Verilator Guide

with WSL2 Ubuntu

Sangyoon Oh

cs311-2025ta@postech.ac.kr

POSTECH

Installation Guide

POSTECH

What is Verilator

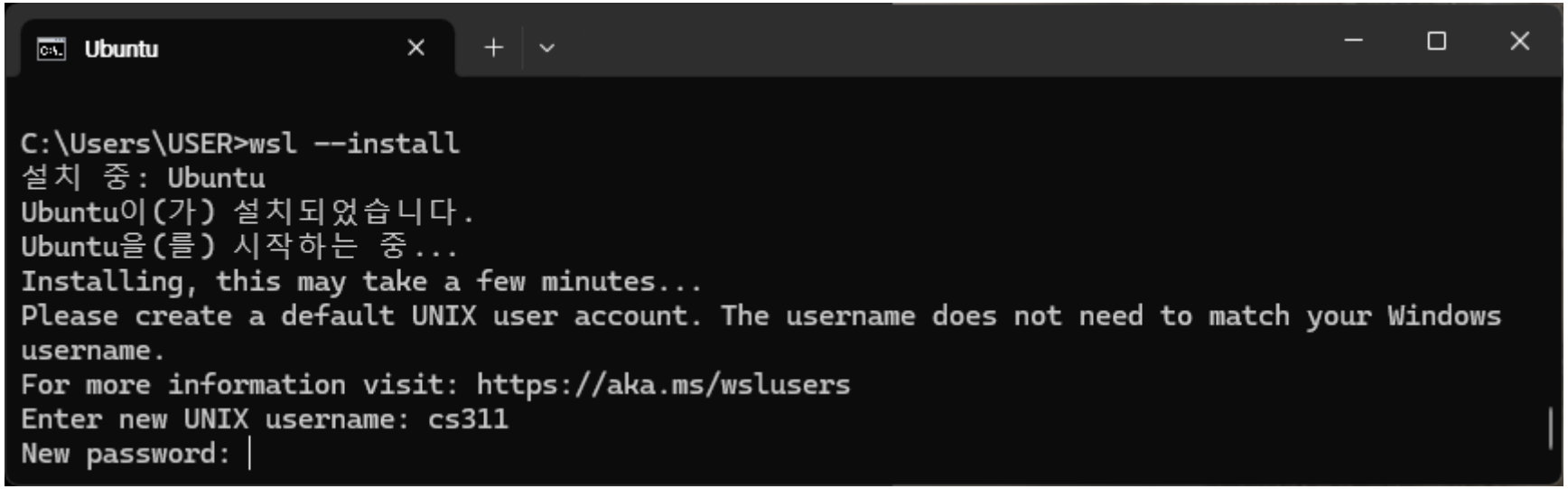
- Verilog/SystemVerilog simulator
- Compiles into multithreaded C++, or SystemC
- Widely used in industry and academy
- Developed on Linux & Mac OS
  - To use on Windows, you need WSL, Cygwin, or MinGW
- For this course, we use Mac or Windows(Cygwin, WSL) + Verilator.
  - You can use other OS or Linux supports for Windows if you want

Verilog for Window

0. Wsl install

- For Windows only
  - Open terminal
  - Install WSL by running “wsl --install”

You can install WSL this way on Windows 10 version 2004 or later (build 19041 or later) and Windows 11.
If not, install WSL in a different way.



  - Make sure that there are no spaces or Korean characters in the username.

