**实验5**

**Verilog代码编写**

# **实验介绍**

作为一门编程语言，我们自然关心应当如何舒适地进行代码编写工作。在本学期的实验中，我们推荐大家使用 Visual Studio Code（VSCode）作为代码编辑器进行编程工作。

# 实验内容

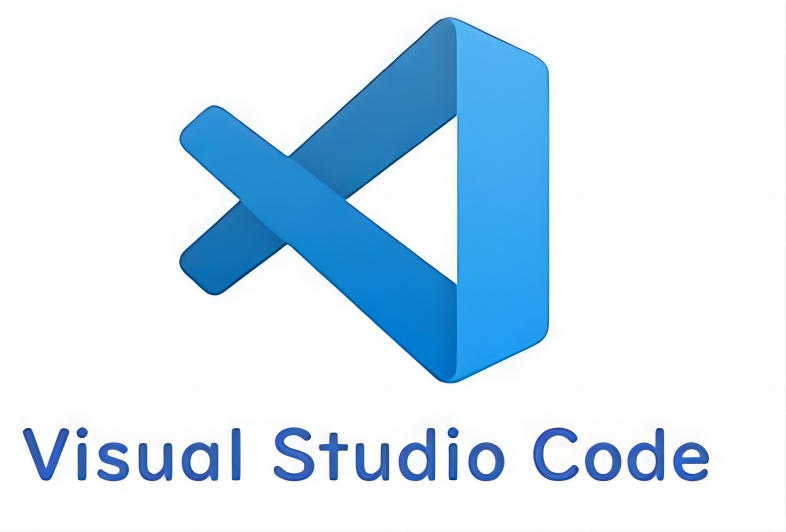
## 1 VSCode

### 1.1 VSCode概述

VSCode 是微软公司开发的一款代码编辑器，有着“宇宙最强编辑器”的美称。它的主要特点为：

* 开源免费
* 轻量级
* 跨平台

除此之外，它对各种编程语言有着强大的支持，包括主流的 C/C++、Java、Python、Go 等等，而丰富的扩展以及社区生态也为我们的代码编写工作提供了极大的便利。

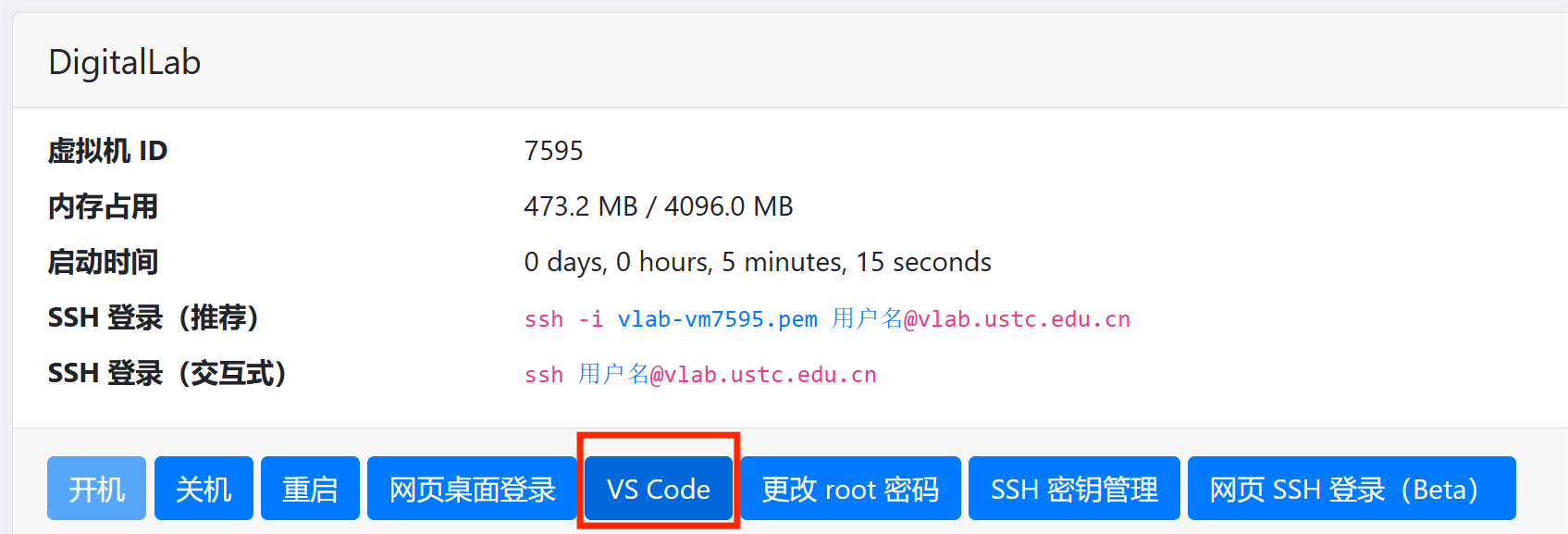
[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/vscode_logo.png)

你可以访问其[官方网站](https://code.visualstudio.com/)（https://code.visualstudio.com/）获取下载方式和更多信息。

