An Alternative for Vivado Simulation (Optional)[1]

Introduction

> ***Vivado*** is a very important software supporting ***Verilog language*** in *VE270 Intro. to Logic Design*. It is a very powerful software to logic circuit design. An alternative way is introduced, namely `VSCode` + `Linter` (for edition) + `iverilog` + `gtkwave` (for simulation), to write and simulate Verilog code. You can keep reading this manual if you are interested, and it is totally Okay if you ignore it. Please notice that Vivado still need to be used for implementation of the circuit on FPGA board .

Icarus Verilog and Gtkwave Installation

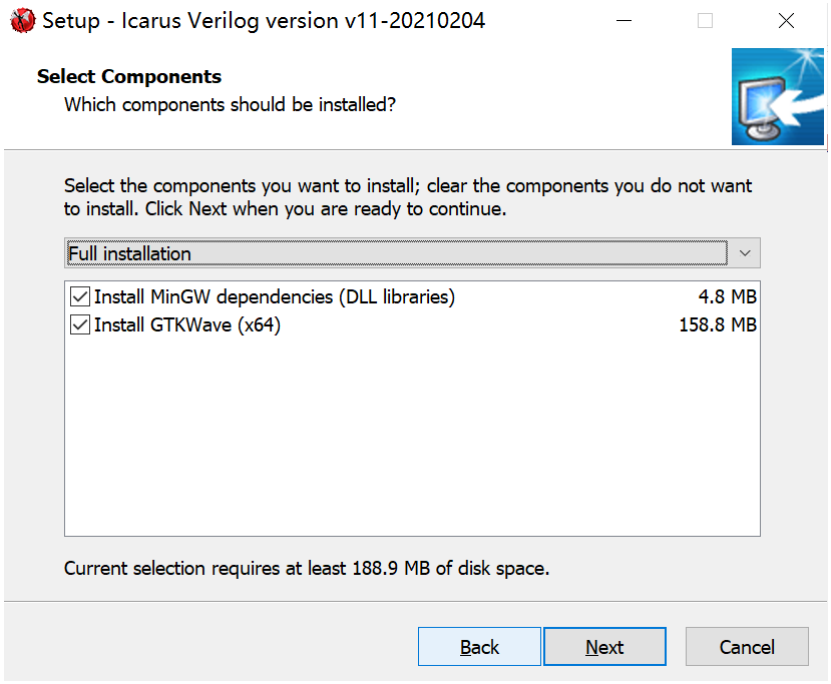
> ### Introduction:
>
> - Icarus Verilog is a free compiler implementation for the IEEE-1364 Verilog hardware description language. Icarus is maintained by Stephen Williams and it is released under the GNU GPL license.[2]
> - GTKWave is a fully featured GTK+ based wave viewer for Unix, Win32, and Mac OSX which reads LXT, LXT2, VZT, FST, and GHW files as well as standard Verilog VCD/EVCD files and allows their viewing.[3]

Windows Installation

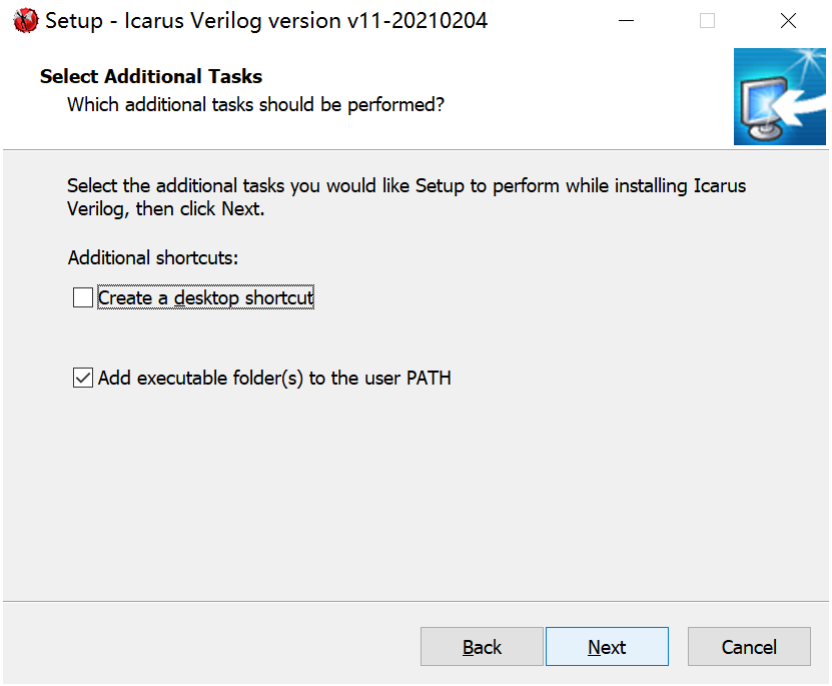
> **Installation Procedures**:
>
> - Go to Icarus Verilog website and download the proper package (iverilog-v11-20210204-x64_setup.exe)

