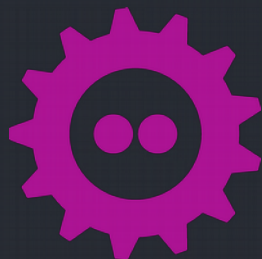




TerosHDL

an open source HDL IDE



FOSDEM 2021

Ismael Pérez

Carlos Alberto Ruiz

Alfredo Sáez



1
2
3
4
5
6
7
8
9
10
11
12
13

Index:

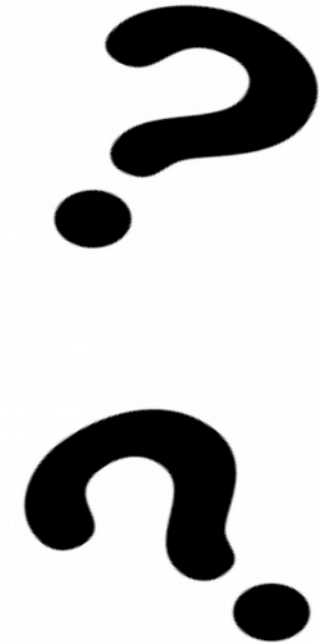
1. What is TerosHDL?
2. The team
3. Structure
4. Open Source
5. Requisites & configuration
6. TerosHDL VScode flavour
7. We need help!
8. Thanks



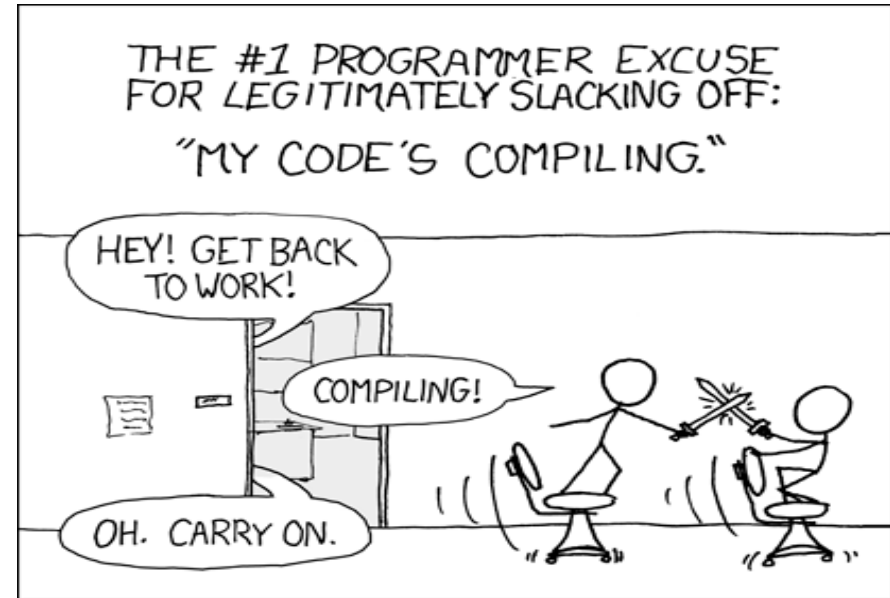
An open source HDL IDE

- Based in Atom
- Based in Vscode
- Based in...

Software tools for Hardware designers



- **Non-intrusive**
Keep your files and folders structure
- **Closing the gap with software tools**
- **Multi platform**
Linux, Mac, Windows
- **From begginers to experts**
- **Task automation**
Do not repeat loads of commands



Carlos Alberto:

- DSP FPGA engineer in the company SRS (Software Radio Systems)



Alfredo:

- Software engineer

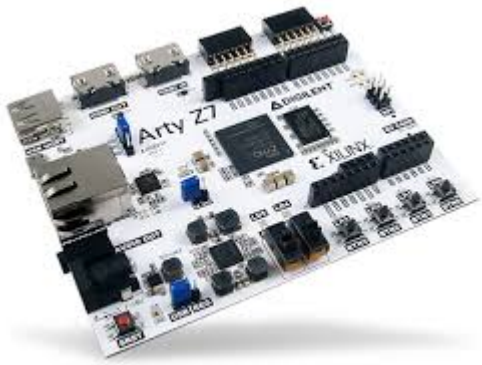


Ismael:

- FPGA & electronics engineer



- Multiple boards, multiple IDEs.