### 1.2 VSCode安装

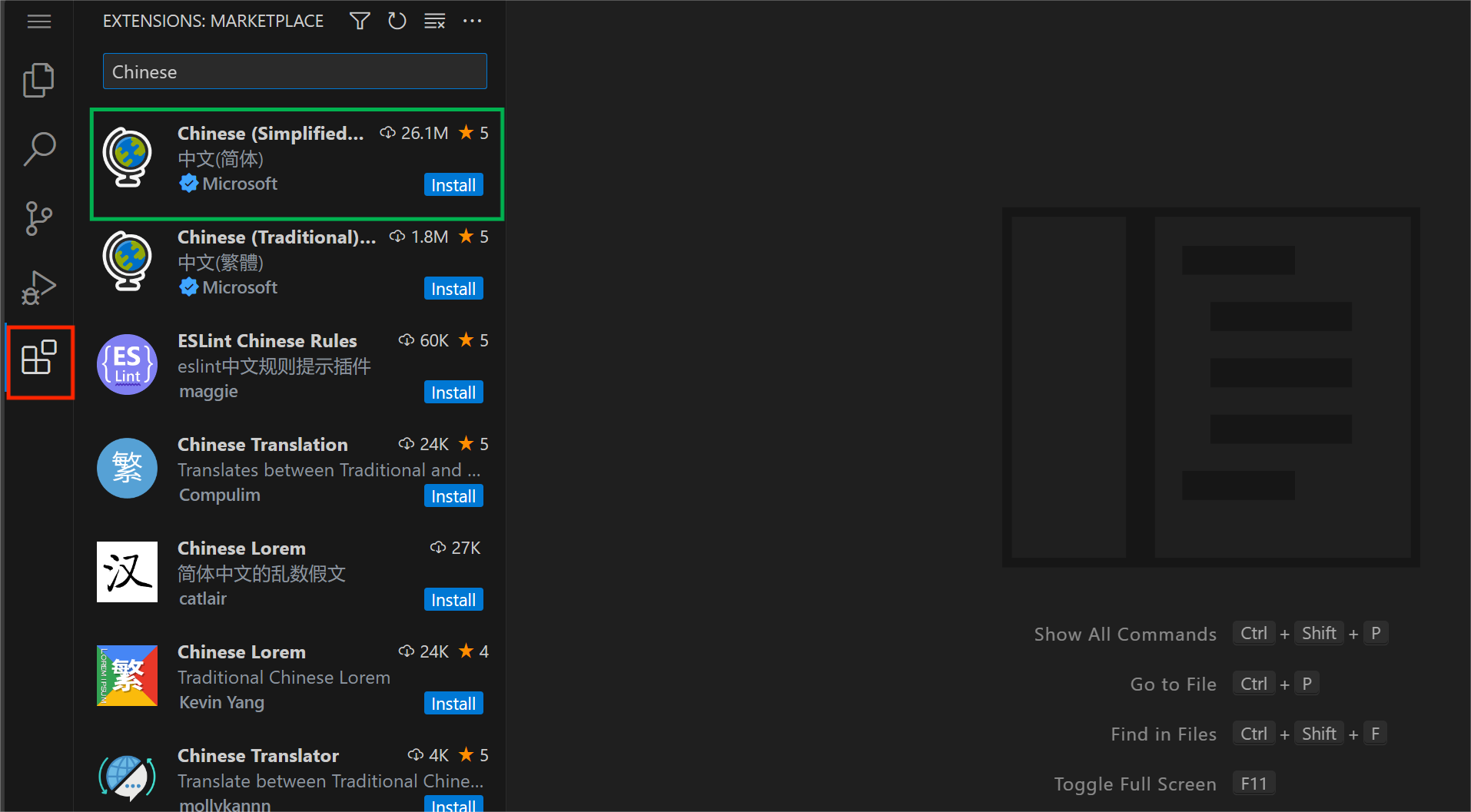
关于如何在本地下载并安装 VSCode 请大家自行查阅有关教程。在下面的教程中，我们默认大家已经拥有了可以运行的 VSCode。

如果你是 VLAB虚拟机用户，则可以使用环境自带的 VSCode 网页登录。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/vlab_vscode.png)

为此，你需要进行初步的配置工作。在VLAB中打开终端，输入 vscode start 即可，同时这会将 VSCode 服务设为开机自动启动，使得你下次可以直接使用 VSCode 而无需重新操作命令行。若你想要停止 VSCode 服务，请使用 vscode stop，同时这会取消 VSCode 服务的开机自动启动。

如果你的 VSCode 是英文界面，可以在左侧扩展（Extensions）栏中输入 "Chinese"，安装如图所示的中文语言扩展。你也可以在搜索栏中输入扩展的 id 进行精确搜索。对应的中文语言扩展的 id 为 ms-ceintl.vscode-language-pack-zh-hans。

