



TerosHDL

an open source HDL IDE

Carlos Alberto Ruiz Naranjo
Ismael Pérez Rojo



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

Carlos Alberto:

Open RAN FPGA engineer in
the company SRS (Software Radio Systems)



Ismael:

FPGA & electronics
Engineer (Quside)



- Editor para VHDL, Verilog y SV
- Interfaz gráfica para herramientas de FPGA/ASIC
- Ayuda a la edición de código
- Extensión para VSCode y VSCodium



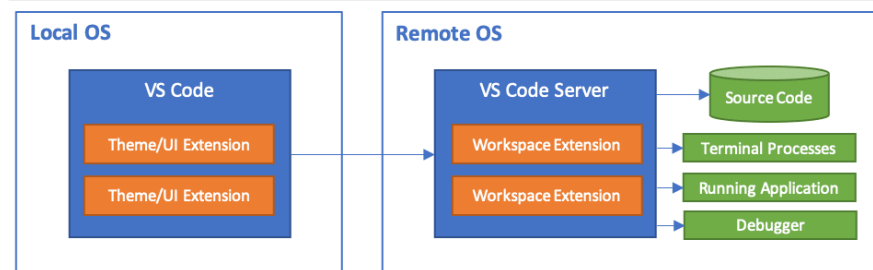
- Un “compilador” de FPGAs
- Un simulador





Visual Studio Code

- Extensiones: C/C++, Python, TCL, Matlab, VHDL, Verilog/SV...
- Live development
- Integración con Git
- Github, Jenkins, Gitlab...
- Buena API: Typescript
- Desarrollo remoto
- Integración con: Docker, Azure, Kubernetes...

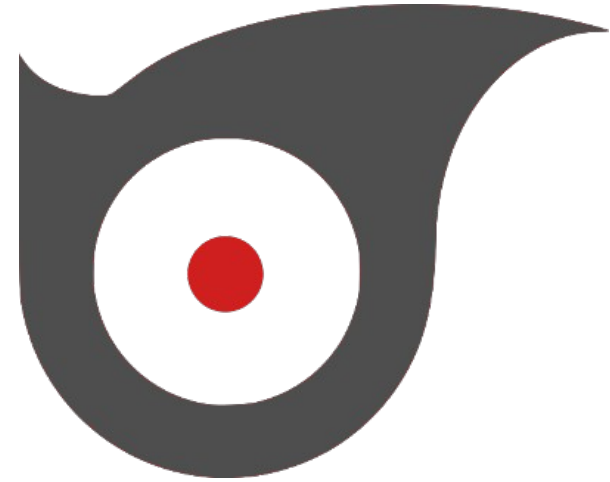


“Software” features for HW designers

- Linter: ModelSim, Vivado, Icarus, GHDL, verilator...
- Formatter
- Snippets
- Intelligent templates

- Tool manager (+15): Quartus, Vivado, GHDL, Icarus, QuestaSim, Xsim...

- State machine viewer
- Automatic documentation



Linux, Windows, Mac



Entorno de desarrollo FPGA/ASIC :