- Accept the license and go through all the procedures. Please remember to check the following procedures
- **Attention**: `GTKWave` is included in the latest `Icarus Verilog` package.

- Remember to click `Install GTKWave (x64)` button (by default it is chosen)



- Remember to `Add executable folder(s) to the user PATH`



Alternative Installation Method

- You can also use `chocolatey` to install iverilog and gtkwave from terminal. `Chocolatey` in Windows resembles `apt-get` in Linux.

Check Installation:

- Go to `powershell.exe` to check the installation.
- Click `Win + X` and then choose `Powershell`
- Execute the following command-line instructions respectively

    - `iverilog -v` to print the version of icarus verilog
    - `gtkwave --version` to print the version of gtkwave
  - and you will see something similar as below

```
PS E:\unRar\ve270> iverilog -v
Icarus 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.

  You should have received a copy of the GNU General Public License along
  with this program; if not, write to the Free Software Foundation, Inc.,
  51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

D:\iverilog\bin\iverilog.exe: no source files.

Usage: iverilog [-EiSuvV] [-B base] [-c cmdfile|-f cmdfile]
                [-g1995|-g2001|-g2005|-g2005-sv|-g2009|-g2012] [-g<feature>]
                [-D macro[=defn]] [-I includedir] [-L moduledir]
                [-M [mode=]depfile] [-m module]
                [-N file] [-o filename] [-p flag=value]
                [-s topmodule] [-t target] [-T min|typ|max]
                [-W class] [-y dir] [-Y suf] [-l file] source_file(s)

See the man page for details.
PS E:\unRar\ve270> gtkwave --version
GTKWave Analyzer v3.3.108 (w)1999-2020 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 E:\unRar\ve270>
```

  - Successfully install in Windows now!

Windows Subsystem for Linux (WSL) installation

> **Before start**: since now the GUI of wsl is not supported for all windows system, installation of `iverilog in WSL` [4] has to be combined with installation of `GTKWave in Windows`, which you can refer to the previous part.

> **Preparation for installation**: change the software source of your `WSL` to speed up the downloading

- Type in `sudo vim /etc/apt/sources.list` to edit the file `/etc/apt/sources.list`
- Press `dd` until all the lines are cleaned
- Copy the mirrors of the software source to the file `/etc/apt/sources.list`
  - some mirrors: Aliyun and Tuna
  - not try `Ctrl + V` in WSL, instead just click the right (mouse) button
- Click `Esc` to exit the "edit mode"

- Press `Shift + Z + Z` to exit the file to the command line: quickly click `Z` twice while pressing `Shift`
- Type in `sudo apt-get update` in the command line to retrieve the new lists of package

> **Installation Procedures**:
>
> - Type in `sudo apt install make iverilog` to install `make` and `iverilog`