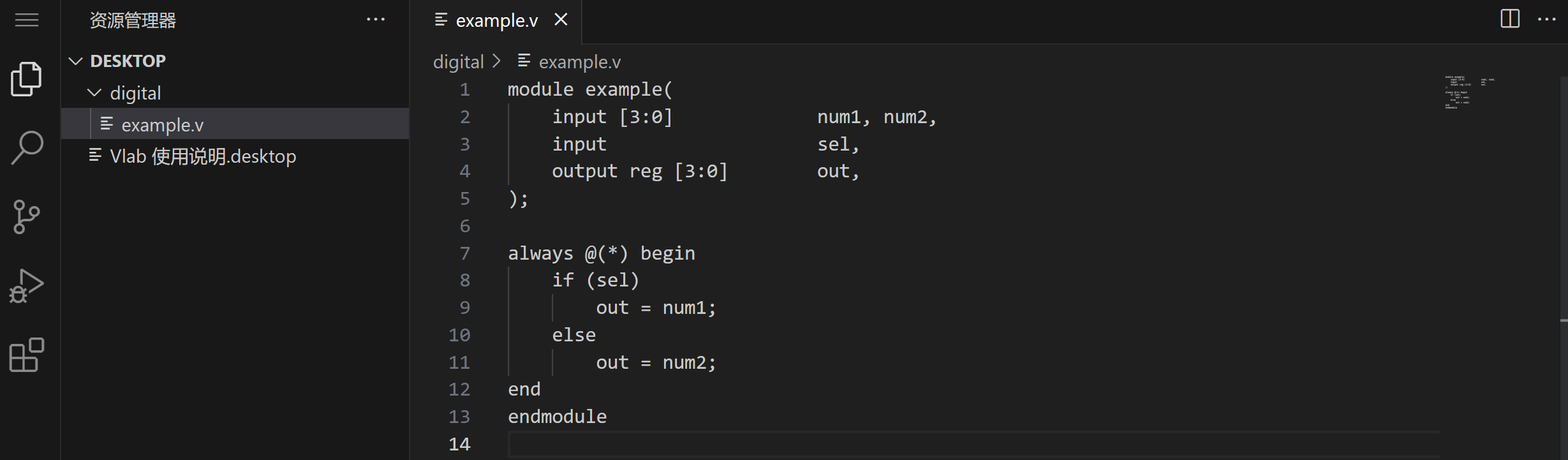


## 2 VSCode开发环境

### 2.1 第一个Verilog程序

首先，在磁盘空间充足的位置新建文件夹，将其重命名为 Digital 或其他你喜欢的名字，这就是我们数电实验的项目文件夹。在项目文件夹中，创建一个新的文本文档，并将其重命名为 example.v。随后在里面输入下面的代码：



[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/example.png)

这是一个简单的 4bits 二选一选择器的模块代码。注意到在模块输出端口声明部分，变量 out 后面多了一个逗号。接下来我们将一步步完成对 VSCode 的功能配置。

### 2.2语法高亮

首先，我们需要让自己的 Verilog 语言对应的 .v 文件与常见的 .txt 文件有着最基本的区别。因此我们希望能为自己的 Verilog 代码也引入语法高亮。你可以在 VSCode 应用商店（Ctrl + Shift + X）中输入 verilog，找到下图所示的插件 Verilog-HDL/SystemVerilog/Bluespec SystemVerilog（下文简称 Verilog-HDL）。Verilog-HDL 扩展的 id 为 mshr-h.veriloghdl。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/verilog_vscode_plug_info.png)

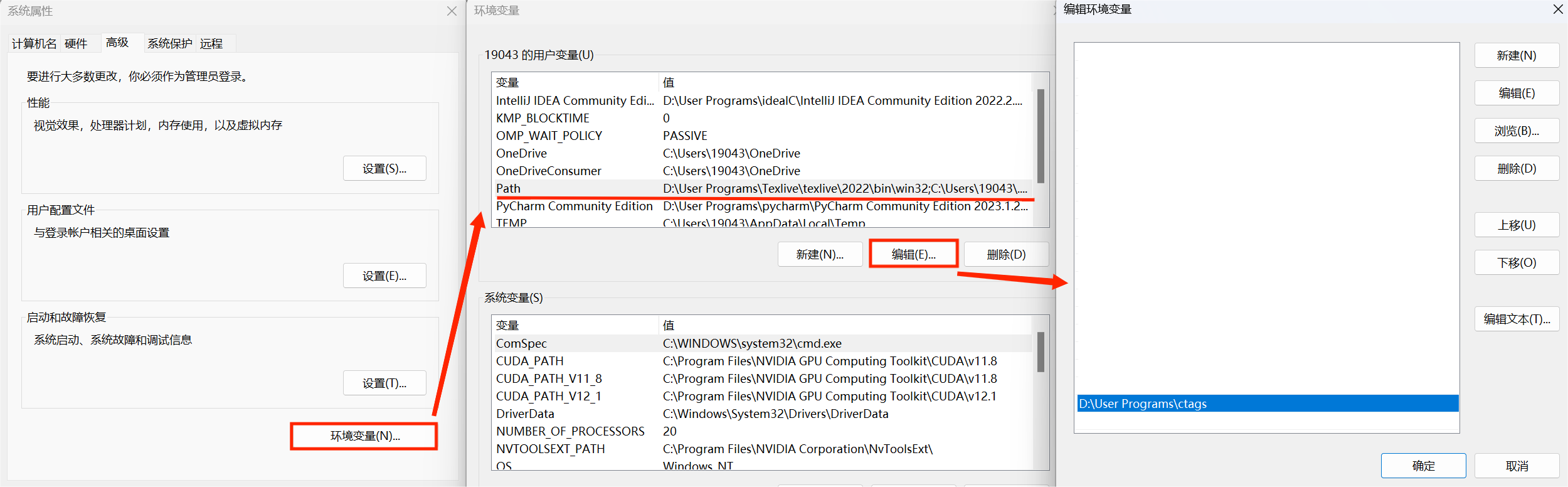
单击安装即可自动安装此扩展。重新启动 VSCode 后，我们的 Verilog 程序中就自动出现了语法高亮。

|  |  |
| --- | --- |
| [IMG_257](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/no_highlight.png)没有语法高亮 | [IMG_258](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/highlight.png)有语法高亮 |

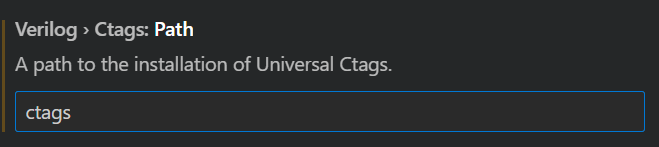
### 2.3变量识别

接下来，我们希望能在程序中识别定义的变量，通过右键的方式定位到变量定义的位置。为此我们需要下载并配置 Ctags。

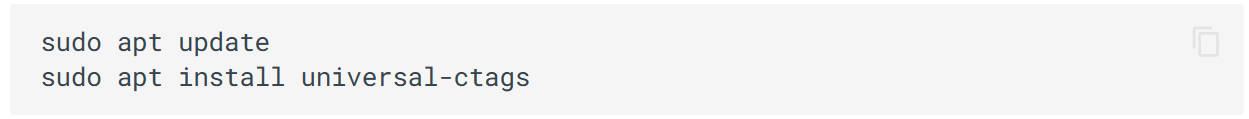
Windows 用户可以在[这里](https://github.com/universal-ctags/ctags-win32/releases/tag/v6.0.0)下载（https://github.com/universal-ctags/ctags-win32/releases/tag/v6.0.0）。请下载最新的 ctags-v6.0.0-x64 版本，将其解压到一个不含特殊字符的目录下。然后，将包含 ctags.exe 文件的最底层路径添加到 Windows 系统环境变量中的 Path 中。例如，编者电脑上的路径为 D:\User Programs\ctags，其中 ctags 目录下包含 ctags.exe 文件。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/ctags_path_2.png)