1. Verilator install

- Run below to install prerequisites packages

```
sudo apt-get update
sudo apt-get install git help2man perl python3 make autoconf g++ flex bison ccache
```

- Run below to build verilator

```
git clone https://github.com/verilator/verilator
cd verilator
autoconf
./configure
sudo make -j `nproc`
sudo make install
cd ..
echo "export VERILATOR_ROOT=~/verilator" >> ~/.bashrc
source ~/.bashrc
```

Caution: You must use `, not '

1. Verilator install

- Run below to check if verilator is installed well.

```
verilator
```

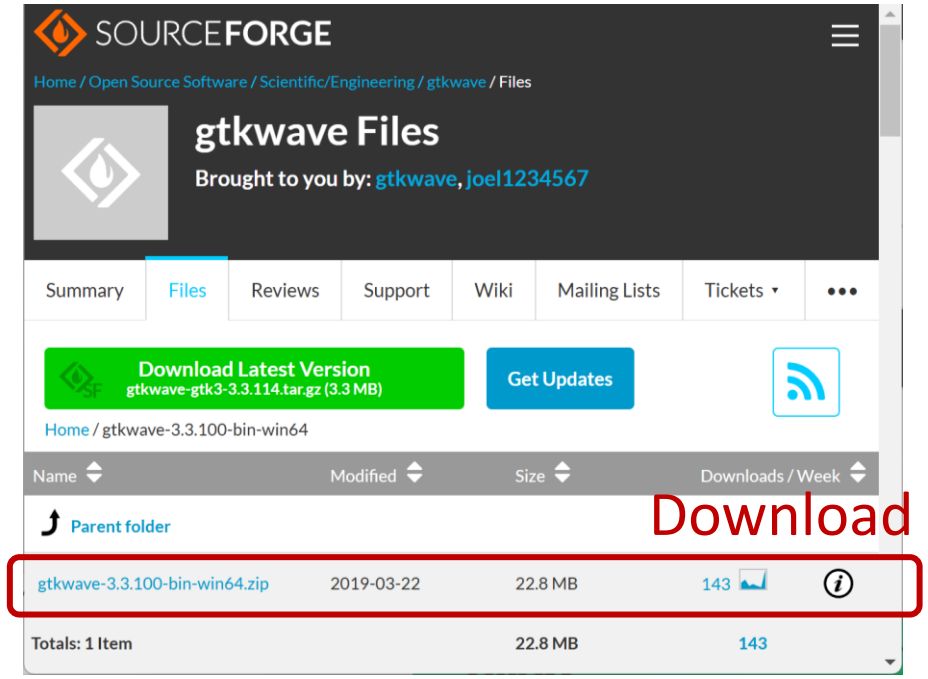
```
owner@DESKTOP-IBN9H8D ~
$ verilator
Usage:
        verilator --help
        verilator --version
        verilator --binary -j 0 [options] [source_files.v]... [opt_c_files.cpp/c/cc/a/o/so]
        verilator --cc [options] [source_files.v]... [opt_c_files.cpp/c/cc/a/o/so]
        verilator --sc [options] [source_files.v]... [opt_c_files.cpp/c/cc/a/o/so]
        verilator --lint-only -Wall [source_files.v]...
```

2. GTKWave install

- GTKWave is waveform viewer for Linux, MacOS and Windows

- For Windows, download https://sourceforge.net/projects/gtkwave/files/gtkwave-3.3.100-bin-win64/