> **Check Installation**: type in `iverilog -v` and `make -v` respectively (shown below)

```
frankling@Frankling-LAPTOP:~$ iverilog -v
Icarus Verilog version 10.3 (stable) ()

Copyright 1998-2015 Stephen Williams

  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.

  You should have received a copy of the GNU General Public License along
  with this program; if not, write to the Free Software Foundation, Inc.,
  51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

iverilog: no source files.

Usage: iverilog [-ESvV] [-B base] [-c cmdfile|-f cmdfile]
                [-g1995|-g2001|-g2005|-g2005-sv|-g2009|-g2012] [-g<feature>]
                [-D macro[=defn]] [-I includedir]
                [-M [mode=]depfile] [-m module]
                [-N file] [-o filename] [-p flag=value]
                [-s topmodule] [-t target] [-T min|typ|max]
                [-W class] [-y dir] [-Y suf] [-l file] source_file(s)

See the man page for details.
frankling@Frankling-LAPTOP:~$ make -v
GNU Make 4.2.1
Built for x86_64-pc-linux-gnu
Copyright (C) 1988-2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

> - Successfully install in WSL now!

Linux Installation

> **Installation Procedures**:
>
> - Open the terminal and retrieve the newest package list `sudo apt-get update`
> - Type in `sudo apt install make iverilog gtkwave`

> **Installation Check**: type `gtkwave --version` in the terminal. If you also see some similar problems with regard to `canberra-gtk-module`, fix it by `sudo apt install libcanberra-gtk-module`. Similarly, check `iverilog` and `make` respectively by `iverilog -v` and `make --version`

> - Successfully install in Linux now!

Mac Installation 5

> **Installation Procedures**:
>
> - Ignore this step if you have already installed homebrew:
>   - Open the terminal and type `/bin/zsh -c "$(curl -fsSL https://gitee.com/cunkai/HomebrewCN/raw/master/Homebrew.sh)"`
>   - Choose mirror: ustc
>   - Enter your password whenever it asks you to do so
> - To install iverilog, type in terminal `brew install icarus-verilog`
> - To install gtkwave,
>   - Type in terminal `brew install gtkwave`
>   - Edit environment variables using vim:
>   - Type `vim ~/.zshrc` in terminal and press `Enter`
>   - Vim GUI will show up, navigate to the line shown below and press `I` on your keyboard to enter the edit mode of vim
>
> ```
>  9 # HomeBrew
> 10 export HOMEBREW_BOTTLE_DOMAIN=https://mirrors.ustc.edu.cn/homebrew-bottles
> 11 export PATH="/usr/local/bin:$PATH"
> 12 export PATH="/usr/local/sbin:$PATH"
> 13 # HomeBrew END
> 14
> 15 export PATH="$(pyenv root)/shims:$PATH"
> 16
> 17
> 18 #export LANGUAGE="en_US.UTF-8"