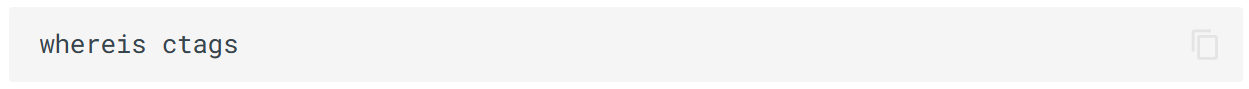
下载并配置完成后，打开 Verilog-HDL 插件的设置界面（在扩展界面单击右下角的齿轮图标），如下图所示将 Ctags:Path 设置为默认的 ctags。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/ctags_path.png)

Vlab 以及 Linux 用户可以在终端中输入下面的命令自动安装 Ctags：

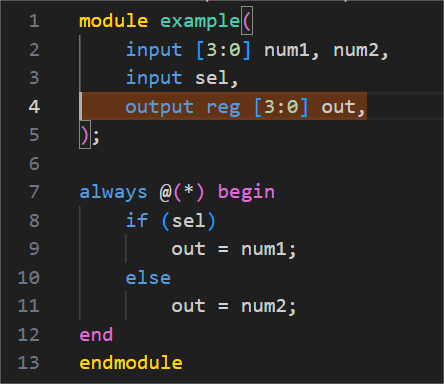


安装完成后，输入



查看 Ctags 的安装目录，一般来说位于 /usr/bin/ctags 目录下。同样地，打开 Verilog-HDL 插件的设置界面，将 Ctags:Path 设置为 Ctags 的安装目录。

设置完成后退出 VSCode，重新打开代码界面。此时选中变量 out，对其右键后选择「转到定义」即可自动跳转到 out 变量定义的位置。

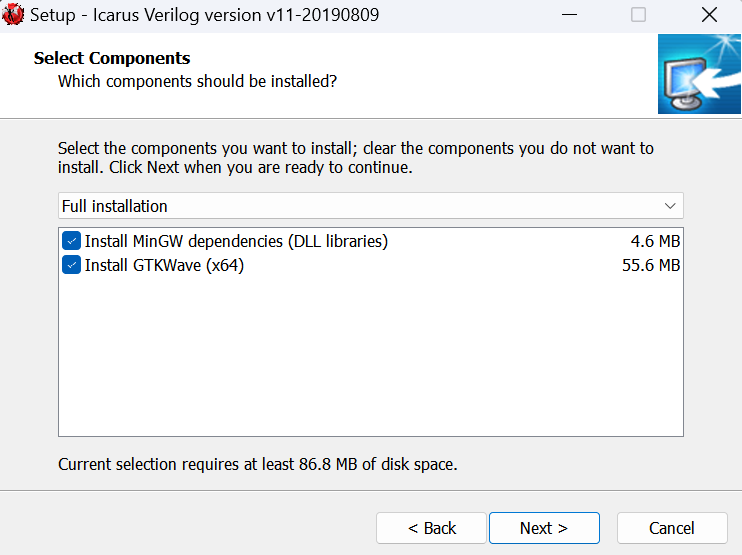
[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/ctags_use.png)

### 2.4语法检查

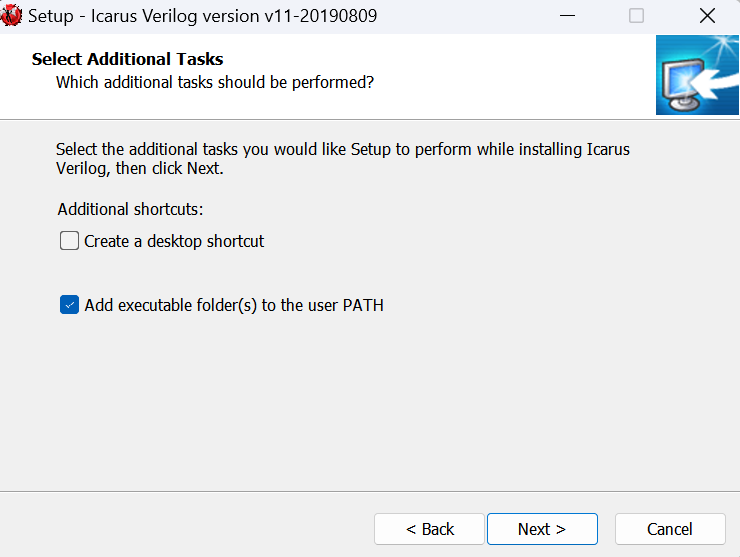
Linter 是一种静态代码分析工具，可以在不运行代码的形况下（静态检查）检查代码中的语法错误，以及一些不规范的写法。这个术语最早来源于一个名叫 lint 的 UNIX 工具，最初用于检测 C 语言的源码。后来各个主流开发语言或大型项目均打造了自己的检查工具，linter 的术语也逐渐流传开来。

在后续的实验中，我们将选择 iverilog 与 gtkwave 作为我们进行语法检查与仿真的工具。其中，iverilog（Icarus Verilog）是一个轻量、免费、开源的 Verilog 编译器，将 Verilog 编写的源代码编译为某种目标格式。而其自带的 GTKWave 可以用于查看标准的 Verilog VCD/EVCD, 以及其他的一些格式的波形文件。

