

如何使用VSCode编写并测试Verilog程序？

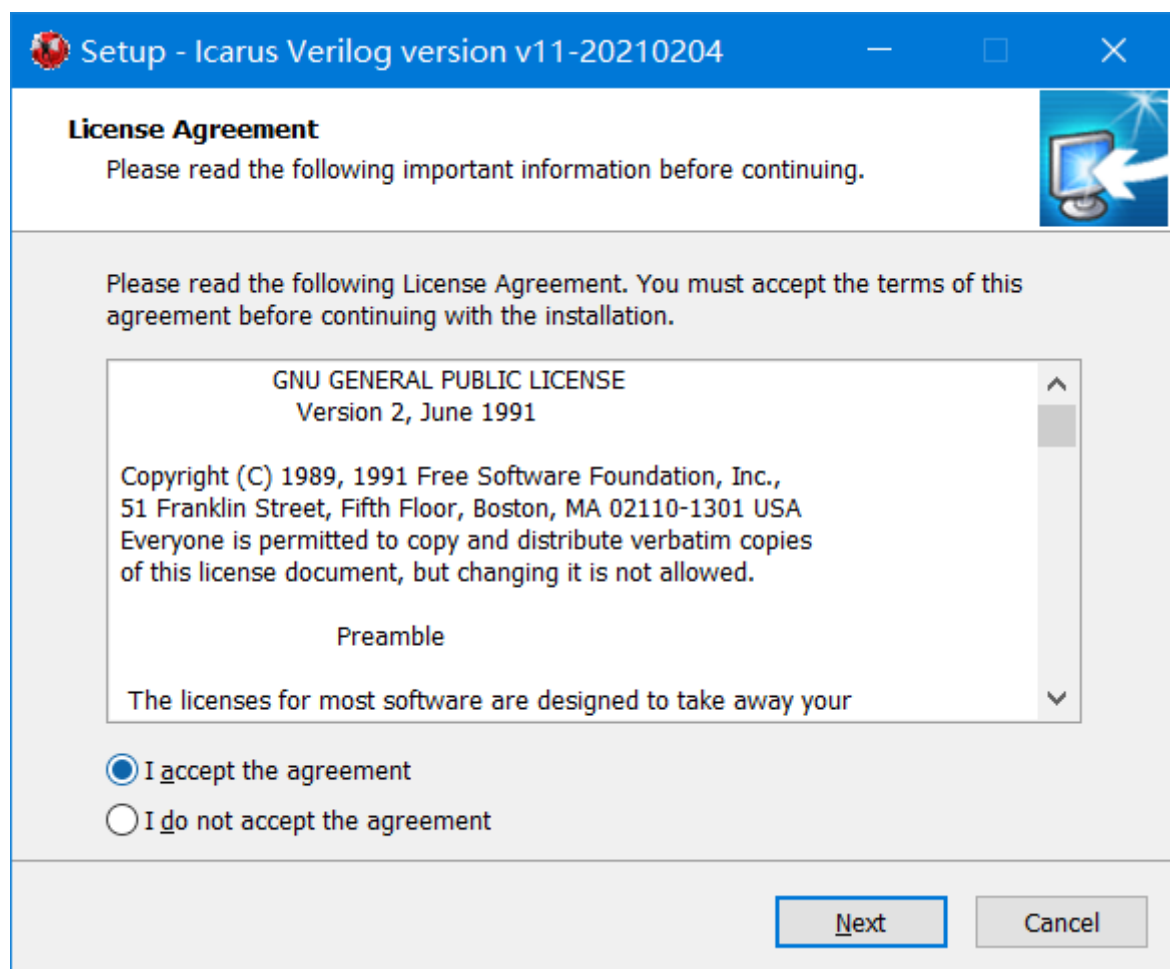
第一步：安装VSCode

首先确定你在电脑安装的是 Visual Studio Code，而不是 Visual Studio 2019/2017/2012等软件