- VHDL, SV/Verilog, TCL editor
- Tool manager

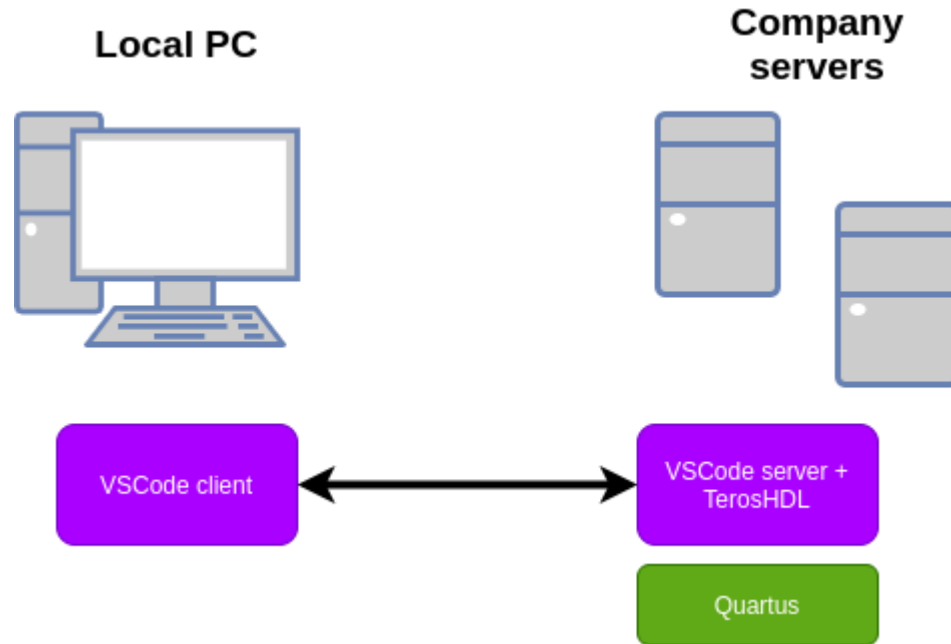
- C++, Python, Matlab editor
- Desarrollo remoto
- Integración con Git



Visual Studio Code

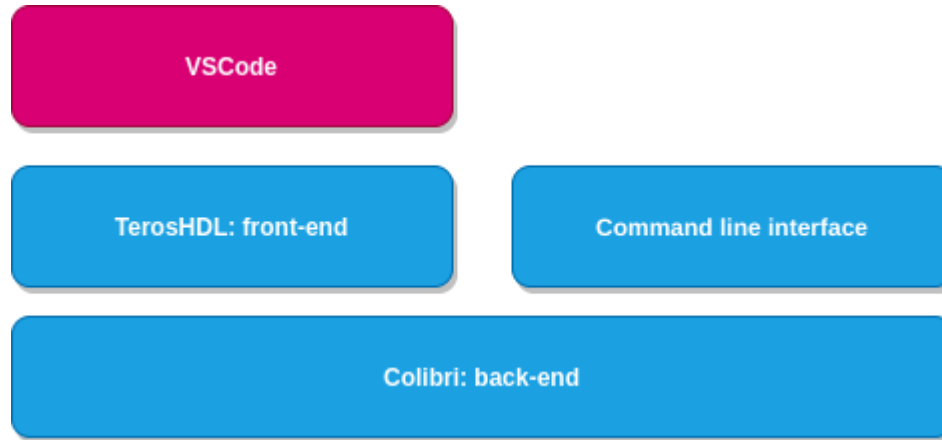
Why TerosHDL?

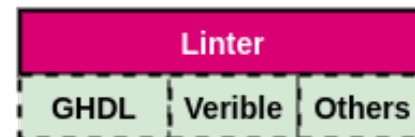
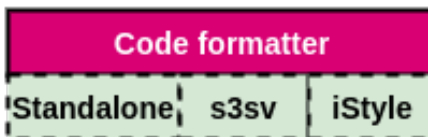
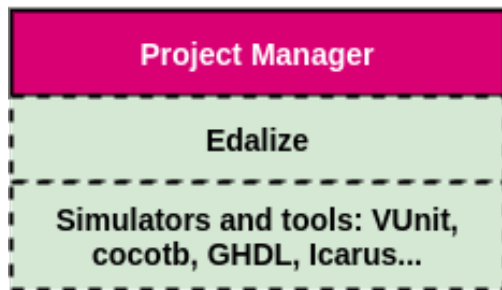
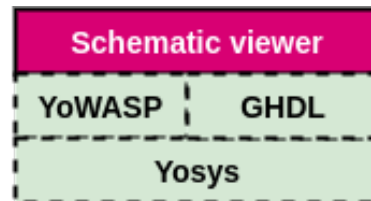
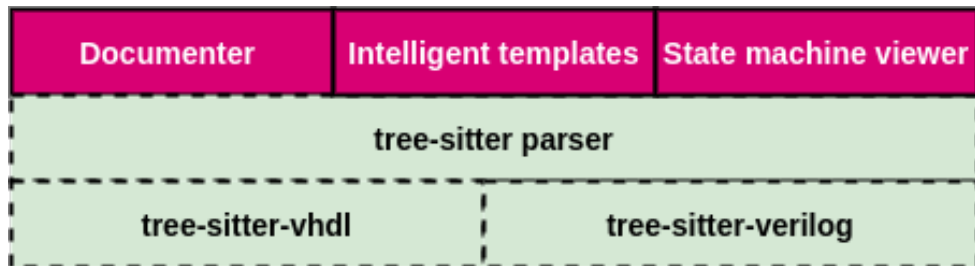
TerosHDL



