写在前面：

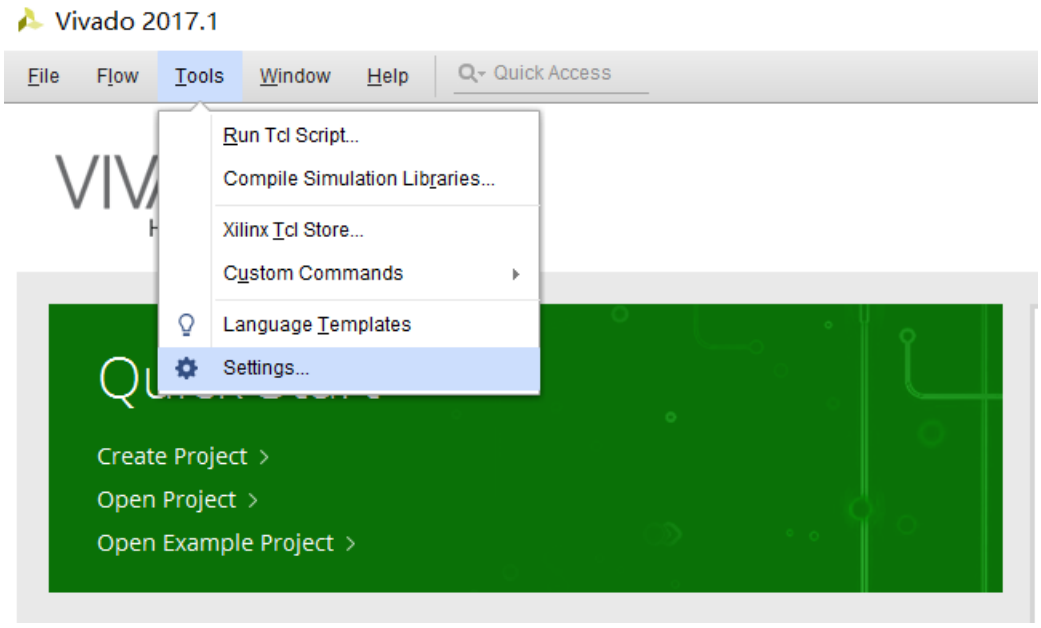
第一届系统能力培养大赛结束，进入决赛的各个队伍，基本上都做到了五级流水+神经网络/深度学习加速的异构 SOC，这些名词也许你们不知道，或者听起来似乎很有难度，但是有了第一届的参赛经验，并且学习到了包括清华、南大、西北工业、北航等学校的优秀经验，所以做一个非官方的系列教程，打包为系统能力培养的基础教程。