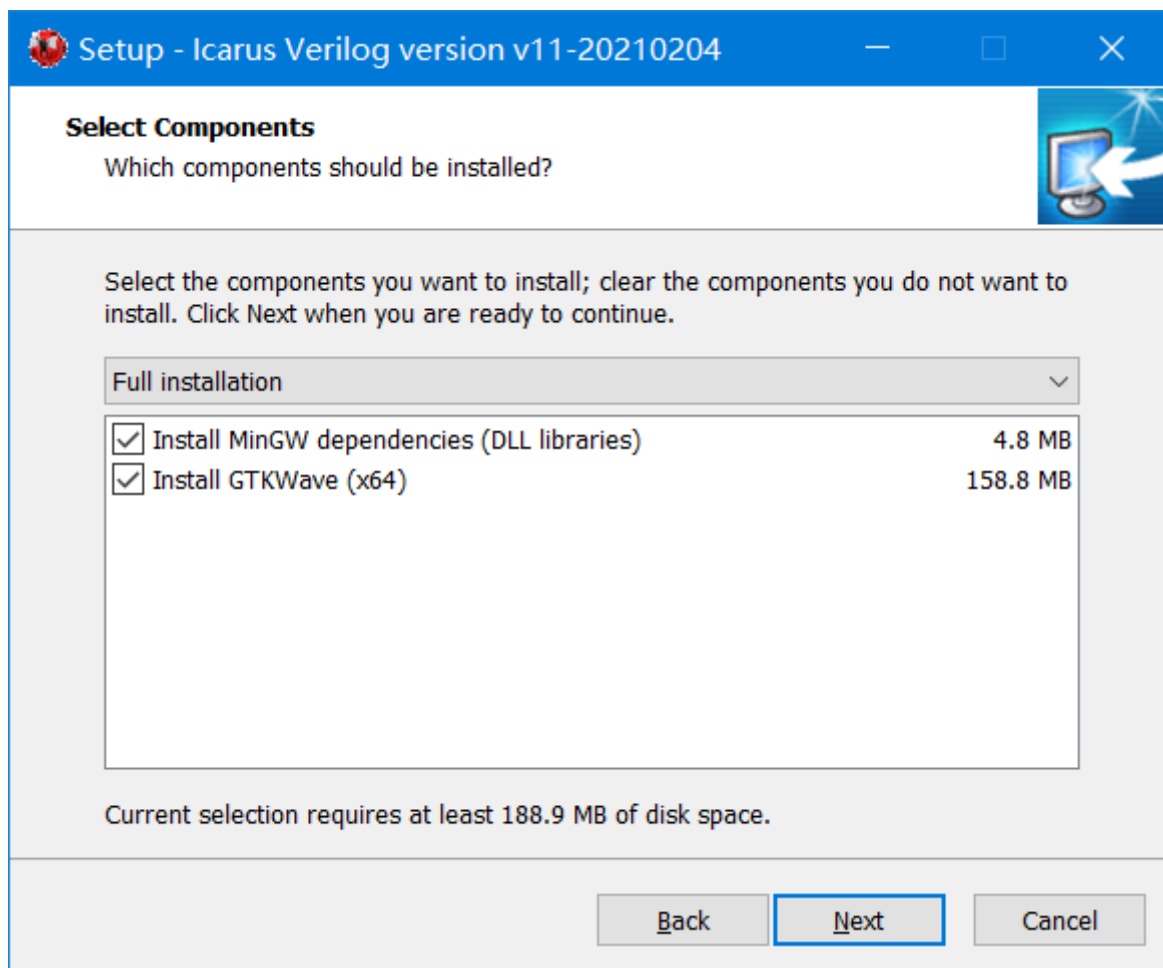
如果你没有安装，这里有链接<https://bhpan.buaa.edu.cn:443/link/7FEF5347A22666A147EF74BDEB6F47F9>，也可以通过官网下载，但是速度非常缓慢