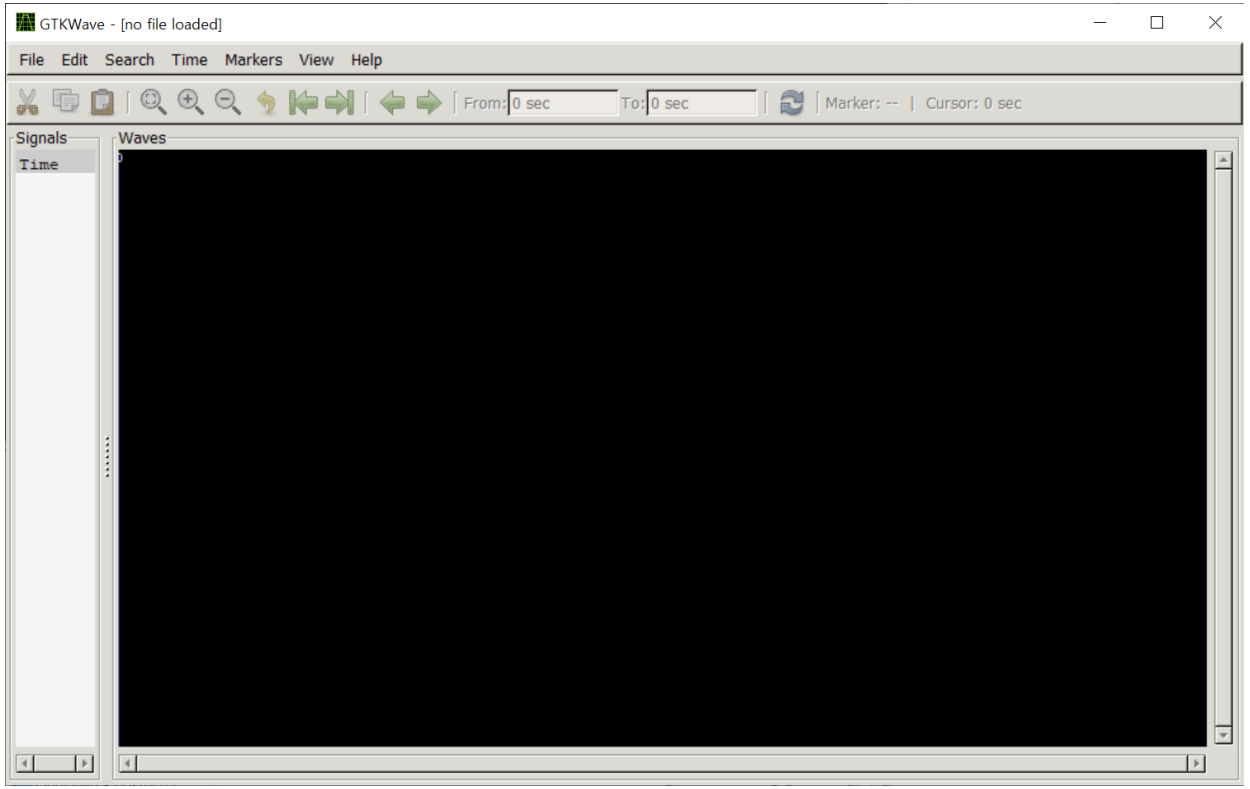
- Or else, check https://gtkwave.sourceforge.net/



Download this .zip file

2. GTKWave install

- Under “bin” folder, run “gtkwave.exe”



Verilator Simulation Guide

POSTECH

Simulation Directory

- .v : Verilog code
- sim_main.cpp : test bench code
- Makefile : build executable binary file

```
$ ls
Makefile  sim_main.cpp  top.v
```

Example : Hello World

- Go to “{workspace}/verilator/examples/make_hello_c”

```
owner@DESKTOP-IBN9H8D ~
$ cd verilator/examples/make_hello_c

owner@DESKTOP-IBN9H8D ~/verilator/examples/make_hello_c
$ ls
Makefile  sim_main.cpp  top.v
```

- Open and check each codes
  - Use your favorite code editor(VSCode, Notepad…)
  - WSL users can find {workspace} folder by running “explorer.exe . ”

Example : Hello World

- Build by “make”

```
owner@DESKTOP-IBN9H8D ~/verilator/examples/make_hello_c
$ make
-- Verilator hello-world simple example
-- VERILATE & BUILD --------
verilator -cc --exe --build -j top.v sim_main.cpp
make[1]: Entering directory '/home/owner/verilator/examples/make_hello_c/obj_dir'
ccache g++  -I.  -MMD -I/usr/local/share/verilator/include -I/usr/local/share/verilat
f-protection=none -Wno-bool-operation -Wno-shadow -Wno-sign-compare -Wno-tautological
```