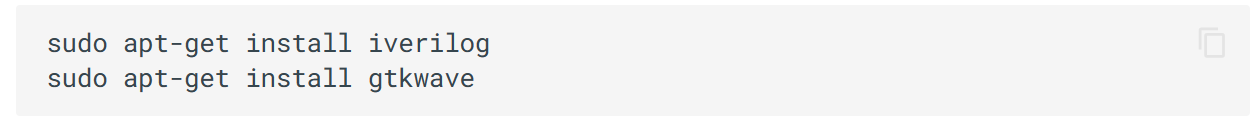
Windows 用户可以在 [iverilog 官方网站](http://bleyer.org/icarus/)（http://bleyer.org/icarus/）选择自己要下载的版本。双击运行下载好的 .exe 程序后，选择同时下载 gtkwave。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/iverilog_download_1.png)

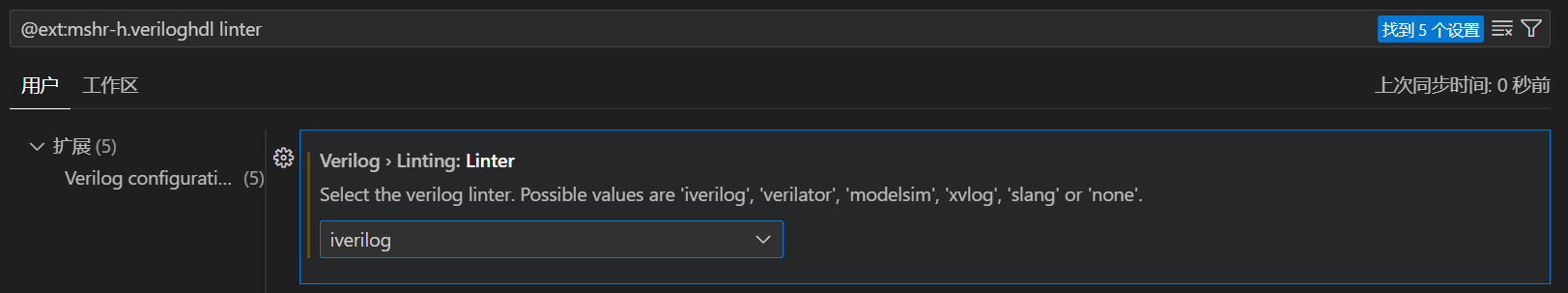
接下来一路 next 即可。注意：iverilog 的安装目录不能有中文和空格。如果想省事可以在安装时勾选添加环境变量（Add folder to PATH）。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/iverilog_download_2.png)

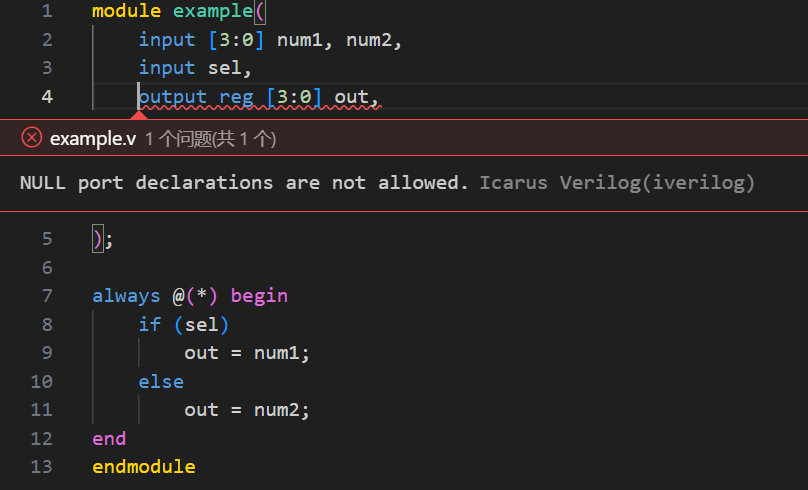
Vlab 以及 Linux 用户可以通过下面的命令安装：



安装完成后，打开 Verilog-HDL 插件的设置界面，如下图所示，将 Verilog > Linting:Linter 设定为 iverilog。

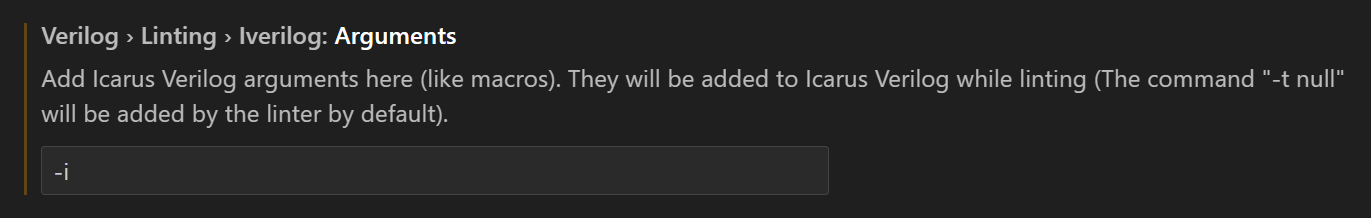
[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/iverilog_path.png)

保存后重新进入 VSCode，打开先前的 example.v 程序，可以发现语法错误已经被正确检测。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/iverilog_use.png)

需要注意的是，你可能会遇到如下的报错信息：Unknown module type这是因为「编译」是以单个文件为单位编译的，而多个文件编译需要「链接」。Verilog-HDL 插件无法智能推断出多个文件之间的依赖关系。为此，我们可以使用命令参数抑制此类错误，即 iverilog 编译器不再检查模块实例化的时候引用的模块是否已经定义。

打开 Verilog-HDL 插件的设置界面，在 Verilog > Linting > Iverilog:Arguments 一栏添加 -i 命令参数。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/iverilog_arg.png)