第二步：安装iVerilog程序与GTKWave波形查看工具

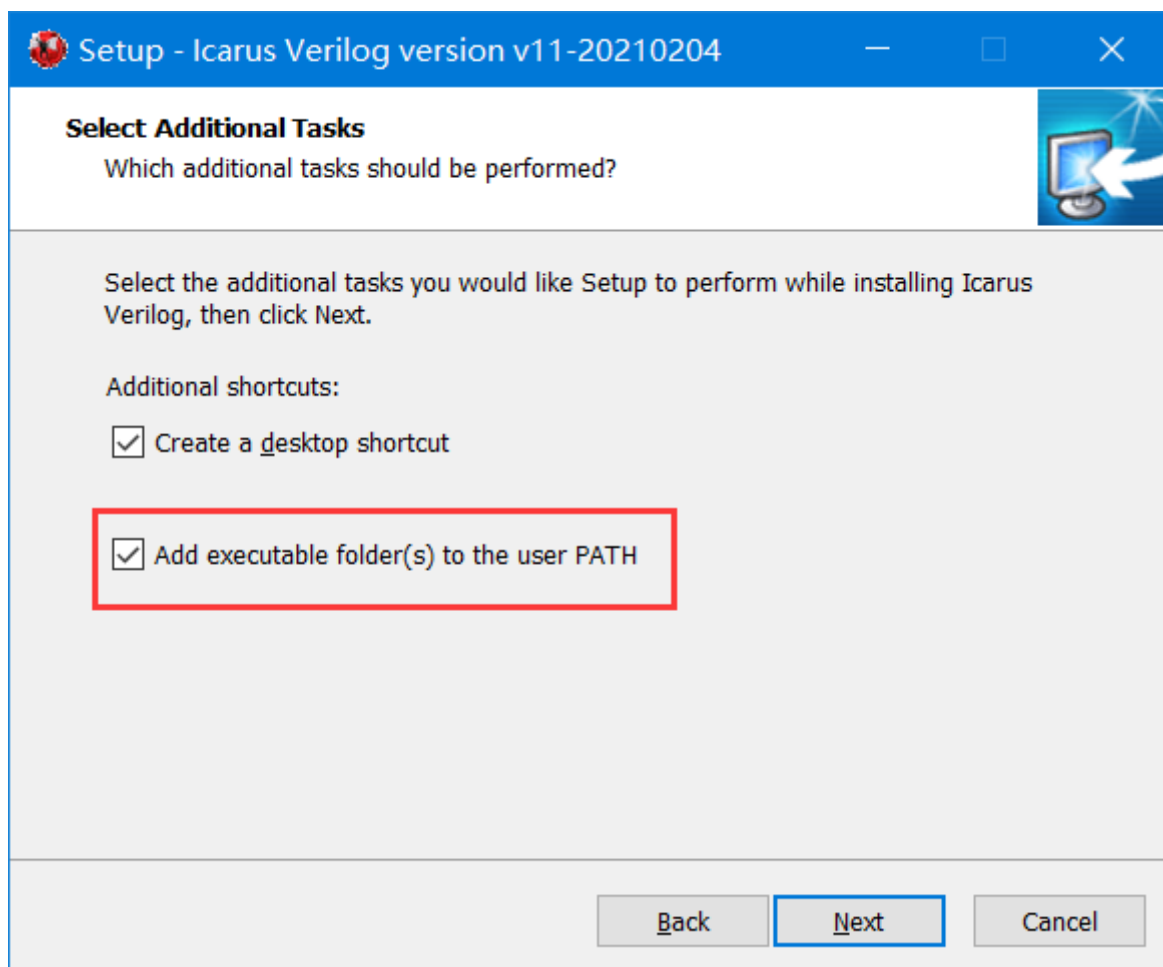
同样，下载链接也已经放在了云盘上<https://bhpan.buaa.edu.cn:443/link/F7904CA675A8362CEE2139EE7425FA49>