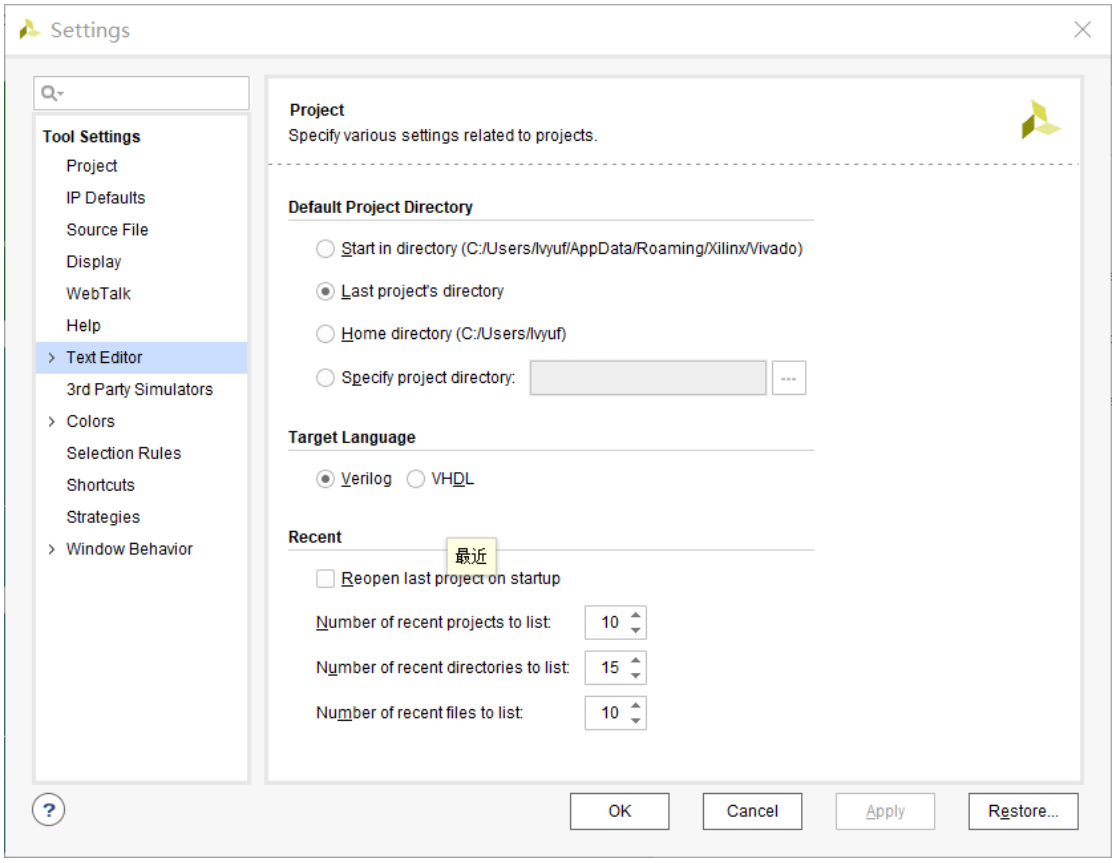
这个教程的目的有二：其一，降低准入门槛，相信上学期你们被包括 Vivado 环境、IDE 使用等令人抓狂的问题而对 FPGA 开发充满反感，这是包括老师、助教等，在多方面经验缺失所导致的问题，所以我尽可能把一些 tips 给你们做成完整教程，少走弯路，提高效率。其二，提高难度，当然这里并不是说要把你们难倒，而是让你们充分体会包括流水线、超标量等等组成原理和体系结构所能够培养的能力，结合你们的实际能力，并没有什么实现上的困难，更多的仍旧是缺少引导；并且，对于你们听过或者没听过的 FPGA 加速，或者国内一些领先企业完成的人工智能芯片，在你们手里完全具备可实现性。

最后有一个长期的目标，就是将组成原理、操作系统、编译原理以及计算机网络四门课程合体，让你们有一个通透的硬件底层逐步抽象直到软件应用层的理解，这是计算机科学与技术与软件工程等其他专业的区别所在，也是真正的专业能力。想象一下你们自己的操作系统跑在自己的 CPU 上，做各种外设的应用，并且能够随心所欲地增改指令来完成更多的功能，这比起做一个 APP，一个创业比赛，实在是高出不知道多少。

