BBM233 Logic Design Lab Fall 2023

Guide to Verilog Lab Experiments Getting Started With Icarus Verilog and GTKWave

(Unofficial software but good enough for this course) Installation Steps For Windows November 17, 2023

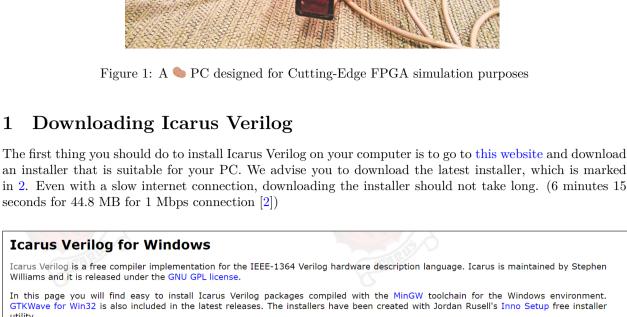
Icarus Verilog is a free and open-source Verilog compiler and simulator which can be used to develop HDL

code. You don't need any accounts or large storage space available to download this tool. You can learn more about this software, read installation and usage instructions both for Windows and for other platforms

Icarus Verilog

from their official website. However, if you are looking for some simple instructions for Windows, you can just keep following our guide.

TUDENTS WAITING FOR XILINX SOFTWARE TO DOWNLOAD can be used to develop HDL code, thanks to this amazing tool.



You can find Icarus Verilog sources and binaries for most platforms at the Icarus site FTP. The sources available here have been

compressed with 7-zip. iverilog-v11-20210204-x64_setup.exe [44.1MB]

1

utility.

Download

the step illustrated below:

iverilog -v gtkwave -V

PowerShell 7 (x64)

named "hello_world.v":

module hello_world; initial begin

end endmodule;

vvp example_1

endmodule

\$display("Hello world!");

source file, and enter the following commands:

iverilog -o example_1 hello_world.v

PS C:\Users\alperen> iverilog -∀ Ccarus Verilog version 12.0 (devel) (s20150603-1110-g18392a46) Copyright (c) 2000-2021 Stephen Williams (steve@icarus.com)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

 iverilog-v11-20201123-x64_setup.exe [1
 iverilog-10.1.1-x64_setup.exe [9.77MB] iverilog-10.0-x86_setup.exe [11.1MB]
iverilog-20130827_setup.exe (development snapshot) [11.2MB] • iverilog-0.9.7_setup.exe (latest stable release) [10.5MB] iverilog-0.9.6_setup.exe [10.4MB]
iverilog-0.8.6_setup.exe (latest release 0.8 series) [1.29MB] iverilog-0.8.6.7z [800kB] • iverilog-0.7-20040706_setup.exe [1.09MB] iverilog-0.7-20040706.7z [588kB]

 $\mathbf{2}$ Installation Steps Even though installing software on Windows is pretty straightforward, some steps that can potentially make you suffer if you don't pay attention to them; but don't worry, we explain those steps below. Setup - Icarus Verilog version v11-20210204 Please read the following important information before continuing. Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation. GNU GENERAL PUBLIC LICENSE Copyright (C) 1989, 1991 Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

We will not demonstrate every step of the installation procedure as it consists of mostly pressing "Next". But some steps are important. The first one is you have to make sure both of the options are selected in

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

The second important step is you have to make sure "Add executable folder(s) to the user PATH" option is selected in the step illustrated below: 🚳 Setup - Icarus Verilog version v11-20210204

Cancel

4.8 MB 158.8 MB

X

Next

Create a desktop shortcut Add executable folder(s) to the user PATH Next Cancel Back You can trust us and continue pressing "Next" until the end of the installation. After the installation, all the necessary folders which contain Icarus Verilog binaries should be added to the user PATH. You should test if everything is alright by typing the following commands on your preferred Windows terminal application: The result of those commands should be similar to the below figure:

After seeing the proper output of those commands, you are pretty much ready to go. You can just start

In this example, we are going to compile and simulate the following source code which is saved in a text file

To compile and simulate this piece of code, we are going to open a new terminal in the same folder as the