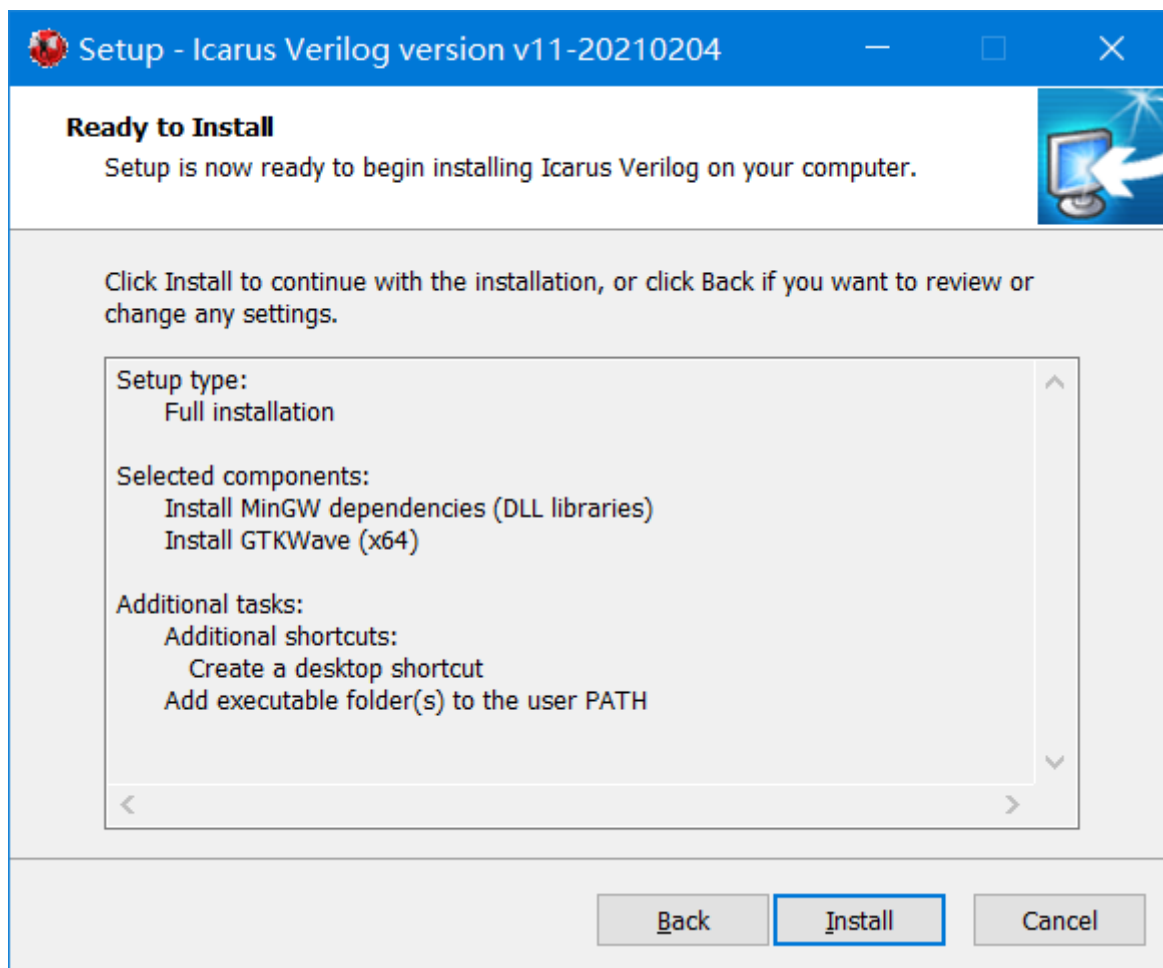
第一步肯定要同意



这里同时勾选安装MinGW依赖和GTKWave波形查看程序



