

Step one

[← Index \(/wiki/Problem_sets\)](/wiki/Problem_sets)

[zero !\[\]\(c3d993ca47bfe2a953c700506ce31fa0_img.jpg\) \(/wiki/zero\) →](/wiki/zero)

Welcome to HDLBits!

Getting started in digital logic design can be overwhelming at first because you need to learn new concepts, a new **H**ardware **D**escription **L**anguage (e.g., Verilog), several new software packages, and often an FPGA board, *all at the same time*. HDLBits provides a way to practice designing and debugging simple circuits with a single click of "Simulate".

Designing a circuit requires several steps: Writing HDL (Verilog) code, compiling the code to produce a circuit, then simulating the circuit and fixing bugs.

Contents

- 1 Writing Code
- 2 Compiling (Logic Synthesis)
- 3 Simulation
- 4 Final Status
- 5 Problem Statement

Writing Code

The easiest way to write your code is to do so in the code editor box below. For this problem, we have filled in most of the code for you already. Go ahead and finish the code for this circuit.

Click **Simulate** to compile and simulate your design.

Compiling (Logic Synthesis)

Your code is compiled using Altera Quartus to produce a circuit. Quartus produces a large number of messages. Click **Show Quartus messages** to show/hide them. It's good practice to reduce the number of warnings, but it is sometimes not practical to remove them all.

Simulation

Your compiled circuit is simulated to test whether it functions correctly. HDLBits use ModelSim to simulate your circuit and our reference solution in parallel, then compares the outputs of the modules. The simulation reports back two things:

First, it reports whether your circuit matches the reference circuit exactly (zero "mismatches") or how many "mismatches" occurred. A mismatch is the number of samples where the output of your circuit does not match the reference output.

Second, it may produce timing diagrams that show your circuit outputs when running our test vectors. The simulation waveform is grouped into three sections: "Inputs", "Yours", and "Ref". In a correct circuit, "Your" outputs will be the same as the "Ref" outputs. The "Mismatch" signals tells you which samples have a mismatch.



The module name and port names of the top-level `top_module` must not be changed, or you will get a simulation error.

Final Status

If your circuit was correct, you will see **Status: Success!**. There are a few other possibilities:

- **Compile Error** — Circuit did not compile.
- **Simulation Error** — Circuit compiled successfully, but simulation did not complete.
- **Incorrect** — Circuit compiled and simulated, but the outputs did not match the reference.
- **Success!** — Circuit was correct

You can track or share your progress on the **My Stats (/wiki/Special:VlgStats/Me)** page.

Problem Statement

We're going to start with a small bit of HDL to get familiar with the interface used by HDLBits. Here's the description of the circuit you need to build for this exercise:

Build a circuit with no inputs and one output. That output should always drive 1 (or logic high).

Expected solution length: Around 1 line.

Module Declaration

```
module top_module( output one );
```

Hint...

We want to assign 1 to the output one.

Write your solution here

```
1 module top_module( output one );
2
3 // Insert your code here
4     assign one = [fixme];
5
6 endmodule
7
```

Submit

Submit (new window)

Upload a source file...

Solution

[Show solution](#)

← [Index \(/wiki/Problem_sets\)](/wiki/Problem_sets)

[zero](/wiki/zero) ✓ [\(/wiki/zero\)](/wiki/zero) →

Retrieved from "http://hdlbits.01xz.net/mw/index.php?title=Step_one&oldid=1573
(http://hdlbits.01xz.net/mw/index.php?title=Step_one&oldid=1573)"

Problem Set Contents

▼ Getting Started

✓ [Getting Started \(/wiki/step_one\)](/wiki/step_one)

✓ [Output Zero \(/wiki/zero\)](/wiki/zero)

- ▶ [Verilog Language](#)
- ▶ [Circuits](#)
- ▶ [Verification: Reading Simulations](#)
- ▶ [Verification: Writing Testbenches](#)