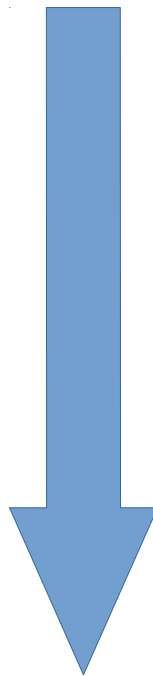
Homemade scripts

TerosHDL:

- Atom plugin
- Python backend

TerosHDL:

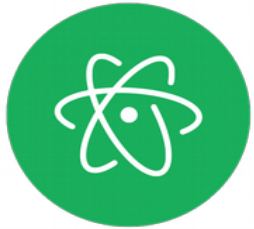
- VSCode plugin
- JS backend
- Self contained
- Multi platform



Visual Studio Code



TerosHDL Atom Flavour



- Version 1.1.9
- Atom store
- Only bug fixing
- Limited features
- VHDL

TerosHDL VSCode Flavour

- Version 0.1.0
- Newest editor features
- VHDL & Verilog/SV
- Ongoing development



Thanks!

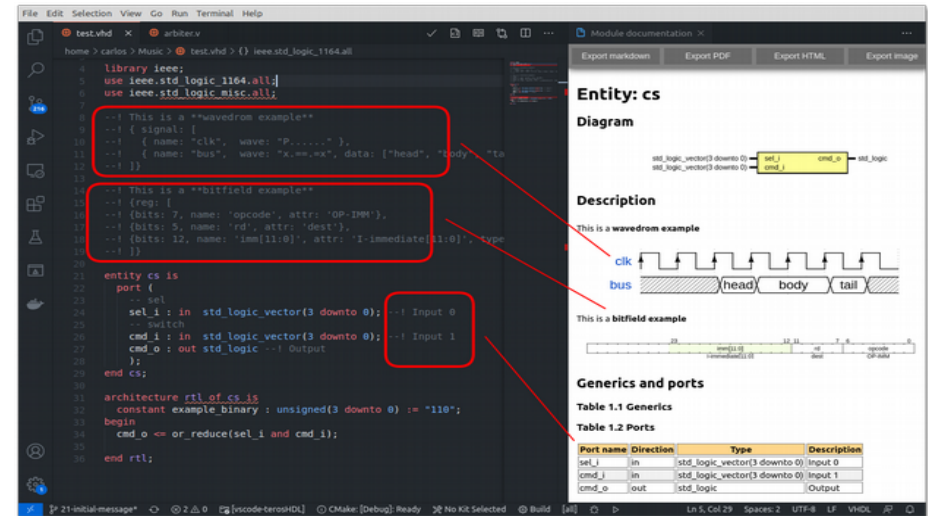
- Verilog HDL/SystemVerilog of mshr-h
- VUnit (<https://vunit.github.io/>)
- VHDL Style Guide (VSG) (<https://github.com/jeremiah-c-leary/vhdl-style-guide>)
- Wavedrom (<https://github.com/wavedrom/wavedrom>)
- GHDL (<http://ghdl.free.fr/>)
- Tree-sitter-VHDL, Tree-sitter-Verilog



GHDL



- Linter.
- Documentation generator.
- Go to definition.
- Templates.
- Snippets.
- Formatter.
- State machine viewer.
- State machine designer.
- Project manager.
- Dependencies viewer.



- Documenter with Wavedrom and Bitfield support.

```
4 library ieee;
5 use ieee.std_logic_1164.all;
6 use ieee.std_logic_misc.all;
7
8 --! This is a **wavedrom example**
9 --! { signal: [
10 --!   { name: "clk", wave: "P....." },
11 --!   { name: "bus", wave: "x.==.x", data: ["head", "body", "tail"] }
12 --! ]}
13
14 --! This is a **bitfield example**
15 --! {reg: [
16 --!   {bits: 7, name: 'opcode', attr: 'OP-IMM'},
17 --!   {bits: 5, name: 'rd', attr: 'dest'},
18 --!   {bits: 12, name: 'imm[11:0]', attr: 'I-immediate[11:0]', type
19 --!   ]}
20
21 entity cs is
22 port (
23   -- sel
24   sel_i : in std_logic_vector(3 downto 0); --! Input 0
25   -- switch
26   cmd_i : in std_logic_vector(3 downto 0); --! Input 1
27   cmd_o : out std_logic --! Output
28 );
29 end cs;
30
31 architecture rtl of cs is
32   constant example_binary : unsigned(3 downto 0) := "110";
33 begin
34   cmd_o <= or_reduce(sel_i and cmd_i);
35 end rtl;
```

Entity: cs

Diagram

std_logic_vector(3 downto 0) sel_i cmd_o std_logic
std_logic_vector(3 downto 0) cmd_i