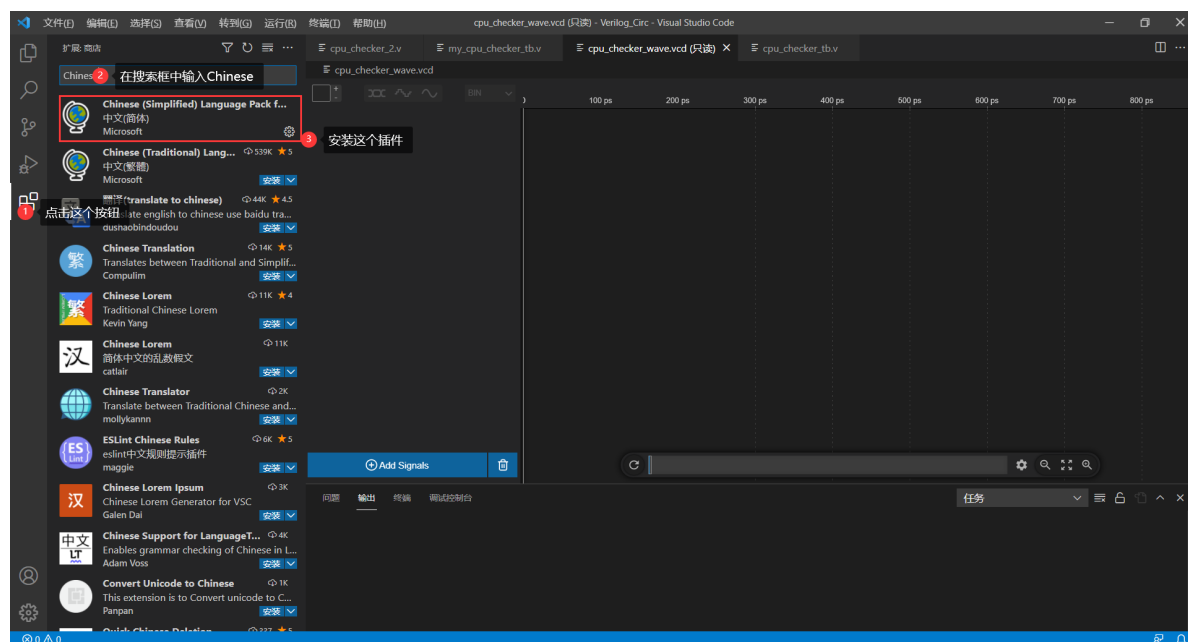
红色方框记得一定要勾选



然后开始安装即可

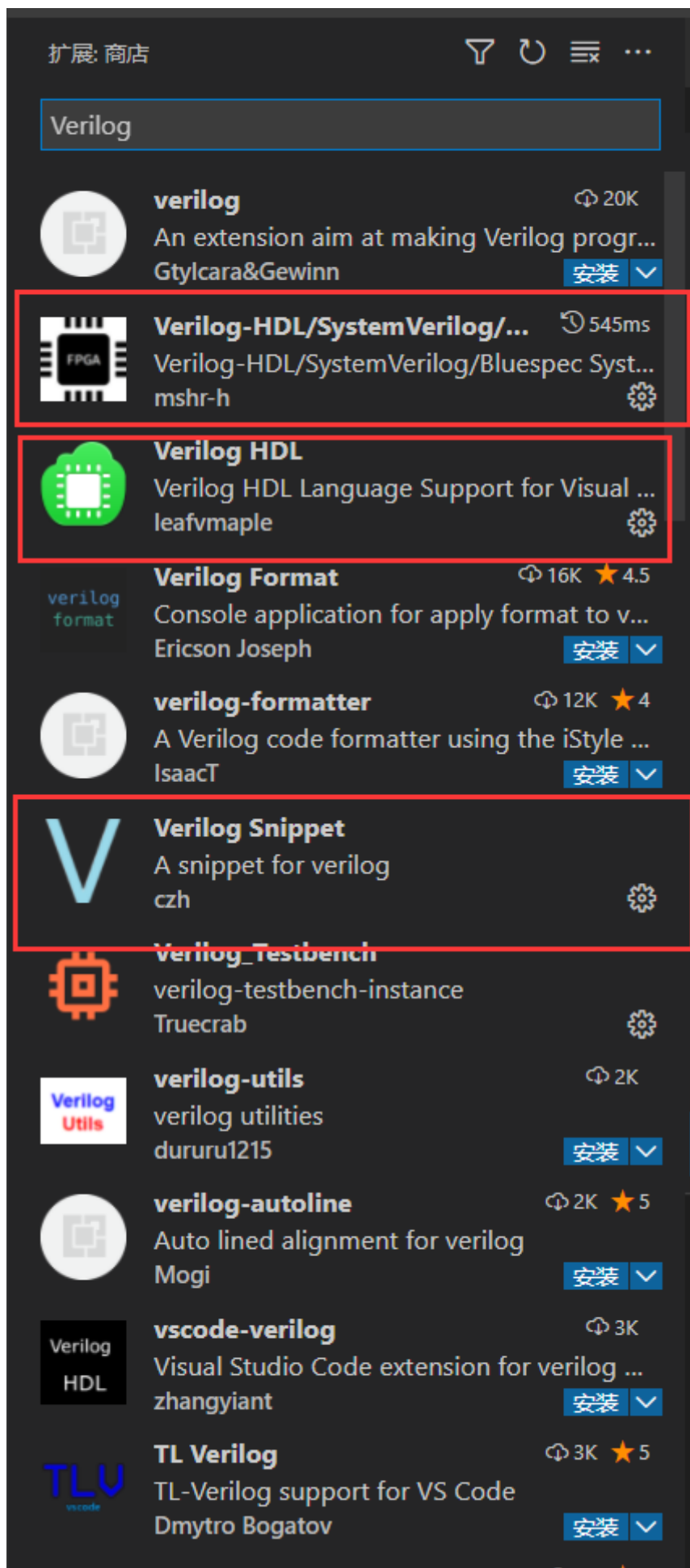
第三步：安装VSCode插件