保存并重新进入 VSCode。此时 iverilog 便不再检查模块引用错误。至此，我们已经得到了基本的 VSCode 开发环境。

## 3 进一步配置

### 3.1更直观：TerosHDL

在上一步的基础上，如果想追求更为极致的体验，可以选择使用 TerosHDL。

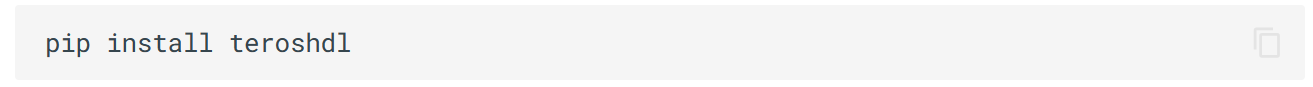
TerosHDL 的使用需要自行配置 Python 环境。在扩展界面搜索 TerosHDL，选择下图所示的插件并安装。该扩展的 id 为 teros-technology.teroshdl。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/teroshdl_search.png)

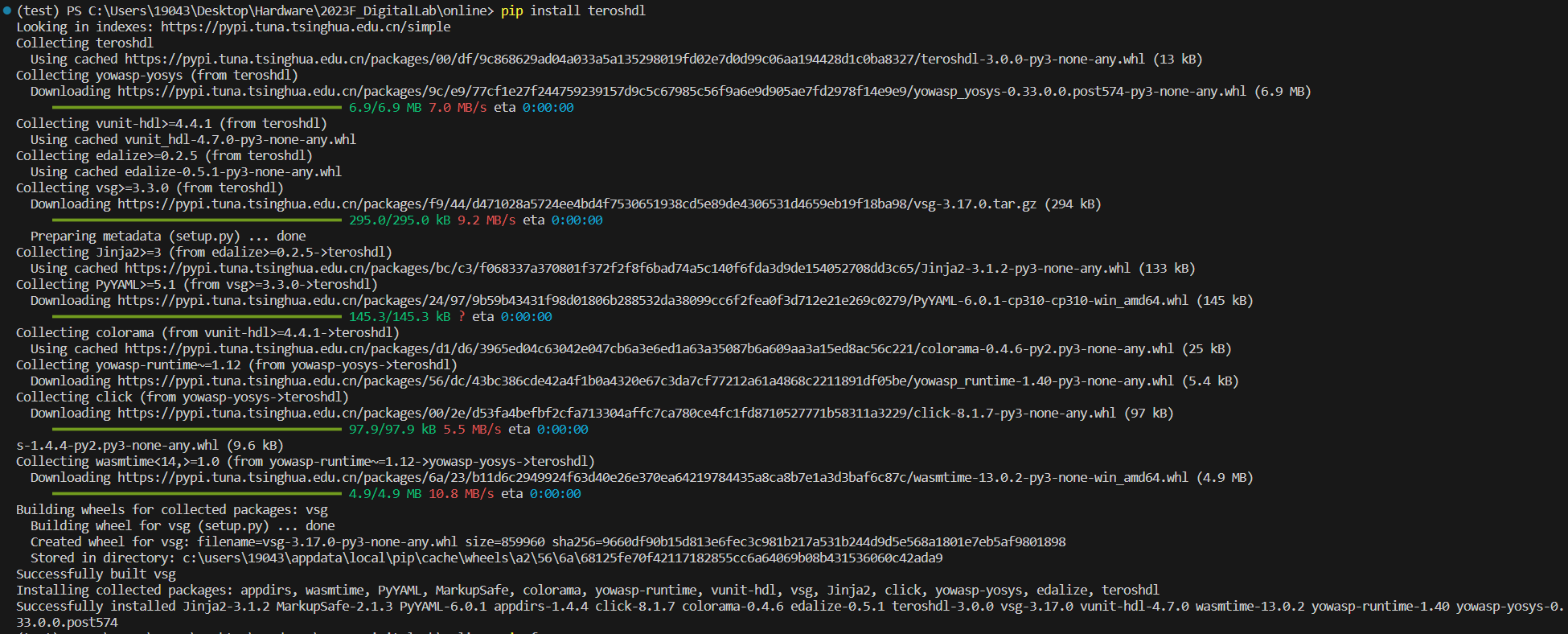
安装完成后，你可以在左侧活动栏找到 TerosHDL 的图标

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/teroshdl_logo.png)

