

# Verilog环境搭建教程

## 参考资料：

- Windows版本
  - <https://zhuanlan.zhihu.com/p/338497672#:~:text=iveril#:~:text=iveril>
  - [https://soc.ustc.edu.cn/Digital/history/2023/lab1/verilog\\_coding/](https://soc.ustc.edu.cn/Digital/history/2023/lab1/verilog_coding/)（中科大安装verilog教程，Mac版本同样有效，推荐使用！）
- Mac版本：[Macbook M1使用vscode+iverilog+gtkwave实现Verilog代码的编译与运行-CSDN博客](#)

我们的lab将使用vscode与iverilog进行Verilog开发，以下是配置流程（以windows版本为例），lab仅需参考资料中配置的部分基础功能即可完成，有需要的同学可以自行根据教程配置更丰富的功能（助教自己有些功能也没配明白）

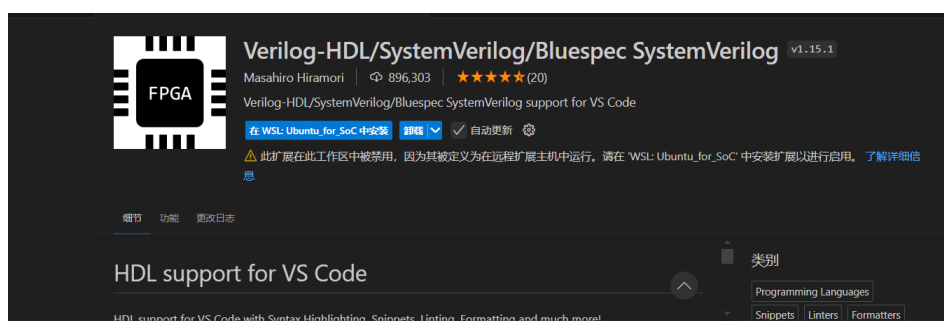
大家可以按照参考资料或者下面的流程进行配置，如果遇到问题可以先自行查看参考资料或者百度，如果解决不了找助教私聊沟通（参考资料比我写的详细很多）

## Step 1. VS Code下载安装

- 访问VS Code官方网站[Visual Studio Code - Code Editing. Redefined](https://code.visualstudio.com/)
- 选择符合自己系统需求的下载

## Step 2. VS Code中下载Verilog HDL

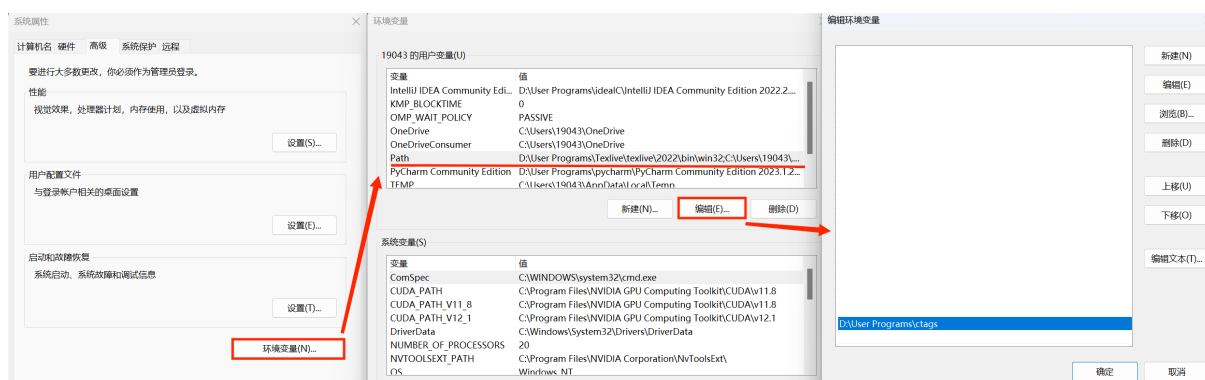
- 这是一个VS Code扩展插件，支持Verilog代码高亮等等功能，但并非Verilog的编译器或者仿真器，下载后需要配合编译器使用（我们使用的是iverilog编译器）