你的VSCode打开以后可能是英文，这时候



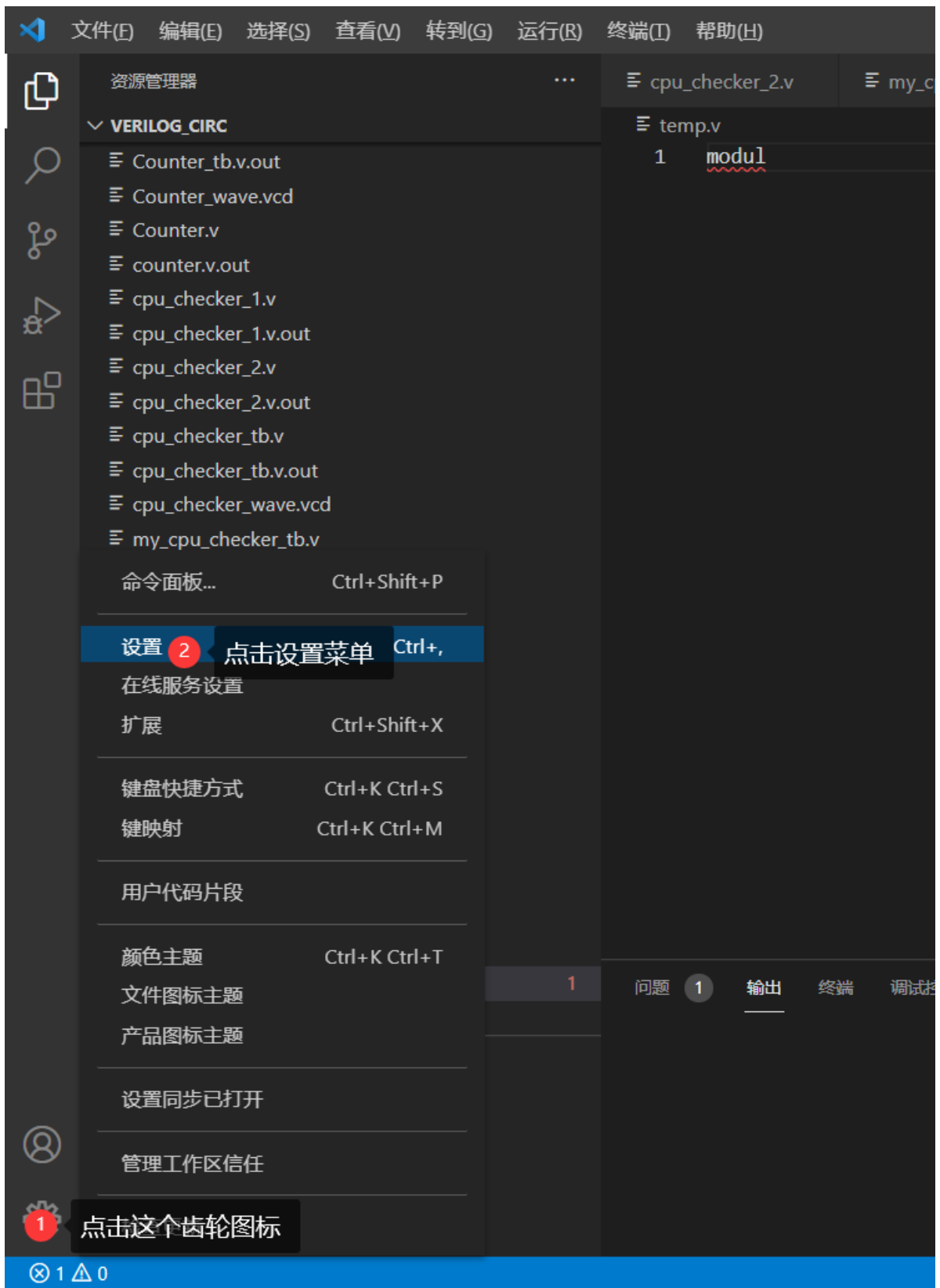
然后重启VSCode就可以看到中文界面了

要想在VSCode中写Verilog代码，安装以下三个插件