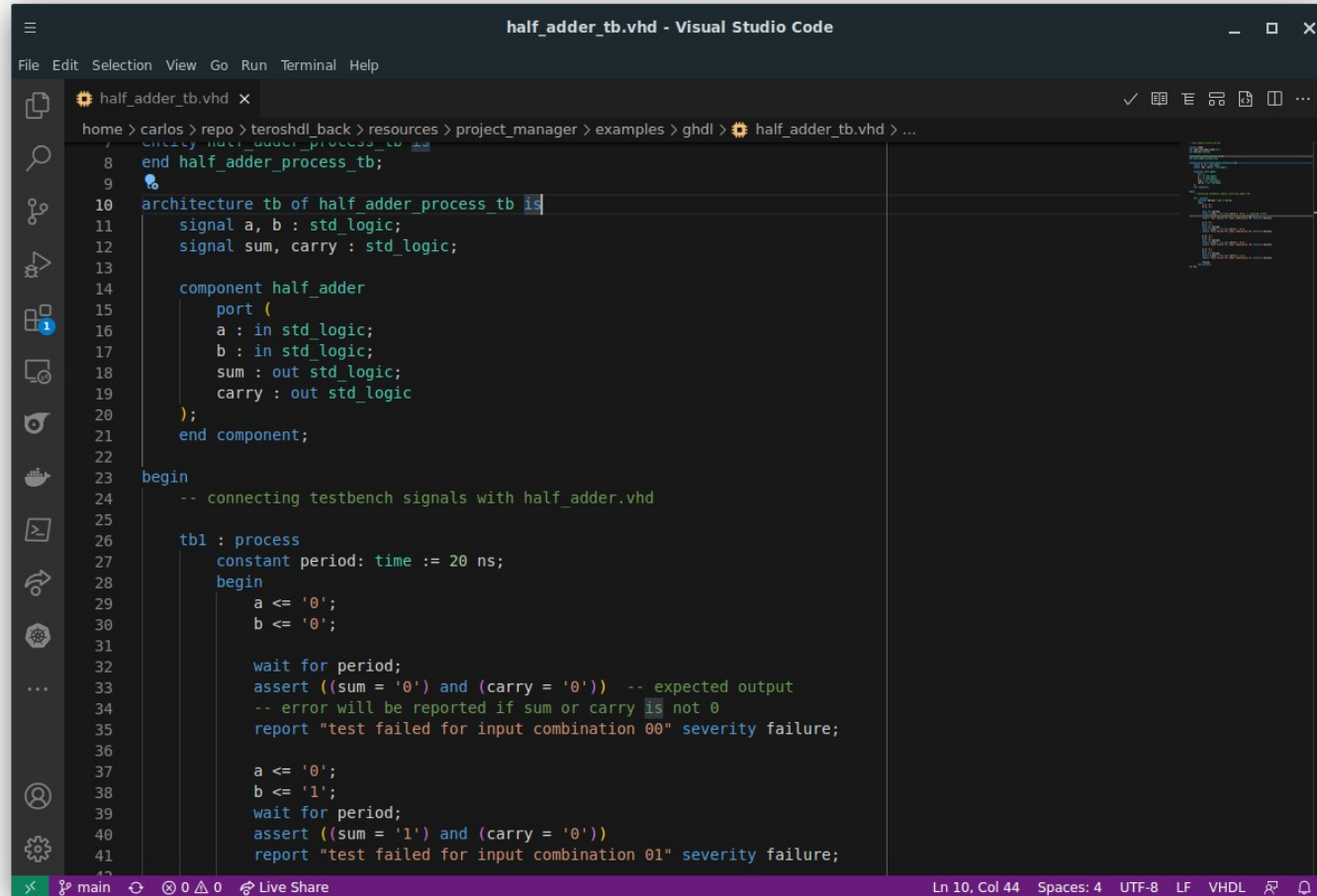
- Easy setup: 40 seconds
- Low latency
- SSH connection