- 直接在vscode扩展中搜索Verilog找到这一栏选择安装即可

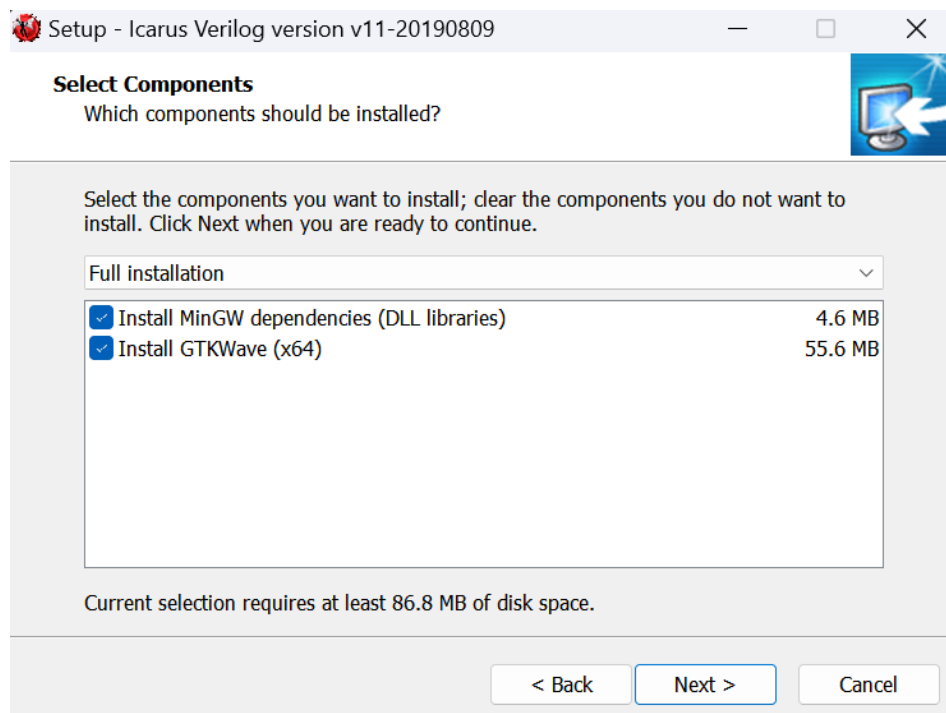
## Step 3. 配置Ctag

- ctag可以在verilog代码中支持快速转到变量定义等功能，较为常用，因此推荐大家安装，但不是必须的
- windows版本ctag安装：<https://github.com/universal-ctags/ctags-win32>
- Mac版本ctag：<https://github.com/universal-ctags/homebrew-universal-ctags>
- 下载完成后需要将包含 `ctags.exe` 文件的最底层路径添加到 Windows 系统环境变量中的 Path 中，如 `D:\verilog\ctag\ctags.exe`，具体路径为：设置 → 系统 → 关于 → 高级系统设置 → 环境变量 → Path → 新建



## Step 4. 配置iverilog

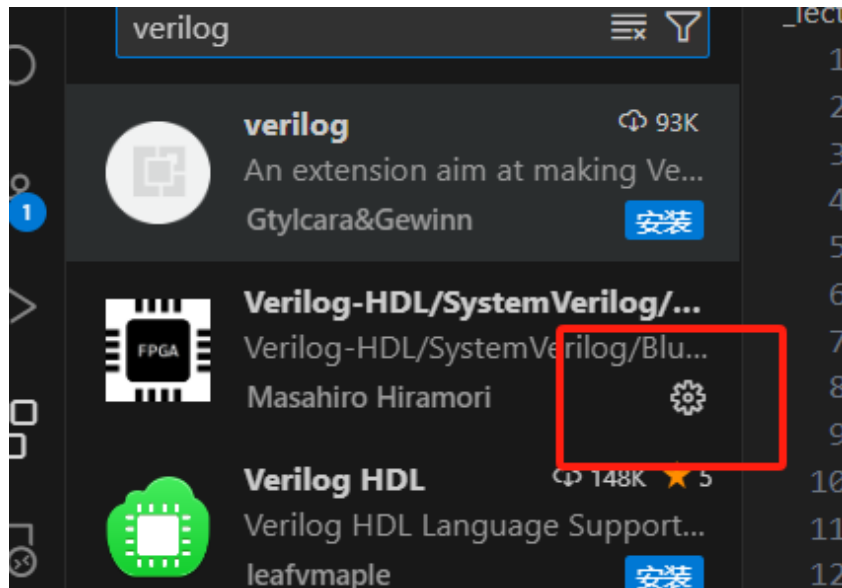
- 可以在官网下载iverilog的安装文件  
[https://soc.ustc.edu.cn/Digital/history/2023/lab1/verilog\\_coding/](https://soc.ustc.edu.cn/Digital/history/2023/lab1/verilog_coding/)
- 下载后同时选择安装iverilog与gtkwave



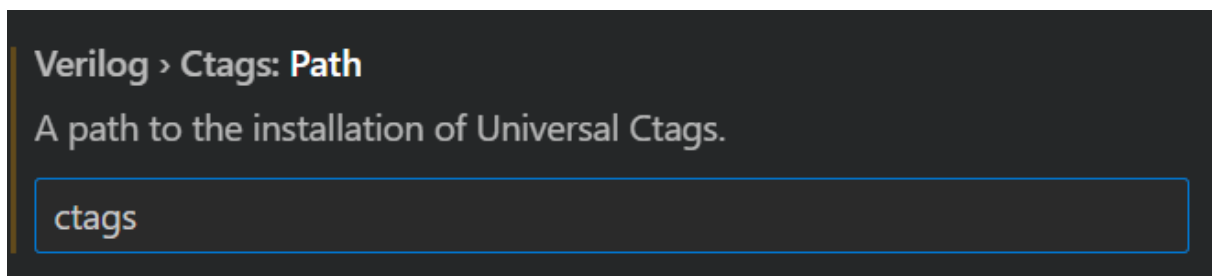
- 接下来一路 next 即可。**注意：iverilog 的安装目录不能有中文和空格。**如果想省事可以在安装时勾选添加环境变量（Add folder to PATH）。可以在刚刚ctag那个同样方法打开Path，检查一下是否有iverilog相关路径，要是没有手动添加一下，下面是助教电脑环境的例子（iverilog相关的Path）
  - `D:\verilog\iverilog\bin\iverilog.exe`
  - `D:\verilog\iverilog\bin`
  - `D:\verilog\iverilog\gtkwave\bin`