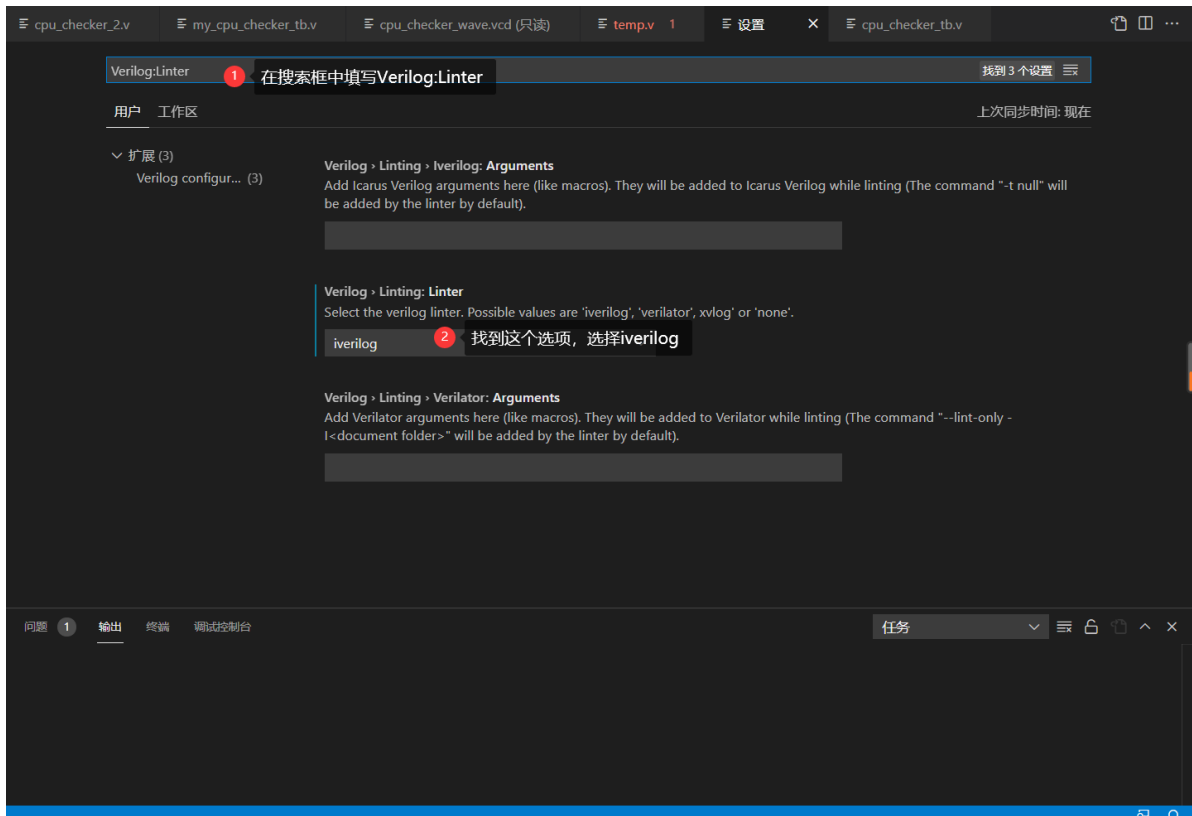


就是上面画红色框框的两个

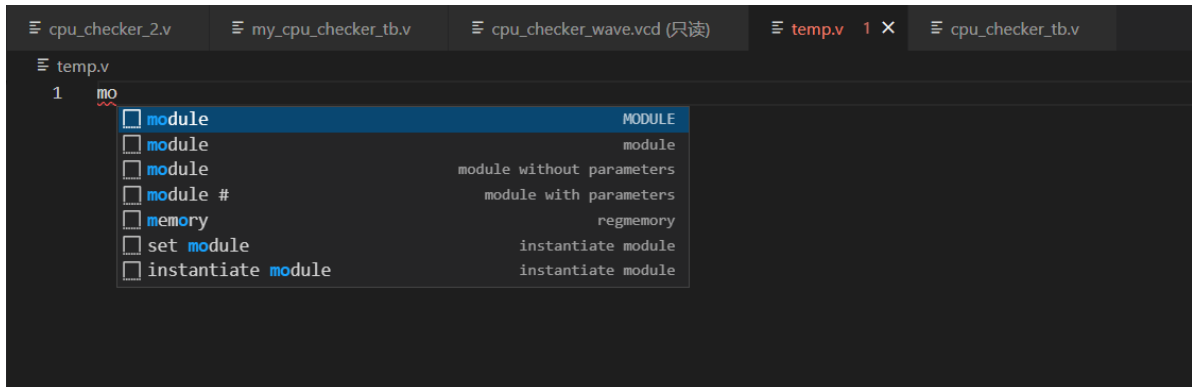