- Resaltado de sintaxis
- Go to definition
- Hover
- Autocompletado
- Renombrado de variables



```
half_adder_tb.vhd - Visual Studio Code
File Edit Selection View Go Run Terminal Help

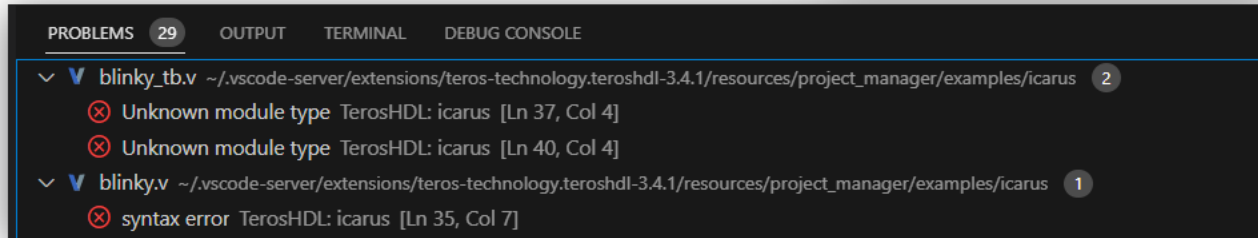
home > carlos > repo > teroshdl_back > resources > project_manager > examples > ghdl > half_adder_tb.vhd > ...

7  entity half_adder_process_tb is
8  end half_adder_process_tb;
9
10 architecture tb of half_adder_process_tb is
11     signal a, b : std_logic;
12     signal sum, carry : std_logic;
13
14     component half_adder
15     port (
16         a : in std_logic;
17         b : in std_logic;
18         sum : out std_logic;
19         carry : out std_logic
20     );
21 end component;
22
23 begin
24     -- connecting testbench signals with half_adder.vhd
25
26     tbl : process
27         constant period: time := 20 ns;
28         begin
29             a <= '0';
30             b <= '0';
31
32             wait for period;
33             assert ((sum = '0') and (carry = '0')) -- expected output
34             -- error will be reported if sum or carry is not 0
35             report "test failed for input combination 00" severity failure;
36
37             a <= '0';
38             b <= '1';
39             wait for period;
40             assert ((sum = '1') and (carry = '0'))
41             report "test failed for input combination 01" severity failure;
42         end process;
43 end;
```



- Detector de errores
- Soporte para tools open Source: GHDL, Icarus...

```
32 always #clk half period clk <= !clk;
33 vlog_tb_utils = 0
34 Unknown module type TerosHDL: icarus
35 View Problem (Alt+F8) No quick fixes available
36 vlog_tb_utils vtu();
37
38
39 blinky #(.clk_freq_hz (clk_freq_hz))
40 dut
41     (.clk (clk),
42      .q (led));
43
44 integer i;
45 time last_edge = 0;
```



- Documentación automática Desde comentarios en el código
- Soporte para Markdown

The screenshot displays the TerosHDL IDE interface. On the left, a VHDL file named `test.vhd` is open, showing code for an entity `cs`. Three sections of the code are highlighted with red boxes: a waveform example comment, a bitfield example comment, and the port declarations. On the right, the 'Module documentation' panel shows the generated documentation for the entity `cs`. It includes a diagram of the entity's ports, a description of the waveform example, a diagram of the bitfield example, and a table of generics and ports.

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 dest opcode 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



