

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** – minimalistic fuzzy-find
<https://github.com/ctrlpvim/ctrlp.vim>
- **fzf.vim** – based on fzf program
<https://github.com/junegunn/fzf.vim>
- **Telescope** (NeoVim only) – utilizes most modern NeoVim features, super extensible and powerful.
<https://github.com/nvim-telescope/telescope.nvim>

Other fuzzy find features

zfz.vim and Telescope can do:

- Fast windows/buffers switching.
- Fast navigation between language objects (functions/entities/classes) with help of ctags.
- Browse syntax check diagnostics.
- Search commands history.
- List/search Git branches/commits.

The screenshot shows a Vim editor with two main windows. The left window displays a commit history search results, listing various commits with their hashes, messages, and dates. The right window shows a diff of a file, highlighting changes in the code. The interface includes a terminal at the bottom and a status bar at the very bottom.

```
commit 6b0b0eef (HEAD -> IDL_lib_rework, origin/IDL_lib_rework) Remove some trailing..
commit 675b1093 set default simulator in run_test.py 3 weeks ago
commit 939f23e add some comments 6 weeks ago
commit 1de121e Add some comments 6 weeks ago
commit e9a9ae2 add a pin to reset the local time 3 months ago
commit 57a81d1 Add QpixDebug.vhd 3 months ago
commit e62be45 Add QpixDebug module to control debug pins. 3 months ago
commit 6031f72 Disable local data when in transferring state, a reg for get current..
commit 443e419 Add some comments. 3 months ago
commit 904b2ba Hard and soft interrogations, modified routing FSM, scalable Endeavo..
commit 431c244 Change the FIFO depth parameters 4 months ago
commit eb01809 Adding special register to invoke the ASICs self-positioning. Add a ..
commit 2d8c496 Input FIFO optimisation, lots of bug fixing, add cocotb simulation 4..
commit b0f1c2b Added ASIC self-positioning functionality 5 months ago
commit 1128c95 add cocotb simulation 6 months ago
commit 008b570 add QpixASICArrayDaq which contains ASICs array along with DAQ Tx/R..
commit f465a6e fix a bug in Endeavor Tx/Rx 6 months ago
commit 5042014 Receive Qpix reset pulses from the analog component instead of full ..
commit c199d66 vhdl 2088 -> vhdl 93 7 months ago
commit 5e4f28b (origin/protocol_test, protocol_test) vhdl-2088 -> vhdl-93 7 month..
commit e7c9b55 add qpix_daq.xdc constraints for DAQ node connected to the externa..
commit a2b1a5e add tx/rx ports for QpixRxTestTop 10 months ago
commit ce75a4d add QpixTxTester.py 10 months ago
commit 40f8baf implement dummy asic and daq nodes for tarx tests 10 months ago
commit 8d0cddb small changes 10 months ago
commit 806e109 (master) endeavor as default protocol 10 months ago
commit c448bde ASIC connections with 2d array 12 months ago
commit 1a7b0a7 remove integer ports 1 year, 2 months ago
Press CTRL-~ to toggle sort, CTRL-V to yank commit hashes

Commits: 61/51 +5 (0)

diff --git a/firmware/src/QpixASICTop.vhd b/firmware/src/QpixASICTop.vhd 1/43c
index 08cece9..80f98b7 100644
--- a/firmware/src/QpixASICTop.vhd
+++ b/firmware/src/QpixASICTop.vhd
@@ -39,7 +39,7 @@ entity QpixASICTop is
    -- disable debugging output pins
    disableDebugut : in std_logic := '0';
    -- disable transceivers
    TxRxDisable : in std_logic_vector(3 downto 0);
    TxRxDisable : in std_logic_vector(3 downto 0) := (others => '0');
    -- qpix reset pulses from QpixAnalog
    inPorts : in QpixInPortType;
    -- TX ports to neighbour ASICs
    dbgExtFifoFull : out std_logic;
    dbgExtFifoFull : out std_logic;
    dbgExtFifoFull : out std_logic_vector(2 downto 0);
    dbgDataValid : out std_logic;
    dbgDataValid : out std_logic;
    dbgClkDiv : out std_logic;

end entity QpixASICTop;
@@ -79,7 +79,7 @@ architecture behav of QpixASICTop is
    signal RxBusy : std_logic := '0';
    signal RxError : std_logic := '0';

    signal localDataIn : std_logic := '0';
    signal routeBusy : std_logic := '0';
```

Syntax checking

Vim/NeoVim syntax checkers

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

- **CoC** – a little slow, great autocompletion, not easy to configure.
<https://github.com/neoclide/coc.nvim>
- **ALE** – minimum configuration needed, fast linting, not good autocompletion.
<https://github.com/dense-analysis/ale>
- **Syntastic** – no longer maintained
<https://github.com/vim-syntastic/syntastic>
- **NeoVim** builtin LSP – fast, nice looking error notifications, difficult to configure.

hdl_checker

- HDL language server for VHDL/Verilog/SystemVerilog
- Supports Modelsim/Questa, GHDL, Vivado Simulator
- 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/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>

:BTags – fast navigation through *tags* with fzf.vim!

- <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 interactive (!) 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

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