> 19 #export LANG=en_US:zh_CN.UTF-8
> 20 #export LC_ALL=C
> 21
> 22 source ~/.bash_profile
> 23
> ```
>
>   - Copy this line `source ~/.bash_porfile` using `Ctrl+Shift+V` and press `esc` to escape from edit mode
>   - Press `:` + `w` + `q` + `Enter` to save your changes and leave from Vim GUI
>   - Now you are back in terminal again type `vim ~/.bash_profile` in terminal and press `Enter`
>   - Vim GUI will show up again, navigate to the line shown below and press `I` on your keyboard to enter the edit mode of vim
>
> ```
>  9 # HomeBrew
> 10 export HOMEBREW_BOTTLE_DOMAIN=https://mirrors.ustc.edu.cn/homebrew-bottles
> 11 export PATH="/usr/local/bin:$PATH"
> 12 export PATH="/usr/local/sbin:$PATH"
> 13 # HomeBrew END
> 14
> 15
> 16 export PATH="$(pyenv root)/shims:$PATH"
> 17
> 18 export SCALA_HOME="/usr/local/scala-2.13.4"
> 19 export PATH=$PATH:$M2_HOME/bin:$SCALA_HOME/bin
> 20
> 21 export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-15.0.1.jdk/Contents/
>    Home
> 22 export PATH=$JAVA_HOME/bin:$PATH:.
> 23 export CLASSPATH=$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar:.
> 24
> 25 export PATH="/Applications/gtkwave.app/Contents/Resources/bin:$PATH"
> 26
> ```
>
>   - Copy this line `export PATH="/Applications/gtkwave.app/Contents/Resources/bin:$PATH"` using `Ctrl+Shift+V` and press `esc` to escape from edit mode

        - Press `:`+`w`+`q`+`Enter` to save your changes and leave from Vim GUI
    - After finishing previous steps, if you want to open gtkwave simulation result directly from terminal, Enter `open xxx.vcd` or `open -a gtkwave xxx.vcd`, where `xxx` should be figured out by yourself.

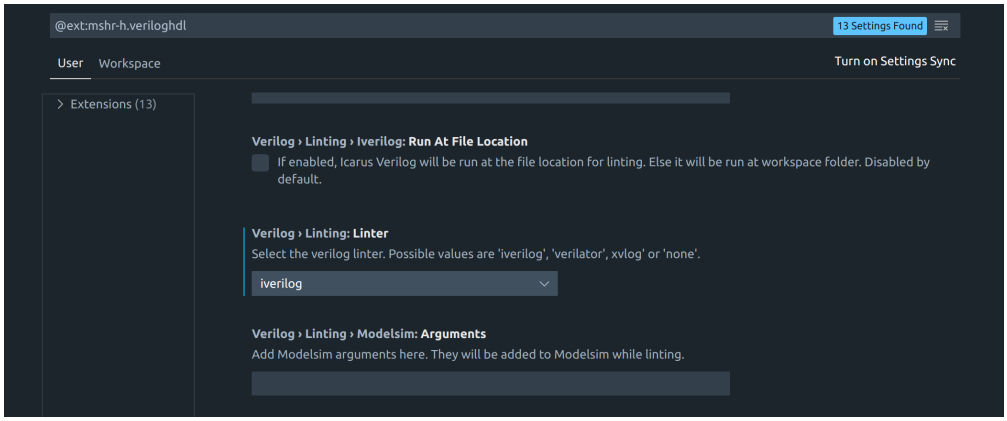
> **Check Installation**: type `iverilog` in the terminal. If you see its user guide popping up in terminal.....

> Successfully install in Mac now!

Programming in VSCode

> **Useful Verilog Extensions for VSCode** : 6

- `Verilog-HDL/SystemVerilog/Bluespec SystemVerilog support for VS Code`
    - go to the setting to customize by pressing `Ctrl + ,`
    - search `@ext:mshr-h.veriloghdl` or just go to its extension setting
    - choose `iverilog` in `Verilog > Linting: Linter`
    - reload the windows, then it can check the syntax for `Verilog code`



- `verilog formatter` , `Verilog Snippet` and .....

Simple Instructions on how to use Iverilog and GTKWave 7

> **Sample Code for D Flip-flop and testbench** : create a source code file `dff.v` and a test code file `dff_tb.v` in the same fold. The codes shown below are original codes accepted by verilog. The following procedures will show how to adapt them to `iverilog` and `gtkwave`. The general idea is to change the test code `dff_tb.v`

