

Demo 01: Verilog & Icarus Verilog

Eddy Yau

May 1, 2018

1 Introduction

1.1 Verilog

Verilog and VHDL are the most commonly used **HDL**(Hardware Description Language) in the world. On the other hand, high-level synthesis will utilize SystemC instead. Comparing with verilog and VHDL, verilog is more easily to study, since verilog is similar to C.

Here is some example code for a NAND gate in verilog:

Source Code 1: NAND.v

```
1 module NAND {
2     input  A,
3     input  B,
4     output Z
5 };
6 // Assign nand-gate logic to Z
7 assign Z=~A&B;
8 endmodule
```

Declaration for a module:

```
1 module NAND {
```

Pin list of the module

```
2     input  A,
3     input  B,
4     output Z
```

Assign combinational logic to output

```
7 assign Z=~A&B;
```

End of module declaration

```
8 endmodule
```

1.2 Icarus Verilog

Icarus Verilog is a Verilog simulation and synthesis tool. It operates as a compiler, compiling source code written in Verilog (IEEE-1364) into some target format. [\[iverilogwebsite\]](#) It is free to use.

Iverilog is the compiler to synthesis the verilog source code to vvp assembly. **Vvp** is the tool to execute the vvp assembly.

2 Simulation

In this demo, we will create a “Hello World” program executing by Icarus Verilog. For this demo, we are using Ubuntu 17.10 dist.

2.1 Install

First, we install the Icarus Verilog. Open a terminal and install Icarus Verilog via apt

```
$> sudo apt install iverilog
```

2.2 Create verilog source code

Create a verilog file as below:

Source Code 2: hello_world.v

```
1 module hello_world;
2 initial begin
3     display>Hello World!);
4     finish;
5 end
6 endmodule
```

2.3 Compile & Execute

```
$> iverilog -o hello_world.vvp hello_world.v
$> vvp hello_world.vvp
Hello World!
```