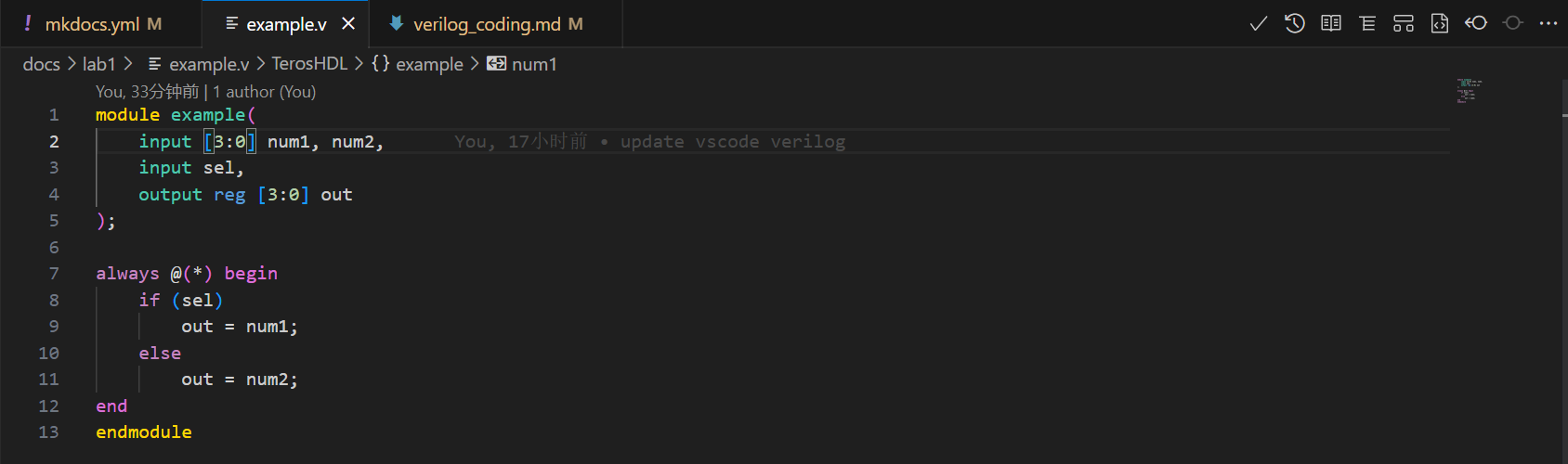
单击图标。新建一个终端，并输入



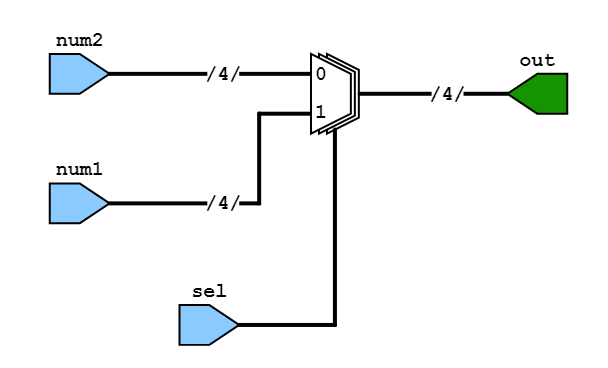
等待相关包下载完成即可。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/teroshdl_download.png)

完成后点击 example.v 文件，此时右上角便出现了若干功能按钮。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/teroshdl_coding.png)

例如：Schematic Viewer 可以查看电路的预览图。如下图所示，可以很直观地看到二选一选择器的构造。

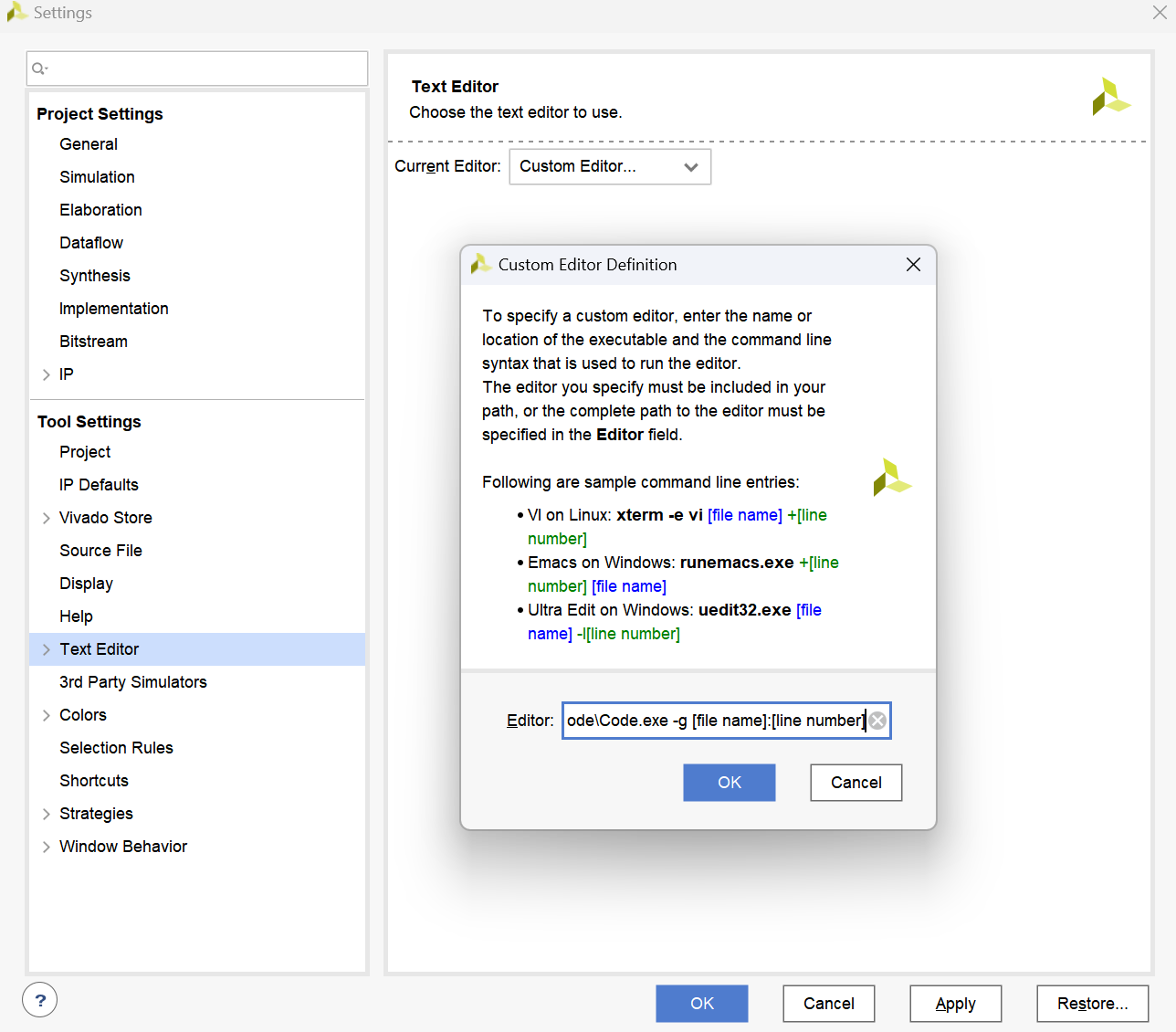
[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/teroshdl_example.png)

如果遇到了其他配置或使用上的问题，你也可以访问[官方文档](https://terostechnology.github.io/terosHDLdoc/docs/getting_started/installation)（https://terostechnology.github.io/terosHDLdoc/docs/getting\_started/installation）获取帮助。