```
// dff.v
module dff (
    input D,
    input clk,
    output reg Q
);
    always @(posedge clk) begin
        Q <= D;
    end
endmodule
```

```
// dff_tb.v
module dff_tb;
    parameter half_period=50;

    reg Din;
    wire Qout;
    reg clock;

    dff UUT (.D(Din),.clk(clock),.Q(Qout));


    initial begin
        #0 Din = 1'b0; clock = 1'b0;
        #25 Din = 1'b1;
        #100 Din = 1'b0;
    end

    always #half_period clock = ~clock;
    initial #300 $stop;

endmodule
```

> - Include the module source code in the beginning of testbench code. In this case, there is only one module source code `dff.v` and only one testbench code `dff_tb.v`
> - Include the code that generates `.vcd` file (wave file)
> - Pay attention to `$dumpfile("wave.vcd")` and `$dumpvars(0, dff_tb)` (see notation below)

```
// the new dff_tb.v file

// include the module source code
`include "dff.v"

module dff_tb;
    parameter half_period=50;

    reg Din;
    wire Qout;
    reg clock;

    /*iverilog*/
```

```
    initial begin
        // here you define "wave.vcd" actually is the name of the generated wave
file
        $dumpfile("wave.vcd");
        // keep consistent here: the second entry should be the same name of this
testbench module "dff_tb"
        $dumpvars(0, dff_tb);
    end

    dff UUT (.D(Din),.clk(clock),.Q(Qout));


    initial begin
        #0 Din = 1'b0; clock = 1'b0;
        #25 Din = 1'b1;
        #100 Din = 1'b0;
    end

    always #half_period clock = ~clock;
    initial #300 $stop;

endmodule
```

> Some instructions of iverilog and gtkwave:
>
> - `iverilog -o test dff_tb.v`: you can treat `iverilog` like `g++` in C++ language. It complies `dff_tb.v` and generates the executable file `test`
> - `vvp -n ./test -lxt2`: `./test` to execute `test` and `vvp` to generate the wave file
> - `gtkwave ./wave.vcd`: open the generated wave file by `GTKWave`. Notice: `./wave.vcd` is the name you define in the testbench file `dff_tb.v`
> - for more details, you can search it online or type `man iverilog`/`man vvp` to see the built-in documentation
> - To sum up, for simulation of D Flip-flop here, you need to successively execute the following instructions in the same fold of your source code

```
iverilog -o test dff_tb.v
vvp -n ./test -lxt2
gtkwave ./wave.vcd
```

Further Improvement

Windows

> **For windows users**, you can write them in `Makefile` file.
>
> For example, copy the below code in a new file `Makefile`, open the terminal (`Ctrl + J` in VSCode) and type in `mingw32-make all`. You will see that it executes automatically (shown below). To clean the previous files, run `mingw32-make clean`.
>
> Install `GNU make`. For example, you can execute `mingw32-make` as far as you install `MinGw` before in VG101.

```
all:test
    vvp -n ./test -lxt2
    gtkwave ./wave.vcd
test: dff.v dff_tb.v
    iverilog -o test dff_tb.v
clean:
    del /p test wave.vcd
```

```
PS E:\unRar\ve270> mingw32-make all
iverilog -o test dff_tb.v
vvp -n ./test -lxt2
LXT2 info: dumpfile wave.vcd opened for output.
dff_tb.v:28: $stop called at 300 (1s)
gtkwave ./wave.vcd

GTKWave Analyzer v3.3.108 (w)1999-2020 BSI

LXTLOAD | 6 facilities
LXTLOAD | Read 1 block header OK
LXTLOAD | [0] start time
LXTLOAD | [300] end time
LXTLOAD |
LXTLOAD | Finished building 6 facs.
LXTLOAD | Merging in 3 aliases.
LXTLOAD | Building facility hierarchy tree.
LXTLOAD | Sorting facility hierarchy tree.
WM Destroy
PS E:\unRar\ve270> mingw32-make clean
del /p test wave.vcd
E:\unRar\ve270\test，要删除(Y/N)吗？ Y
E:\unRar\ve270\wave.vcd，要删除(Y/N)吗？ Y
```

Mac/Linux

> For Linux/Mac users, since you can't open the gtkwave GUI, you have to rely on gtkwave in Windows, make sure you can execute `make` in the terminal.
>
> In the same folder of source code, create a file named `Makefile` without any file suffixes. Copy the following into it.

