add1`.

Like `module_add` ● (/wiki/module\_add), you are given a module `add16` that performs a 16-bit addition. You must instantiate two of them to create a 32-bit adder. One `add16` module computes the lower 16 bits of the addition result, while the second `add16` module computes the upper 16 bits of the result. Your 32-bit adder does not need to handle carry-in (assume 0) or carry-out (ignored).

Connect the `add16` modules together as shown in the diagram below. The provided module `add16` has the following declaration:

```
module add16 ( input[15:0] a, input[15:0] b, input cin, output[15:0]
sum, output cout );
```

Within each `add16`, 16 full adders (module `add1`, not provided) are instantiated to actually perform the addition. You must write the full adder module that has the following declaration:

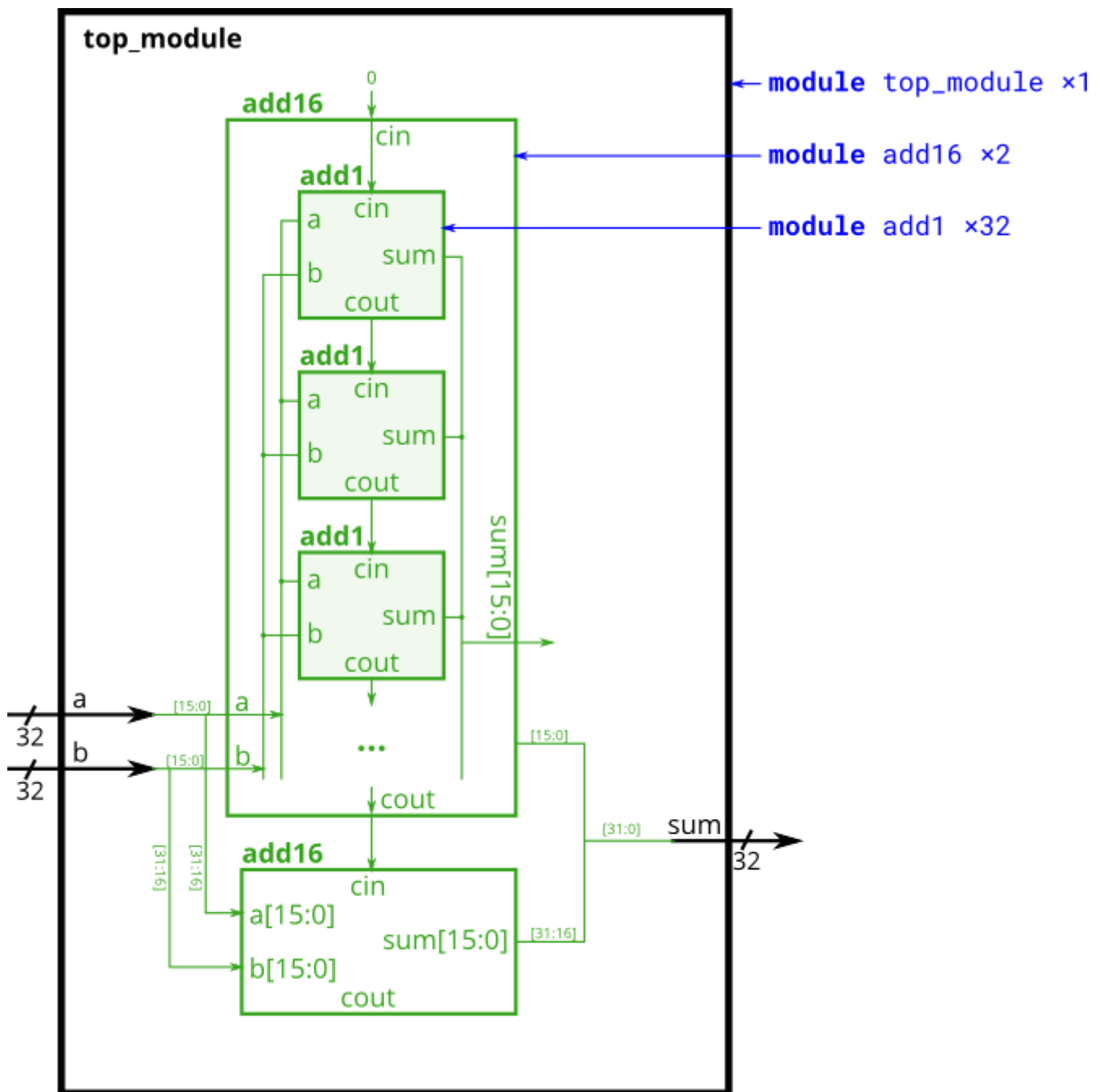
```
module add1 ( input a, input b, input cin, output sum, output cout );
```

Recall that a full adder computes the sum and carry-out of  $a+b+cin$ .

In summary, there are three modules in this design:

- `top_module` — Your top-level module that contains two of...
- `add16`, provided — A 16-bit adder module that is composed of 16 of...
- `add1` — A 1-bit full adder module.

If your submission is missing a module `add1`, you will get an error message that says Error (12006): Node instance "user\_fadd[0].a1" instantiates undefined entity "add1".



(/wiki/File:Module\_fadd.png)

## Module Declaration

```
module top_module (
    input [31:0] a,
    input [31:0] b,
    output [31:0] sum
);
```

Hint...

## Write your solution here

```
1 module top_module (
2     input [31:0] a,
```

```


3      input [31:0] b,
4      output [31:0] sum
5  );//
6
7  endmodule
8
9  module add1 ( input a, input b, input cin,    output sum, output cout );
10
11  // Full adder module here
12
13  endmodule
14

```

Submit

Submit (new window)

Upload a source file... 

[← module\\_add](#)  (/wiki/module\_add)

module\_cseladd  (/wiki/module\_cseladd) [→](#)

Retrieved from "http://hdlbits.01xz.net/mw/index.php?title=Module\_fadd&oldid=1449  
(http://hdlbits.01xz.net/mw/index.php?title=Module\_fadd&oldid=1449)"

Categories (/wiki/Special:Categories):    **Modules (/wiki/Category:Modules)**  
**Combinational (/wiki/Category:Combinational)**

## Problem Set Contents


► Getting Started

▼ **Verilog Language**

► Basics


► Vectors

▼ **Modules: Hierarchy**

 Modules (/wiki/module)

 Connecting ports by position (/wiki/module\_pos)


 Connecting ports by name (/wiki/module\_name)


 Three modules (/wiki/module\_shift)

 Modules and vectors (/wiki/module\_shift8)

 Adder 1 (/wiki/module\_add)

 **Adder 2 (/wiki/module\_fadd)**

 Carry-select adder (/wiki/module\_cseladd)

 Adder-subtractor (/wiki/module\_addsub)

► Procedures

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