### 3.2将VSCode设定为Vivado的默认编辑器

我们已经配置好了完善的 VSCode 开发环境，这个环境比 Vivado 自带的编辑器环境不知道高到哪里去了。我们希望能够将 Vivado 的默认编辑器更改为 VSCode，为此你可以按照如下的步骤进行操作（仅限Windows 用户）。

**Step1：**进入 Vivado，在工具栏中依次点击 Tools→Settings→Text Editor，以打开文本编辑器设定界面。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/vivado_vsc.png)

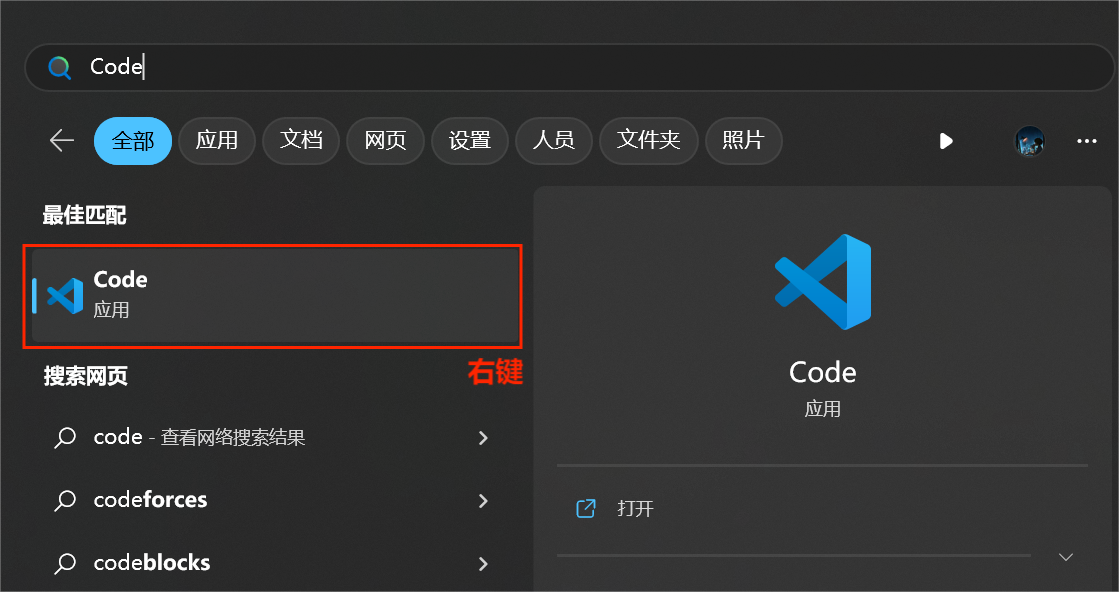
**Step2：**将 Current Editor 栏的内容更改为 Custom Editor，并在弹出的界面中正确设定 VSCode 的路径。其格式为

*到 Microsoft VSCode 的绝对路径/Microsoft VS Code/Code.exe -g [file name]:[line number]*

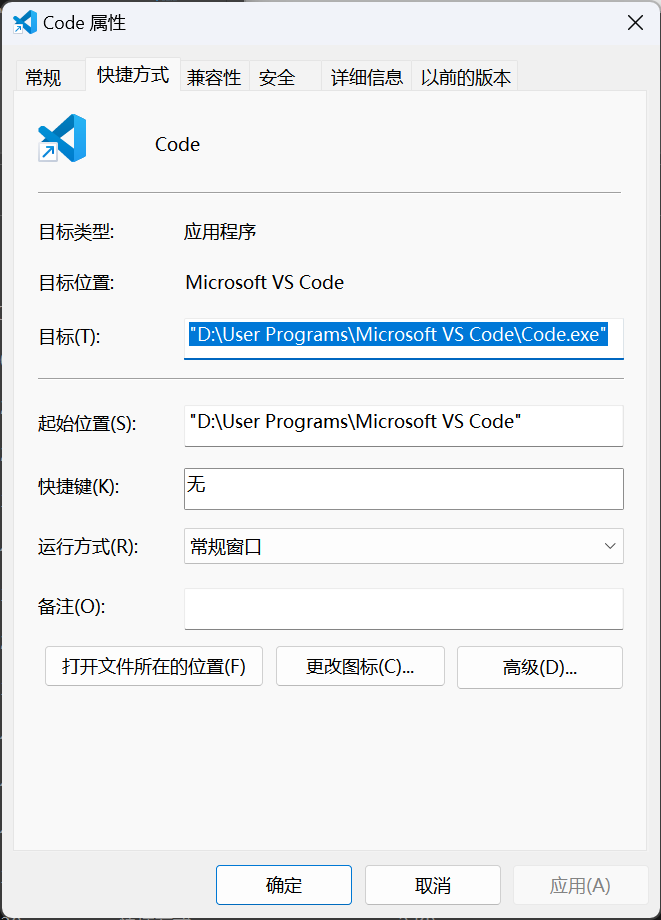
如果你找不到或忘记了 VSCode 的安装路径，可以按照如下的操作进行寻找（Windows 环境）。

(1) 按下 win 键或单击 windows 图标，在搜索栏中输入 Code，即可看到 VSCode 应用。

(2) 如下图所示，在弹出的对应搜索结果上右键，选择“打开文件位置”，进入快捷方式文件夹。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/vscode_search.png)

(3) 对快捷方式右键，选择属性，即可看到 VSCode 的程序路径位置。

[](https://soc.ustc.edu.cn/Digital/lab1/figs/verilog_coding/vscode_search_2.png)

**Step3：**确认无误后，点击 OK 即可完成配置。此时在 Vivado 中单击 .v 文件即可自动使用 VSCode 打开。

# 思考与练习

1. 参考上面的内容，配置VSCode开发环境。