- Go to “obj_dir”

```
owner@DESKTOP-IBN9H8D ~/verilator/examples/make_hello_c
$ ls
Makefile  obj_dir  sim_main.cpp  top.v

owner@DESKTOP-IBN9H8D ~/verilator/examples/make_hello_c
$ cd obj_dir

owner@DESKTOP-IBN9H8D ~/verilator/examples/make_hello_c/obj_dir
$ ls
Vtop.cpp  Vtop__ALL.a    Vtop__Syms.cpp                          Vtop___024root__DepSet_heccd7ead__0.cpp       Vtop__ver.d         sim_main.o          verilated_threads.o
Vtop.exe  Vtop__ALL.cpp  Vtop__Syms.h                            Vtop___024root__DepSet_heccd7ead__0__Slow.cpp  Vtop__verFiles.dat  verilated.d
Vtop.h    Vtop__ALL.d    Vtop___024root.h                        Vtop___024root__Slow.cpp                      Vtop_classes.mk     verilated.o
Vtop.mk   Vtop__ALL.o    Vtop___024root__DepSet_h84412442__0.cpp Vtop__pch.h                                   sim_main.d          verilated_threads.d
```

Example : Hello World

- Run Vtop

```
owner@DESKTOP-IBN9H8D ~/verilator/examples/make_hello_c/obj_dir
$ ./Vtop
Hello World!
- top.v:12: Verilog $finish
```

Example : Counter

- Go to “{workspace}/verilator/examples/make_tracing_c”

```
owner@DESKTOP-IBN9H8D ~/verilator/examples/make_hello_c
$ cd ..

owner@DESKTOP-IBN9H8D ~/verilator/examples
$ cd make_tracing_c

owner@DESKTOP-IBN9H8D ~/verilator/examples/make_tracing_c
$ ls
Makefile  Makefile_obj  input.vc  sim_main.cpp  sub.v  top.v
```

- Again, check each codes
- Run “make”

Example : Counter

- Code implementing counter

```
module sub
  (
   input clk,
   input reset_l
   );

   // Example counter/flop
   reg [31:0] count_c;
   always_ff @ (posedge clk) begin
      if (!reset_l) begin
         /*AUTORESET*/
         // Beginning of autoreset for uninitialized flops
         count_c <= 32'h0;
         // End of automatics
      end
      else begin
         count_c <= count_c + 1;
         if (count_c >= 3) begin
            // This write is a magic value the Makefile uses to make sure the
            // test completes successfully.
            $write("*-* All Finished *-*\n");
            $finish;
         end
      end
   end
```