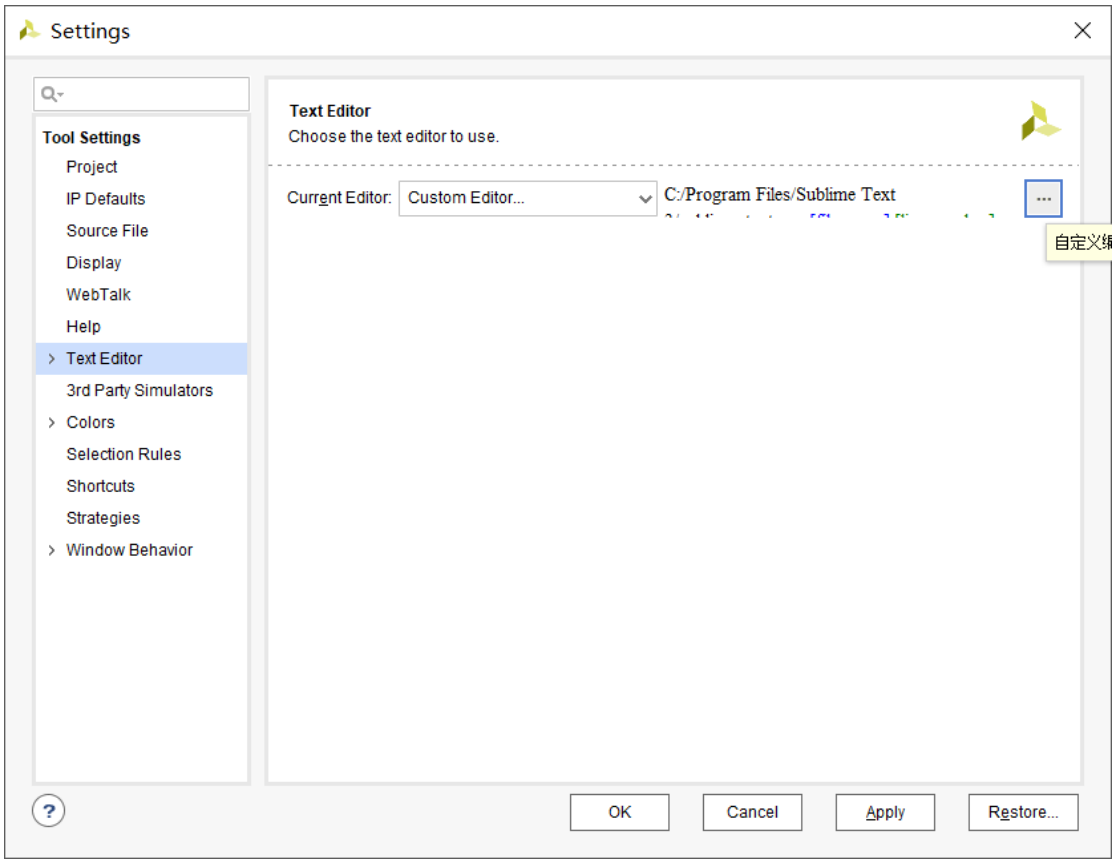
废话不多说，首先要解决的是，vivado 本身十分不友好的 IDE（连语法错误都未必检测的到），以及 Verilog 的自动补全和代码提示。

Step 1. Select *Tools->Settings.. ->Text Editor*





Step 2. Choose *Costom Editor* , then Select the *...* and set the right path of your Editor.



Step 3. Use ***Sublime*** as Example.(By the way, you can use any other Editors as you like)

Input the path as follow:

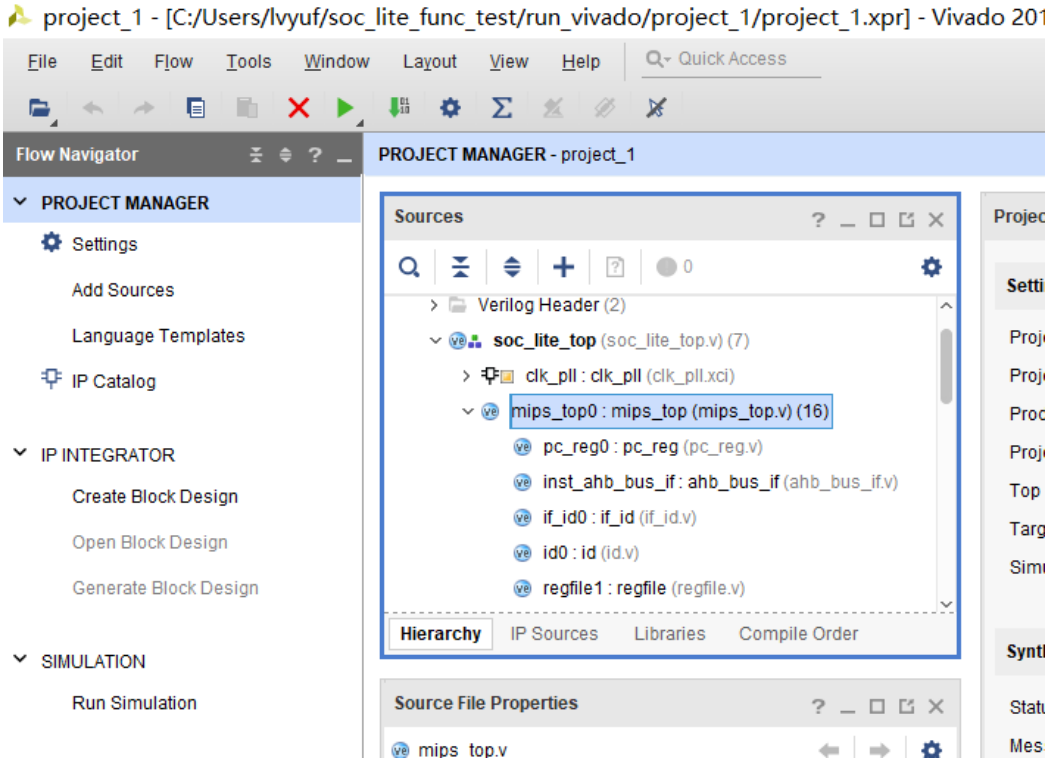
C:/Program Files/Sublime Text 3/sublime_text.exe [file name]:[line number]