```
all:test
    vvp -n ./test -lxt2
    gtkwave ./wave.vcd
test: dff.v dff_tb.v
    iverilog -o test dff_tb.v
clean:
    rm -rf test wave.vcd
```

> Further explanation:
>
> It will be taught carefully in VE280, so no detailed explanation here.

> - `test: dff.v dff_tb.v`: replace your all source code name after the colon. For example, in D Flip-flop, we only have `dff.v` and `dff_tb.v`
> - `wave.vcd` in `gtkwave ./wave.vcd` and `rm -rf test wave.vcd`: the name of wave file defined in the testbench file (`dff_tb.v`).
> - `test` in `vvp -n ./test -lxt2`, `iverilog -o test dff_tb.v` and `rm -rf test wave.vcd`: the executable file generated by `iverilog` and keep consistent in these three commands

> How to execute in the Mac/Linux environment: open the terminal
>
> - `make test`: only executes `iverilog -o test dff_tb.v`
> - `make all`: first execute `make test` and then `vvp -n ./test -lxt2` and `gtkwave ./wave.vcd`. So, in most cases, you directly type in `make all` in the terminal. It will automatically launch `GTKWave`.
> - `make clean`: delete the files `test` and `wave.vcd`. For safety before every `make all`, execute `make clean` to clean the out-of-date file
> - for demo, please see in `WSL` part as they are similar.

WSL

> For WSL users, you can refer to `Mac/Linux` part to understand the principals, but you may change `Makefile` a little bit
>
> Change `gtkwave ./wave.vcd` to `powershell.exe gtkwave ./wave.vcd`

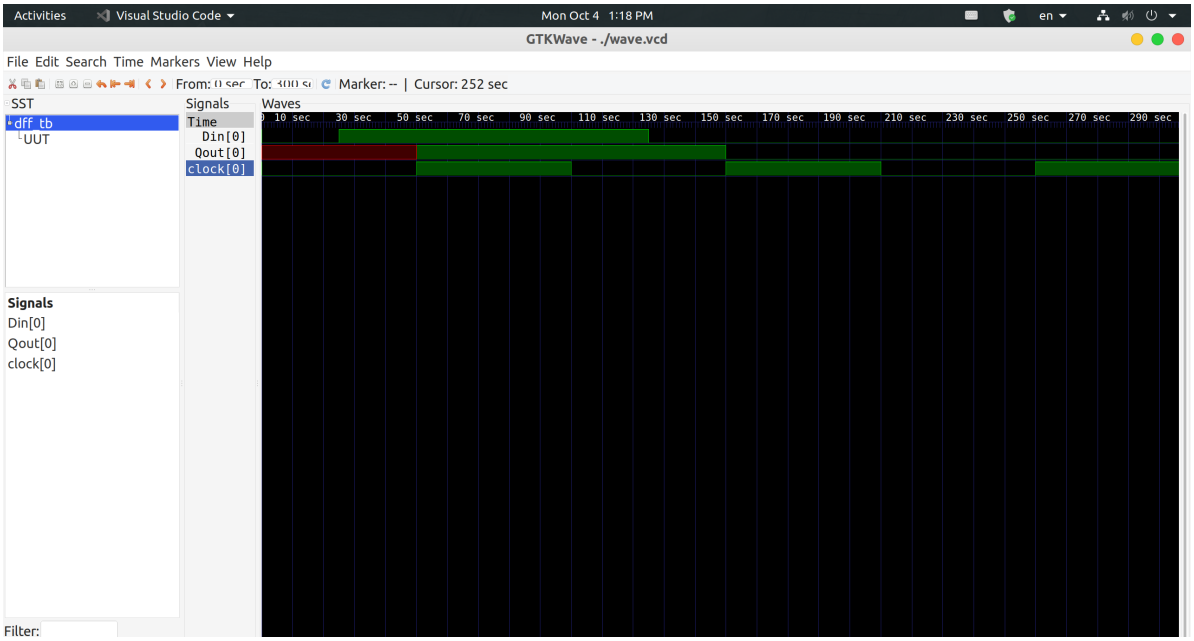
```
all:test
    vvp -n ./test -lxt2
    powershell.exe gtkwave ./wave.vcd
test: dff.v dff_tb.v
    iverilog -o test dff_tb.v
clean:
    rm -rf test wave.vcd
```

