

# Efficient use of a text editor for HDL coding

---

V. Shebalin

December 1, 2022

- **Vim/NeoVim**
- **VSCode** + TerosHDL
- **Emacs** + verilog-mode/vhdl-mode
- **Notepad++**, **Sublime Text**, **Eclipse**, ...

## What we will be using

- NeoVim + numerous plugins
- VSCode + TerosHDL + Python(Microsoft)  
see <https://terostechnology.github.io>

## General

- Fast file navigation
- Fast buffers/files switching
- Fast navigation through the file
- Fast search/replace
- Folding
- Window splitting
- Multi-cursor editing
- Git integration

## Language specific

- Syntax highlighting
- Code formatting
- Snippets/Templates
- Syntax checking
- Pop-up documentation
- Go to definition
- Tags

## HDL specific

- Automatic generation of entity/module instantiations
- Automatic generation of the testbenches

# Fast navigation through file and editing

- Fast search – Vim built-in.
- Sessions – remember opened files, splits, settings, etc (mksession).
- Window splitting.
- Multiple cursor editing <https://github.com/terryma/vim-multiple-cursors>
- Vim easymotion <https://github.com/easymotion/vim-easymotion>

## Files browsing

- Vim native way :Explore, :Sex, :Vex
- **NerdTree**  
<https://github.com/preservim/nerdtree>
- **nnn** <https://github.com/mcchrish/nnn.vim>

## Fuzzy find

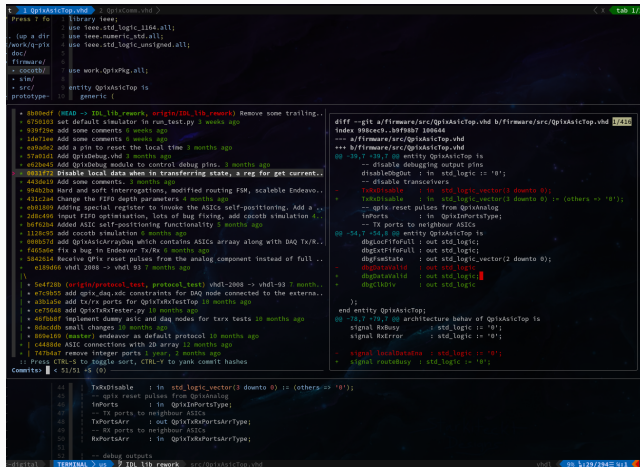
An extremely fast way to open files, navigate buffers, and so on.

- **CtrlP**  
<https://github.com/kien/ctrlp.vim>
- **fzf.vim**  
<https://github.com/junegunn/fzf.vim>
- **Telescope**  
<https://github.com/nvim-telescope/telescope.nvim>

## Other fuzzy find features

All of fzf.vim, and Telescope can do this:

- Fast windows/buffers switching.
- Fast tags navigation.
- Browse syntax check diagnostics.
- Commands history.
- Git branches/commits.



# Syntax checking

## Vim/NeoVim syntax checkers

Provide syntax check, autocompletion, diagnostics, go to definition, show documentation, etc.

- **CoC**  
<https://github.com/neoclide/coc.nvim>
- **ALE**  
<https://github.com/dense-analysis/ale>
- **Syntastic**  
<https://github.com/vim-syntastic/syntastic>
- **NeoVim** builtin LSP

## hdl\_checker

- HDL language server for VHDL/Verilog/SystemVerilog
- Supports Modelsim/Questa, GHDL, Vivado Simulator
- [https://github.com/suoto/hdl\\_checker](https://github.com/suoto/hdl_checker)

## Vim plugins

- Vim-snippets <https://github.com/honza/vim-snippets>
- Ultisnips <https://github.com/SirVer/ultisnips>
- Snipmate <https://github.com/garbas/vim-snipmate>
- Neosnippet <https://github.com/Shougo/neosnippet.vim>

## Templates

- TerosHDL has templates for entity instances, testbenches, and cocotb tests.
- For Vim this plugin can generate entity/components instances: <https://github.com/vim-scripts/VIP> .
- There are a number of testbench generators on Github for instance:  
<https://github.com/phillbush/tbgen>  
[https://github.com/JC-LL/vhdl\\_tb](https://github.com/JC-LL/vhdl_tb)  
<https://github.com/connorcl/testbench-gen>



**Ctags** is a programming tool that generates an index (or tag) file of names found in source and header files of various programming languages to aid code comprehension. (c)Wikipedia

- Exuberant ctags <https://ctags.sourceforge.net>
- Universal ctags <https://github.com/universal-ctags/ctags>

## Vim plugins

- vim-tagbar <https://github.com/universal-ctags/ctags>
- Vista.vim <https://github.com/liuchengxu/vista.vim>

Fast navigation through *tags* with FZF!

- <https://github.com/airblade/vim-gitgutter>  
Marks modified lines of the current buffer.
- <https://github.com/tpope/vim-fugitive>  
*“it’s so awesome, it should be illegal”*  
(c) it’s GitHub page.

Show last changes, diff with a given commit, show logs and more.

## Fugitive

:Git	Gives an <b>interactive</b> (!) version of the git status: can add/reset specific file, open file, undo changes and everything.
:Gwrite	Stage the current file to the index.
:Gread	Revert current file (git checkout).
:Gremove	Delete the current file and vim buffer.
:Gllog	Git log.
:Gvdiffsplit	Open line by line diff in a vertical split.

:GBranches	Open fzf-preview window with a list of branches where one can switch, create, merge, delete branches interactively
:Commits	Show commits with fuzzy find through them and preview of the changes
:BCommits	Show commits log for the current buffer
:GTags	List of git tags
:GFiles	List of tracked files

# Integration with terminal

## Vim

- `:terminal` – open built-in terminal
- `!command` – execute command and put output into the quickfix window
- `:new | r ! command` – execute command and put output into a new buffer
- AsyncRun – <https://github.com/skywind3000/asyncrun.vim>

Run shell comand in the background. E.g. :

`:AsyncRun make` – do make in the background.

`:AsyncRun git push` – git push.

`:AsyncRun --mode=term --pos=tab ./%` – run current buffer and show output in the new tab

`:AsyncRun --mode=term --pos=right ./%` – run current buffer and show output on the right split

## VSCoDe

- VSCode has a built-in terminal window.
- Can run commands asynchronously as well.

See <https://terotechnology.github.io/terosHDLdoc/> for the full list of features with gif previews.

- State machine viewer
- Templates : components/entities/testbenches/cocotb
- Hierarchy view/dependencies graph
- Schematic viewer (synthesis with YoSys)
- Automatic documentation (with wavedrom/bitfield integration)  
Wavedrom (<https://wavedrom.com>) is a powerful tool to draw timing diagrams.  
It has online editor: <https://wavedrom.com/editor.html>.
- Integration with Modelsim, Quartus, Vivado, Radiant, and everything.

## Vim

- <https://www.vim.org/>
- <https://vimawesome.com/>
- Introduction to Vim in Russian <https://www.youtube.com/watch?v=jkxLIFVGfd4>

## VsCode

- <https://www.youtube.com/@terostechnology5520>
- Presentation “TerosHDL: an open source IDE for FPGA developers”  
[https://www.youtube.com/watch?v=\\_wxTjOSO5oY](https://www.youtube.com/watch?v=_wxTjOSO5oY)