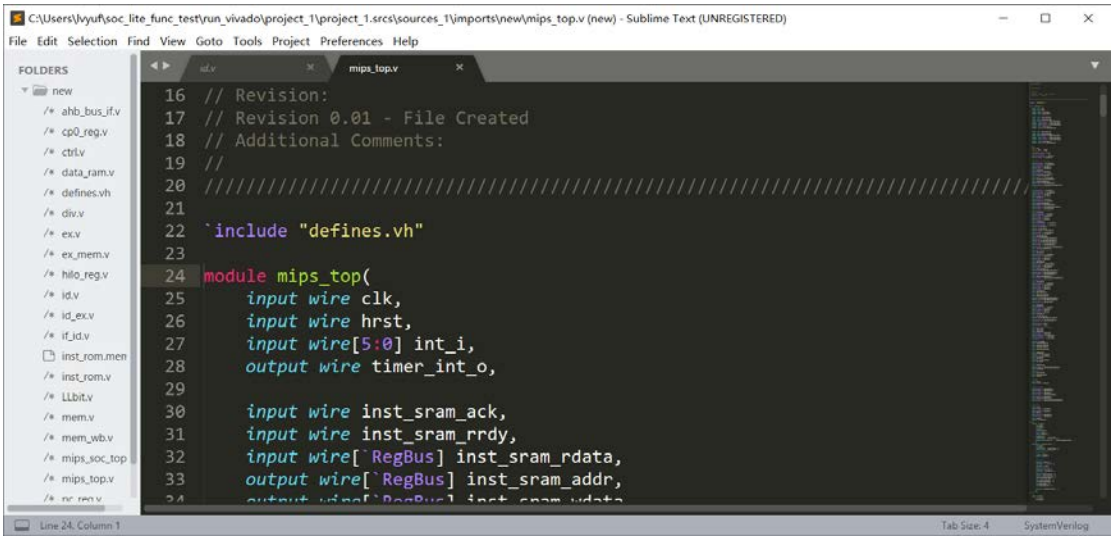
(The **red** path is the full path where you installed the Sublime, and the **black** words just keep it)

Step 4. Choose **OK** and then **Apply** it.

Then you can choose one file in **Source** window in your project to test it.

Double click it, and you can see Sublime open it.