Example : Counter

- Again, run Vtop

```
owner@DESKTOP-IBN9H8D ~/verilator/examples/make_tracing_c
$ cd obj_dir/

owner@DESKTOP-IBN9H8D ~/verilator/examples/make_tracing_c/obj_dir
$ ls
Vtop.cpp     Vtop__ALL.cpp   Vtop__TraceDecls__0__Slow.cpp           Vtop___024root__DepSet_h84412442__0__Slow.cpp  Vtop__v
Vtop.exe     Vtop__ALL.d     Vtop__Trace__0.cpp                      Vtop___024root__DepSet_heccd7ead__0.cpp        Vtop__v
Vtop.h       Vtop__ALL.o     Vtop__Trace__0__Slow.cpp                Vtop___024root__DepSet_heccd7ead__0__Slow.cpp  Vtop_cl
Vtop.mk      Vtop__Syms.cpp  Vtop___024root.h                       Vtop___024root__Slow.cpp                       sim_mai
Vtop__ALL.a  Vtop__Syms.h    Vtop___024root__DepSet_h84412442__0.cpp Vtop__pch.h                                    sim_mai

owner@DESKTOP-IBN9H8D ~/verilator/examples/make_tracing_c/obj_dir
$ ./Vtop
[1] Model running...

[1] clk=1 rstl=1 iquad=1234 -> oquad=1235 owide=3_22222222_11111112
[2] clk=0 rstl=0 iquad=1246 -> oquad=0 owide=0_00000000_00000000
[3] clk=1 rstl=0 iquad=1246 -> oquad=0 owide=0_00000000_00000000
[4] clk=0 rstl=0 iquad=1258 -> oquad=0 owide=0_00000000_00000000
[5] clk=1 rstl=0 iquad=1258 -> oquad=0 owide=0_00000000_00000000
[6] clk=0 rstl=0 iquad=126a -> oquad=0 owide=0_00000000_00000000
[7] clk=1 rstl=0 iquad=126a -> oquad=0 owide=0_00000000_00000000
[8] clk=0 rstl=0 iquad=127c -> oquad=0 owide=0_00000000_00000000
[9] clk=1 rstl=0 iquad=127c -> oquad=0 owide=0_00000000_00000000
[10] clk=0 rstl=1 iquad=128e -> oquad=128f owide=3_22222222_11111112
[11] clk=1 rstl=1 iquad=128e -> oquad=128f owide=3_22222222_11111112
[12] clk=0 rstl=1 iquad=12a0 -> oquad=12a1 owide=3_22222222_11111112
[13] clk=1 rstl=1 iquad=12a0 -> oquad=12a1 owide=3_22222222_11111112
[14] clk=0 rstl=1 iquad=12b2 -> oquad=12b3 owide=3_22222222_11111112
[15] clk=1 rstl=1 iquad=12b2 -> oquad=12b3 owide=3_22222222_11111112
[16] clk=0 rstl=1 iquad=12c4 -> oquad=12c5 owide=3_22222222_11111112
*-* All Finished *-*
- sub.v:29: Verilog $finish
[17] clk=1 rstl=1 iquad=12c4 -> oquad=12c5 owide=3_22222222_11111112
```

Ignore these values for now

Example : Counter

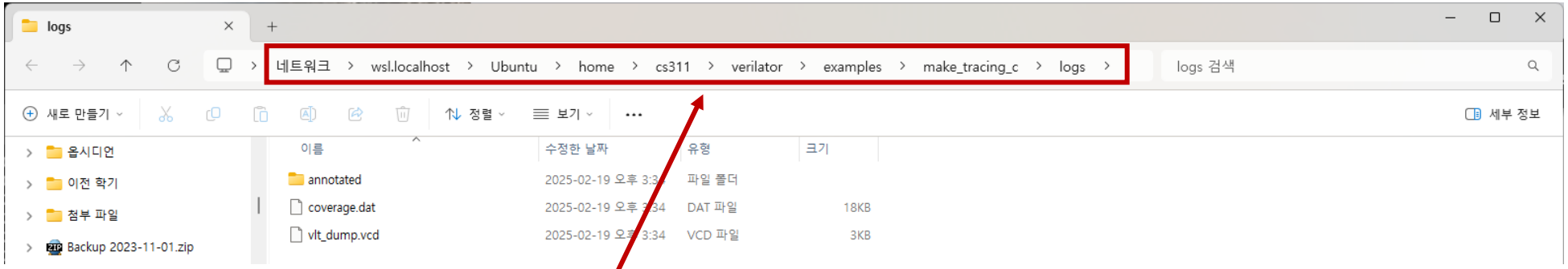
- Go to “make_tracing_c/logs”

```
owner@DESKTOP-IBN9H8D ~/verilator/examples/make_tracing_c/logs
$ ls
annotated  coverage.dat  vlt_dump.vcd
```