Description

This is a wavedrom example

clk
bus

This is a bitfield example

23 12 11 7 6 0
imm[11:0] rd opcode
I-immediate[11:0] dest OP-IMM

Generics and ports

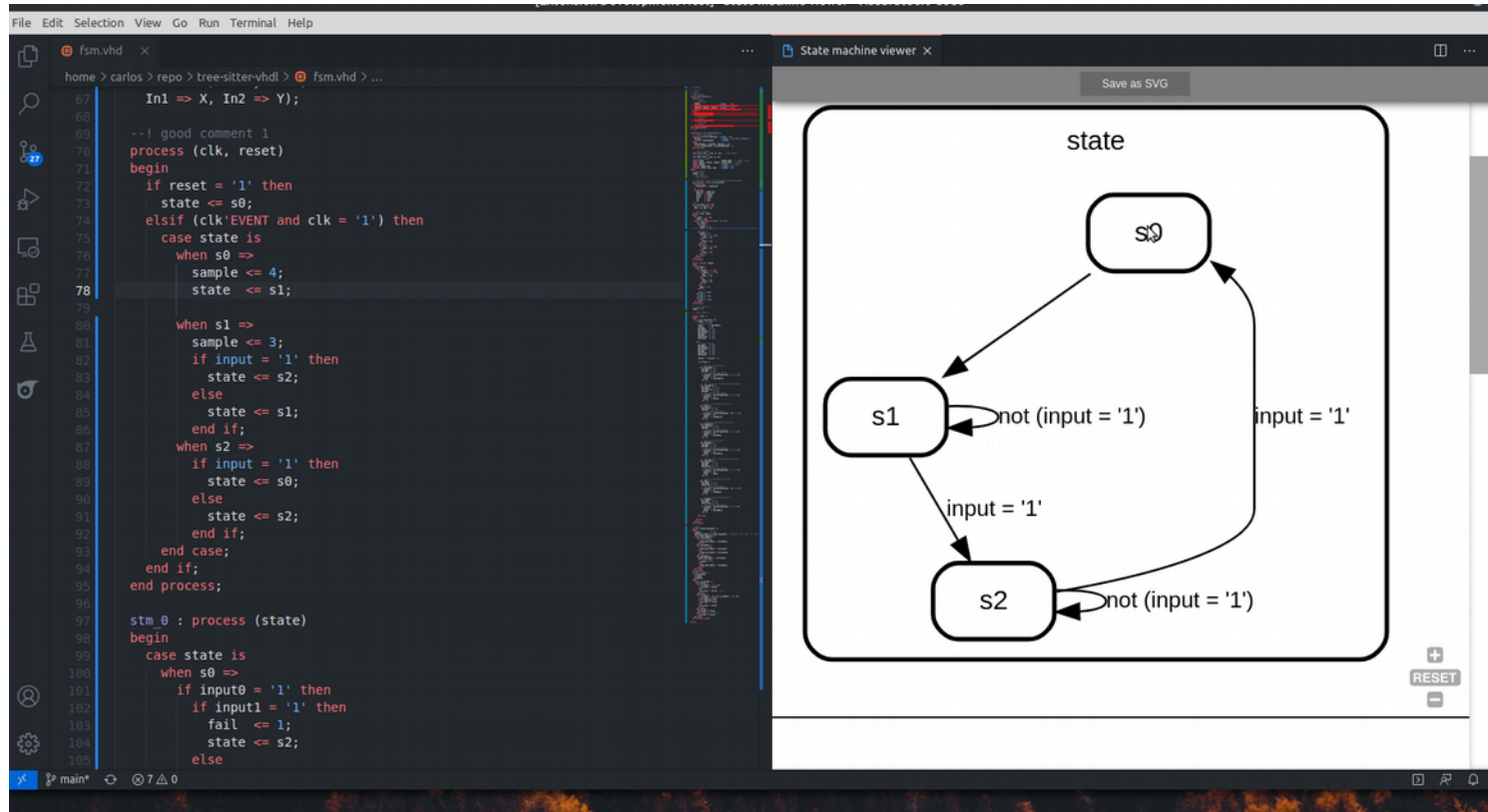
Table 1.1 Generics

Table 1.2 Ports

Port name	Direction	Type	Description
sel_i	in	std_logic_vector(3 downto 0)	Input 0
cmd_i	in	std_logic_vector(3 downto 0)	Input 1
cmd_o	out	std_logic	Output



- You can navigate in the states and go to the code.