## Step 5. 配置VS Code

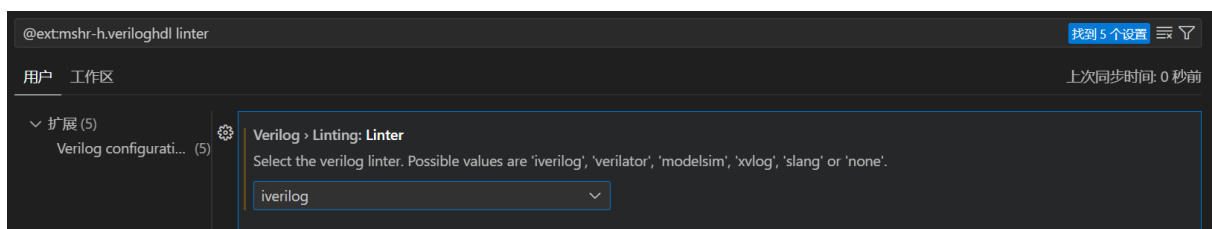
- 上述两步完成后，需配置VS code内部插件使用上述两步的工具
- 打开 Verilog-HDL 插件的设置界面（在扩展界面单击右下角的齿轮图标）



- 如下图所示将 Ctags:Path 设置为默认的 `ctags`



- 如下图所示，将 Verilog > Linting:Linter 设定为 iverilog



## Step 6. 测试环境是否work

- 创建简单的verilog程序，例如以下 `add.v`

```
module add(
    input a,
    input b,
```

```

        output [1:0] out
    );

    assign out = a + b;

endmodule

```

- 创建简单的testbench `add_tb.v`

```

`timescale 10ns/10ns
`include "add.v"

module add_tb();
    reg a, b;

    wire [1:0] out;

    add dut(
        .a(a),
        .b(b),

        .out(out)
    );

    initial begin
        a = 0;
        b = 0;

        #10
        a = 1;
        b = 0;

        #10
        a = 0;
        b = 1;

        #10
        a = 1;

```

```

        b = 1;

        #10
        $finish;

    end

    initial begin
        $dumpfile("wave.vcd");
        $dumpvars(0, add_tb)

    end

endmodule

```

- 接下来我们将进行对上述加法器文件的仿真，打开VS Code中的terminal，输入以下

```

iverilog -o a.out add_tb.v
vvp -n a.out
gtkwave wave.vcd

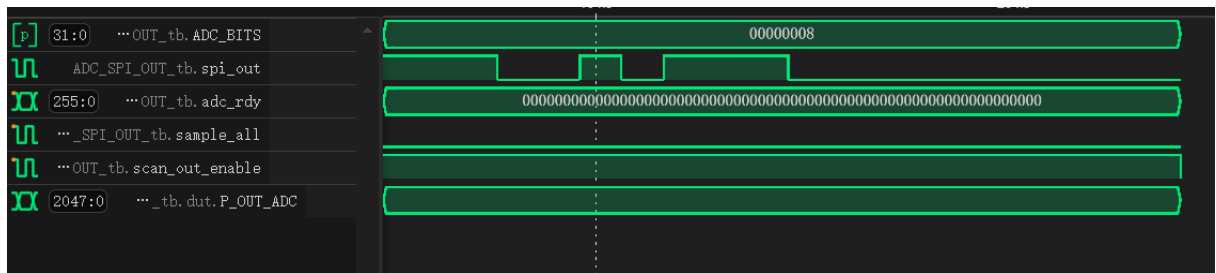
```

- 此时应该可以看到gtkwave的窗口打开，可以查看波形了

至此已经完成环境配置，同学们可以自己写一个简单的decoder验证一下是否成功安装，并且熟悉一下Verilog语法

## Step 7.（可选项）波形查看插件WaveTrace安装

- GTKwave虽然免费，但是在助教电脑上遇到很长的仿真波形缩放的时候会卡住。。。
- 可在VS Code上面安装WaveTrace扩展用来快捷查看仿真波形，有WaveTrace之后在VS Code中直接点击.vcd文件即可打开查看波形



- 但是该插件免费版只支持有限数量的波形同时显示。。。