The first iverilog command is used to compile the given source file into an output file named example_ which can be simulated using the second vvp command. As you can see from the following figure, the correct

PS C:\Users\alperen\Desktop\example\example_1>

To test all the values in the above truth table, the following Verilog testbench is used, saved in a text file named "and_module_tb.v": `timescale 1 ns/10 ps module and_module_tb; localparam period = 20; reg a,b; reg[2:0] count = 2'b00; wire f; integer i;

and_module.v 11/12/2021 6:23 PM V File 1 KB and_module_tb.v 11/12/2021 6:25 PM V File 1 KB result.vcd 11/12/2021 6:26 PM VCD File 1 KB PowerShell 7 (x64)

11/12/2021 6:26 PM

Date modified

Size

3 KB

 \times

Type

File

Insert Replace

Once you have created a configuration file, you may load the file by simply giving it as a second argument gtkwave result.vcd conf.gtkw 4.3 Development Pipeline using a Makefile You may consider coming up with a Makefile to reduce the process of compilation, simulation and waveform

You are also recommended to add the make binary to the path for ease-of-use. 5 Useful Resources

You can visit the official website of Icarus Verilog, using the following link (as stated in the first section): https://steveicarus.github.io/iverilog

Icarus Verilog can come in handy if you are planning to use a low PC or if you have a thing for open-source software. For instance, a very old computer, rocking some lightweight Linux distro (e.g Puppy Linux [1]),

Figure 2: Downloading the installer The licenses for most software are designed to take away your I accept the agreement

Current selection requires at least 188,9 MB of disk space. Back Next Cancel

Which additional tasks should be performed?

✓ Install MinGW dependencies (DLL libraries)

I do not accept the agreement

Setup - Icarus Verilog version v11-20210204

Which components should be installed?

✓ Install GTKWave (x64)

elect Additional Tasks

Select the additional tasks you would like Setup to perform while installing Icarus Verilog, then click Next.



3.2

This is a simple module that takes two inputs a and b, ANDs them and writes the result to the wire f. To

0 0

1 0

0

1 0

1 1 1

f

0

test this simple module, we are going to use the following truth table of the AND gate:

and_module uut(.a(a), .b(b), .f(f)); initial begin \$dumpfile("result.vcd"); \$dumpvars; for (i = 0; i < 4; i++) begin

To compile and those two source files into one output file then simulate it, the following set of commands

The difference from the first example is, in the first command which instructs the binary *iverilog* to compile source files, "*.v" wildcard is used, which finds every Verilog source file in the folder. This is a shortcut

 ${a,b} = count;$ count += 1;#period;

end \$finish;

end

iverilog -o and_module *.v

instead of typing every source file's name one by one as:

iverilog -o and_module and_module.v and_module_tb.v

This situation is reported to the user by the following message:

endmodule

vvp and_module

are used:

'S C:\Users\alperen\Desktop\example> iverilog -o and_module *.v PS C:\Users\alperen\Desktop\example> vvp and_module VCD info: dumpfile result.vcd opened for output. and_module_tb.v:32: \$finish called at 8000 (10ps) PS C:\Users\alperen\Desktop\example> gtkwave .\result.vcd After compilation and simulation are complete, you can see that a new file named "result.vcd" is created.

Using the left panel, you can select the signals that you need to see and add them to the right panel using the "Append" button. After adding the signals, you can zoom in or scroll through the waveform illustrated GTKWave - \result.vcd File Edit Search Time Markers View Help 🔷 🔷 | From: 0 sec To: 80 ns Marker: 49100 ps | Cursor: 8500 ps ▼ <u>S</u>ST Signals Waves Time 由 🚠 and_module_tb a=1 📥 uut b = 0f = 0Type Signals

Ctrl+Q Quit

GTKWave - result.vcd

Open New Window

Read Verilog Stemsfile Read Tcl Script File

CC = iverilog FLAGS = -Walllibrary_input: *.v

Open New Tab

File Edit Search Time Markers View

Shift+Ctrl+R Reload Waveform Waves Export Close Ctrl+W Print To File Ctrl+P Grab To File Read Save File Ctrl+0 Ctrl+S Write Save File Write Save File As Shift+Ctrl+S Read Sim Logfile