Features: project manager

- Currently only Vunit is supported

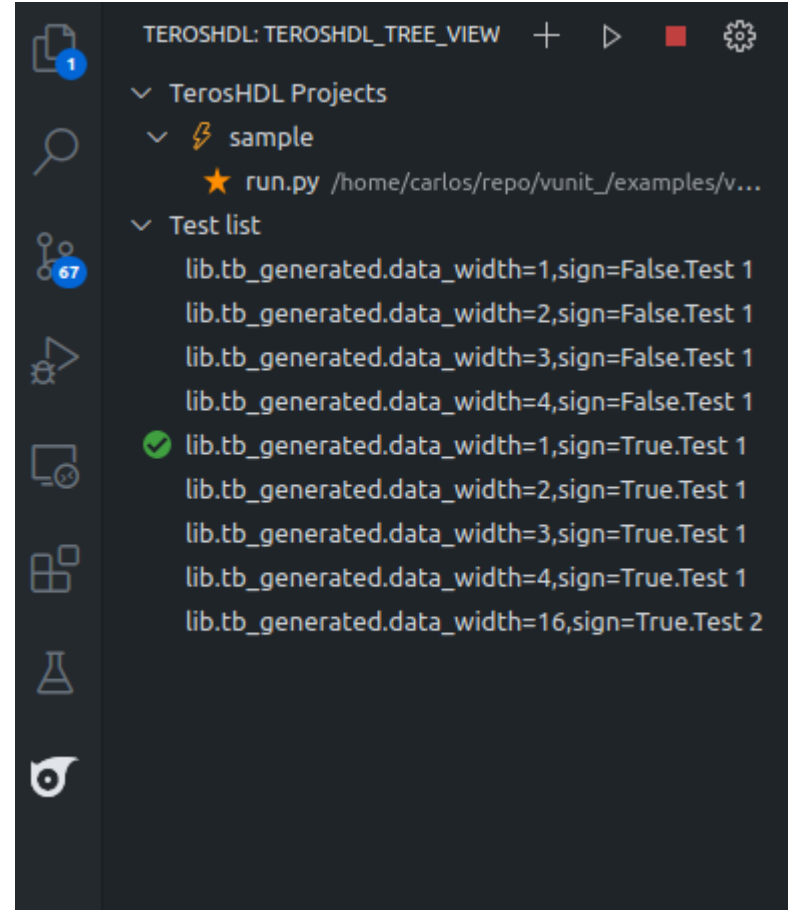
Tools configuration

Configure your tools.

General	GHDL
GHDL	GHDL installation path. <input type="text"/>
Icarus	Analyze options. Extra options used for the GHDL analyze stage (ghdl -a). <input type="text"/>
ModelSim	<input type="text"/>
Xsim	Run options. Extra options used when running GHDL simulations (ghdl -r). <input type="text"/>
VUnit	
Vivado	
Riviera-PRO	
Active-HDL	

Apply and close Cancel

TerosHDL




TEROSHDL: TEROSHDL_TREE_VIEW

- TeroshDL Projects
 - sample
 - run.py /home/carlos/repo/vunit_/examples/v...
- Test list
 - lib.tb_generated.data_width=1,sign=False.Test 1
 - lib.tb_generated.data_width=2,sign=False.Test 1
 - lib.tb_generated.data_width=3,sign=False.Test 1
 - lib.tb_generated.data_width=4,sign=False.Test 1
 - ✓ lib.tb_generated.data_width=1,sign=True.Test 1
 - lib.tb_generated.data_width=2,sign=True.Test 1
 - lib.tb_generated.data_width=3,sign=True.Test 1
 - lib.tb_generated.data_width=4,sign=True.Test 1
 - lib.tb_generated.data_width=16,sign=True.Test 2



<https://terostechnology.github.io/>



Search docs

ABOUT

What is TerosHDL?

Installing

Requisites

Testimonials

Team

Contact

Donate

CONFIGURATION

General settings

Linters

» Installing [Edit on GitHub](#)

All the tools are automatically managed by TerosHDL vscode plugin. But there are some requisites you have to satisfy to enable the all the functionalities. You can search for `TerosHDL` inside VSCode in the extensions tab or Launch VS Code Quick Open `Ctrl+P`, paste the following command, and press enter: `ext install teros-technology.teroshdl`

Requisites

- [Visual Studio Code](#)
- [Python3](#)
- [HDL simulator](#)
- [Vunit](#)

Most features work without external Visual Studio Code dependencies but some special features require manually installed software by the user.

Feature	Requisite
Linter	HDL simulator
Dependencie viewer	Python3
Project manager	Vunit

Visual Studio Code

[Download VSCode](#)

CLI Ubuntu installation example:



Download the VSCode beta version:

<https://marketplace.visualstudio.com/items?itemName=teros-technology.teroshdl>

Documentation, features, configuration:

<https://terostechnology.github.io/>

Feedback!

New features?

Bugs?

Donations.



Many thanks from Teros Team :)



www.terostech.com/teroshdl



<https://github.com/TerosTechnology>



[**terostechology@gmail.com**](mailto:terostechology@gmail.com)



<https://twitter.com/terostech>