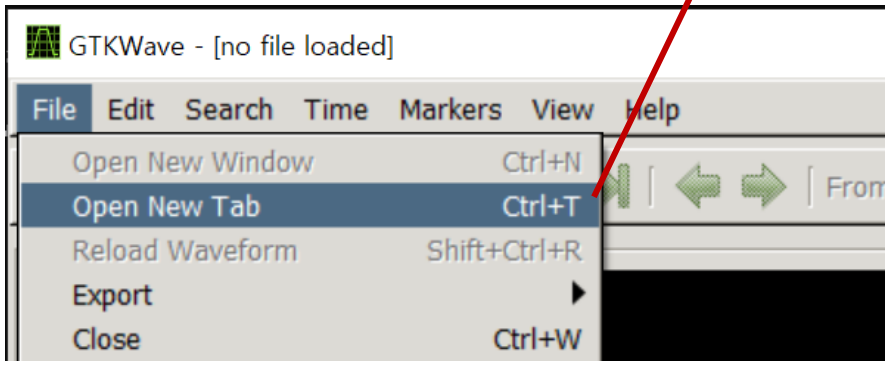
Example : Counter

- Check the path to “vlt_dump.vcd”

```
cs311@ECOSY:~/verilator/examples/make_tracing_c/logs$ explorer.exe .
```



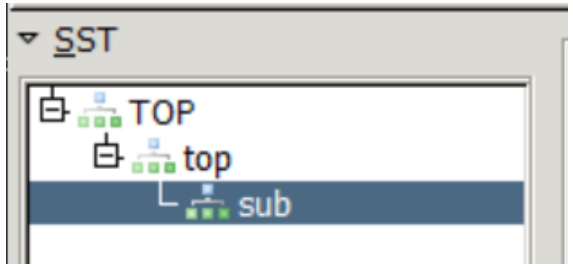
- Open “vlt_dump.vcd” with GTKWave



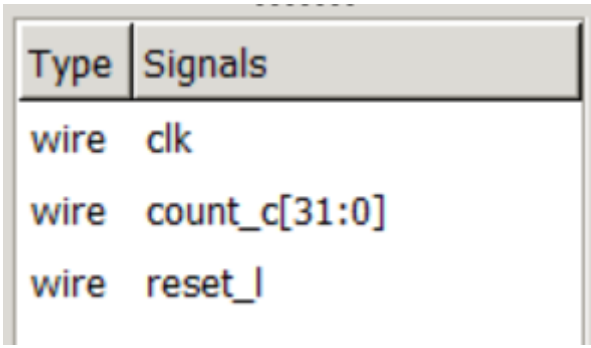
POSTECH

Example : Counter

- Open sub module

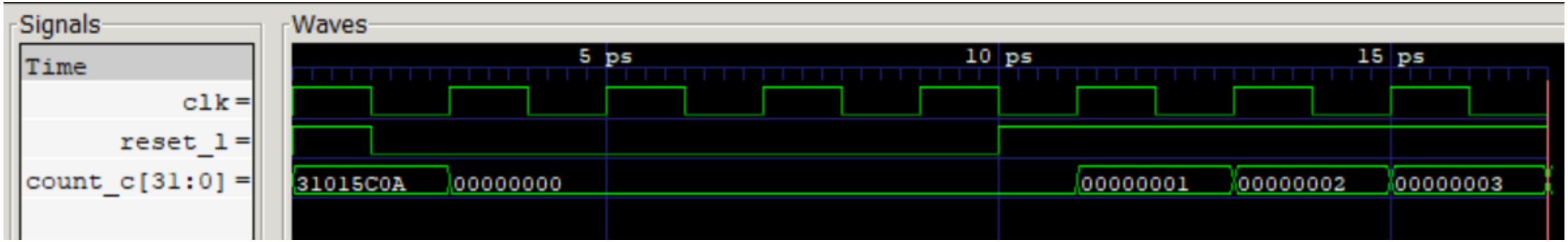


- Add clk, reset_l, count_c by double clicking



Example : Counter

- Now you can see the result



Debug

- For debugging, you have several options
  - gdb
    - https://www.sourceware.org/gdb/
    - https://verilator.org/guide/latest/simulating.html?highlight=debug
    - https://verilator.org/guide/latest/exe_verilator.html?highlight=debug#cmdoption-debug
    - For VSCode users, https://code.visualstudio.com/docs/cpp/cpp-debug
  - printf
  - Waveform