From: 0 sec

To: 80 ns

Marker: -- | Cursor: 6300

You can download and install *make* for Windows using the following link: https://gnuwin32.sourceforge.net/packages/make.htm

See the man page for details. PS C:\Users\alperen> gtkwave -V GTKWave Analyzer v3.3.100 (w)1999-2019 BSI This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. PS C:\Users\alperen> compiling and simulating Verilog code. $\mathbf{3}$ Compiling and Running Your First Verilog Module Example 1 The first example will consist of a single module, which is going to print "Hello world!" to the console. This is not a very good use case in Verilog as it is used to describe hardware, however, this example is quite good for demonstrating the basic usage of Icarus Verilog.

output is generated as the result of the simulation: Name Date modified Size Type example_1 11/12/2021 6:31 PM 1 KB hello_world.v 11/12/2021 6:29 PM 1 KB PowerShell 7 (x64) C:\Users\alperen\Desktop\example\example_1> iverilog -o example_1 hello_world.v PS C:\Users\alperen\Desktop\example\example_1> vvp example_1 Hello world! Example 2 The second example consists of two source files; a Verilog module and a testbench to test the first one. This is a much more realistic use case in Verilog, and for our experiments, you will have to use at least two source files. The first source code is as follows, saved in a text file named "and_module.v": module and_module(input wire a, b, output wire f); and(f, a, b);

The following figure demonstrates the output of the mentioned commands: Name and_module

GTKWave - .\result.vcd

由品and_module_tb

Type | Signals

▼ <u>S</u>ST

File Edit Search Time Markers View Help

Signals

Time

From: 0 sec

Waves

VCD info: dumpfile result.vcd opened for output. and_module_tb.v:32: \$finish called at 8000 (10ps) This is a waveform file which includes the result of the simulation. To view this file, the following command is used: gtkwave result.vcd

When entered, a new GTKWave window will open which allows you to investigate the result of the simulation:

To: 80 ns

Marker: -- | Cursor: 0 sec

Append Insert Replace Þ Some Tips & Tricks Additional iverilog Parameters You may prefer to include the -Wall parameter while compiling your Verilog code to enable more warnings during the compilation. This parameter may come in handy while trying to figure out what's wrong with your code. The below figure summarizes an example usage and output of the -Wall paramater: iverilog -Wall -o and_module *.v warning: Some modules have no timescale. This may cause : confusing timing results. Affected modules are:

-- module and_module declared here: and_module.v:1

You may prefer to use a configuration file for GTKwave for ease-of-use. This approach allows you to get the same view after fixing your code and creating another waveform. After choosing which signals to use, the zoom level, etc., you should select "Write Save File As" from the "File" menu to create a configuration file.

Saving and Re-using GTKWave Configuration

Help

←

Ctrl+N

Ctrl+T

viewing into a single command. The following Makefile contains useful commands regarding the process:

\$(CC) \$(FLAGS) -o and_module *.v

gtkwave result.vcd conf.gtkw

vvp and_module

You can learn about the detailed usage of *iverilog* command, using the following link: https://steveicarus.github.io/iverilog/usage/command_line_flags.html

You can learn about the detailed usage of *vvp* command, using the following link: https://steveicarus.github.io/iverilog/usage/vvp_flags.html You can get a better idea about the detailed usage of Icarus Verilog, using the following link: https://steveicarus.github.io/iverilog/usage/getting_started.html You can learn about installing Icarus Verilog on different systems, using the following links: https://iverilog.fandom.com/wiki/Installation_Guide https://steveicarus.github.io/iverilog/usage/installation.html You can also check out this tutorial which helped us through section 4: https://usermanual.wiki/Document/IcarusVerilogGTKWaveguide.238990987/view You can download the sample files used in this tutorial using the following link: https://web.cs.hacettepe.edu.tr/~alperencakin/tutorial.zip

References [1] https://puppylinux.com/ [2] https://downloadtimecalculator.com/