```
frankling@Frankling-LAPTOP:/mnt/e/unRar/ve270$ make clean
rm -rf test wave.vcd
frankling@Frankling-LAPTOP:/mnt/e/unRar/ve270$ make all
iverilog -o test dff_tb.v
vvp -n ./test -lxt2
LXT2 info: dumpfile wave.vcd opened for output.
powershell.exe gtkwave ./wave.vcd

GTKWave Analyzer v3.3.108 (w)1999-2020 BSI

LXTLOAD | 6 facilities
LXTLOAD | Read 1 block header OK
LXTLOAD | [0] start time
LXTLOAD | [300] end time
LXTLOAD |
LXTLOAD | Finished building 6 facs.
LXTLOAD | Merging in 3 aliases.
LXTLOAD | Building facility hierarchy tree.
LXTLOAD | Sorting facility hierarchy tree.
WM Destroy
frankling@Frankling-LAPTOP:/mnt/e/unRar/ve270$
```

GTKWave: Wave Viewer



- Click `dff_tb` under `SST`
- Click `Din[0]` ... under `Signals`
- Then you can see the green line showing the simulation result now!
- Here is the sample simulation result for D Flip-flop

Further Exploration

> To create, manage, simulate, implement and write-bitstream verilog without opening vivado, you can refer to `Makefile` and the project folder below.

> - Give special credits to VE270-SU21 TA Wu Qinhang

```
# proj=ve270lab5

##### Examples

## make build-project proj=ve270lab5
## make sim proj=ve270lab5 tb=clock_tb
## make write-bitstream proj=ve270lab5
## make program-fpga bs=bitstream_files/ve270lab5.bit

##### VE270-SU21 Verilog Workshop
##### Editer: VE270-SU21 TA Wu Qinhang
##### Credit: VE427-SU21 TA Yuan Yichao
##### Credit: UCB EECS151-FA20 Teaching Group
#####

VERILOG_SRCS = src/*.v

.PHONY: build-project
$(vivado_proj_file) build-project: $(VERILOG_SRCS)
        vivado -mode batch -source scripts/build_project.tcl -tclargs $(proj)

.PHONY: sim
sim: $(vivado_proj_file)
        vivado -mode batch -source scripts/sim.tcl -tclargs $(proj) $(tb)

.PHONY: impl
impl: $(vivado_proj_file)
        vivado -mode batch -source scripts/impl.tcl -tclargs $(proj)

.PHONY: write-bitstream
write-bitstream: $(vivado_proj_file)
        vivado -mode batch -source scripts/write_bitstream.tcl -tclargs $(proj)

.PHONY: program-fpga
program-fpga:
        vivado -mode batch -source scripts/program_fpga.tcl -tclargs $(bs)

.PHONY: clean
# "make clean" won't remove your project folders
clean:
        rm -rf *.log *.jou *.str *.tar
```

名称	修改日期	类型	大小
.git	2021/10/4 21:16	文件夹	
bitstream_files	2021/7/13 22:11	文件夹	
constrs	2021/10/4 21:15	文件夹	
lib	2021/10/4 21:15	文件夹	
scripts	2021/10/4 21:15	文件夹	
sim	2021/10/4 21:15	文件夹	
src	2021/10/4 21:15	文件夹	
.gitignore	2021/10/4 21:15	文本文档	1 KB
Makefile	2021/10/4 21:15	文件	2 KB

---

1. Written by frankling@sjtu.edu.cn and lauren_you.wu@sjtu.edu.cn ↩
2. Retrieved from http://bleyer.org/icarus/ "Icarus Verilog website" ↩
3. Retrieved from http://gtkwave.sourceforge.net/ ↩
4. Instructions for `WSL` installation: install by Powershell or manual installation ↩
5. https://zhuanlan.zhihu.com/p/357988583 MacOS tutorial ↩
6. VSCode can be installed from Visual Studio Code ↩
7. https://zhuanlan.zhihu.com/p/95081329 iverilog + gtkwave tutorial ↩