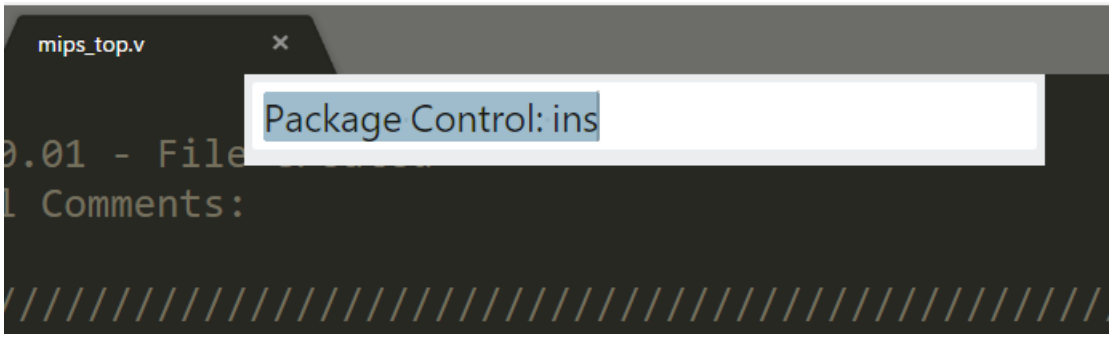




For now, you can use other Editor to program Verilog. But you still need the **Verilog language Plug.**

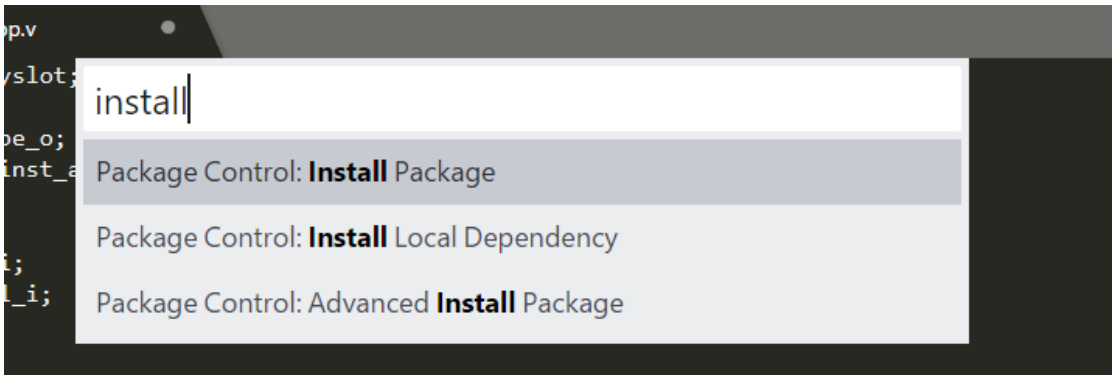
First of all, make sure you have installed the ***Package Control*** like this. (Press **Ctrl+Shift+P**). If you have not installed it, please see this page:

https://www.jeffdesign.net/blog/62/

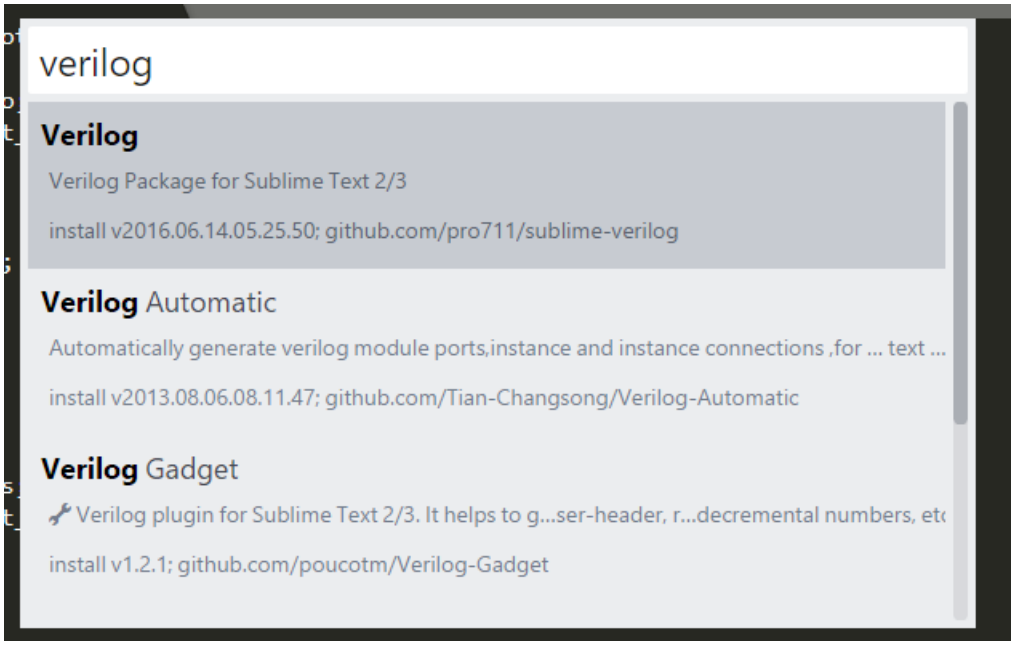


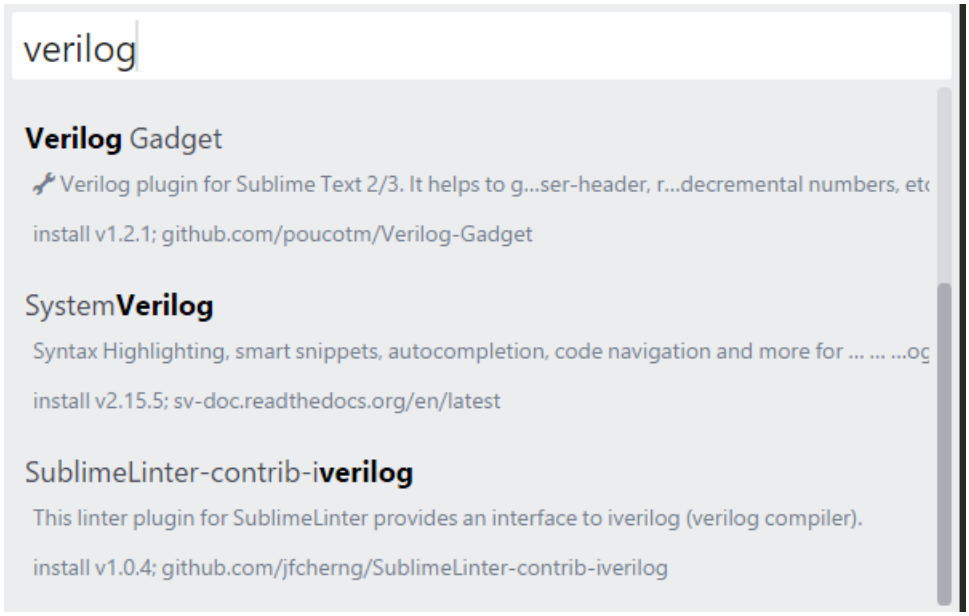
Now we start the remain part.

Step 5. Press **Ctrl+Shift+P** and input **Install,** then Enter.



Step 6. Input **Verilog.** You can see a few selections for you. There are only two type you can use, which is **Verilog** and **SystemVerilog.** If you have enough time, you can make a little comparable operations by yourself, now I just tell you the best one —— **SystemVerilog**.

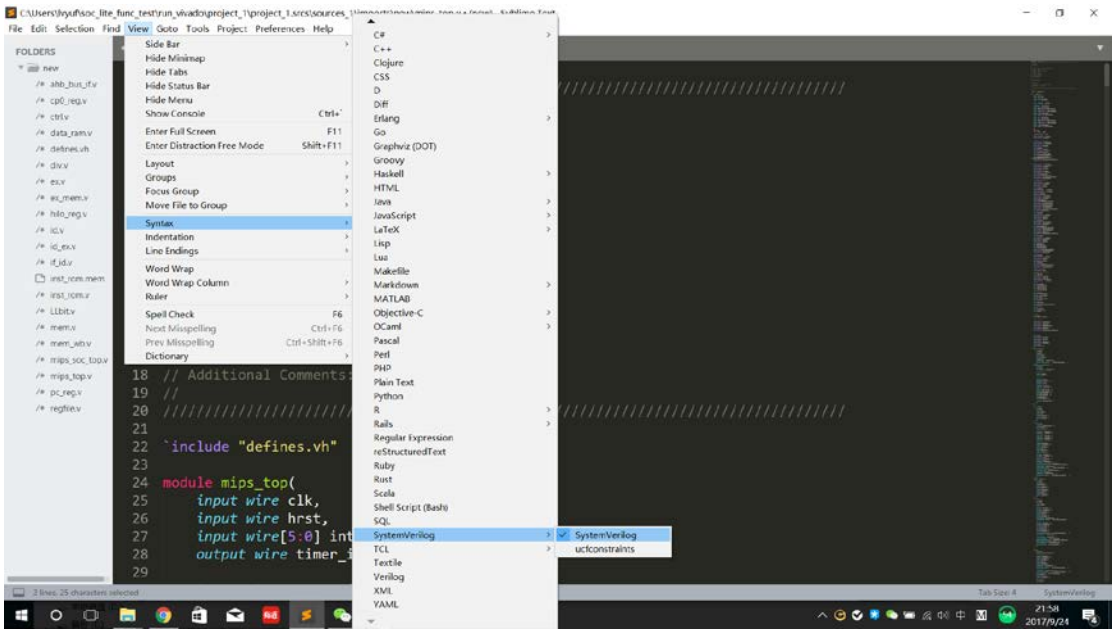


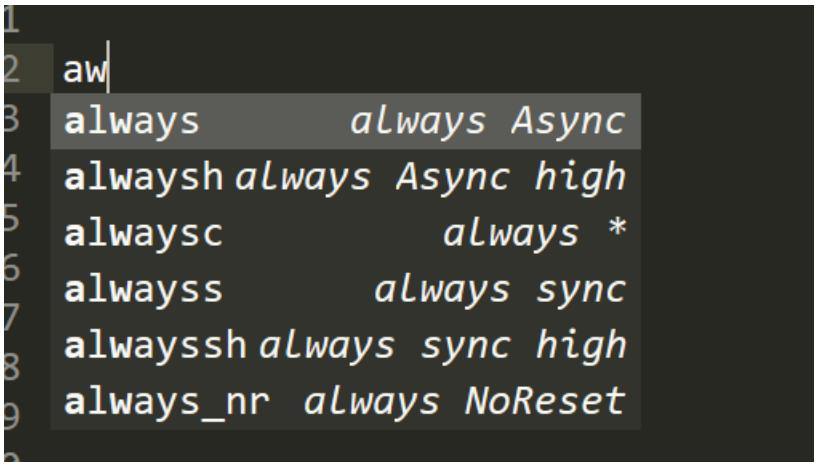


Step 7. Select *View->Syntax.. ->SystemVerilog->SystemVerilog*

And congratulations, now you can enjoy the faster Editor to program Verilog.

A small tip, type in always, see what gonna happen.


File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
new
Side Bar
Hide Minimap
Hide Tabs
Hide Status Bar
Hide Menu
Show Console Ctrl+`
Enter Full Screen F11
Enter Distraction Free Mode Shift+F11
Layout
Groups
Focus Group
Move File to Group
Syntax
Indentation
Line Endings
Word Wrap
Word Wrap Column
Ruler
Spell Check F6
Next Misspelling Ctrl+F6
Prev Misspelling Ctrl+Shift+F6
Dictionary
C#
C++
Clojure
CSS
D
Diff
Erlang
Go
Graphviz (DOT)
Groovy
Haskell
HTML
Java
JavaScript
LaTeX
Lisp
Lua
Makefile
Markdown
MATLAB
Objective-C
OCaml
Pascal
Perl
PHP
Plain Text
Python
R
Rails
Regular Expression
reStructuredText
Ruby
Rust
Scala
Shell Script (Bash)
SQL
SystemVerilog
TCL
Textile
Verilog
XML
YAML
SystemVerilog
ucfconstraints
// Additional Comments:
//
`include "defines.vh"
module mips_top(
input wire clk,
input wire hrst,
input wire[5:0] int
output wire timer_i
2017/9/24


aw
always always Async
alwaysh always Async high
alwaysc always *
alwayss always sync
alwayssh always sync high
always_nr always NoReset