然后重启VSCode，随便打开一个文件夹，新建一个temp.v的文件



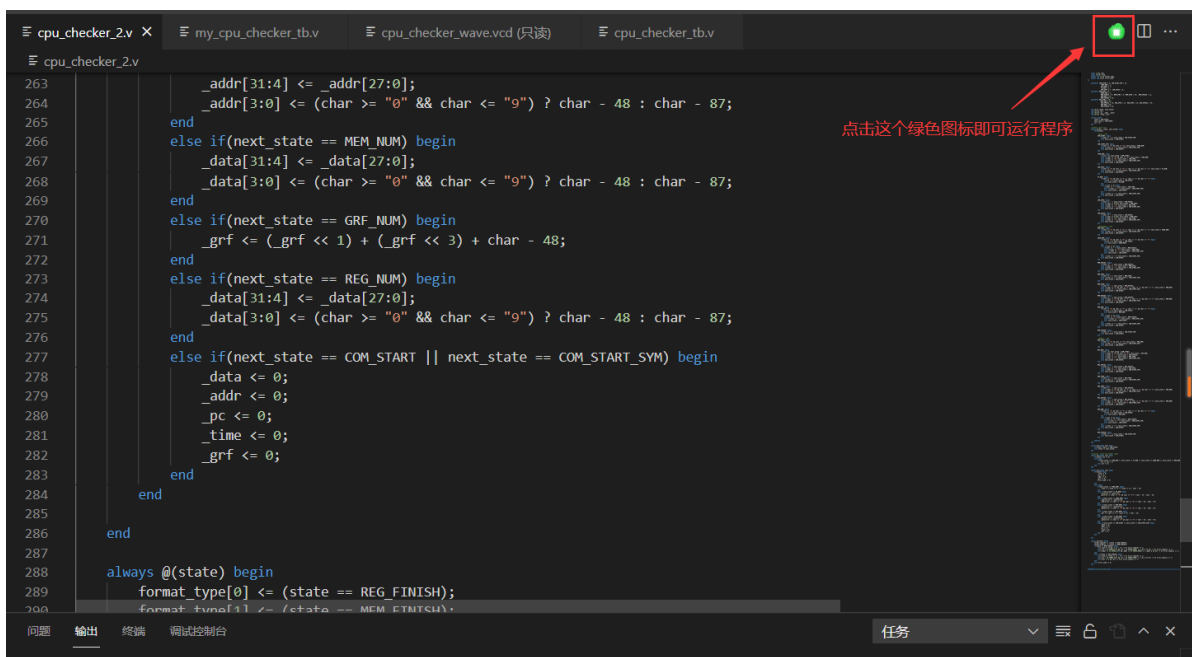
然后你会看到这个界面，按下图操作



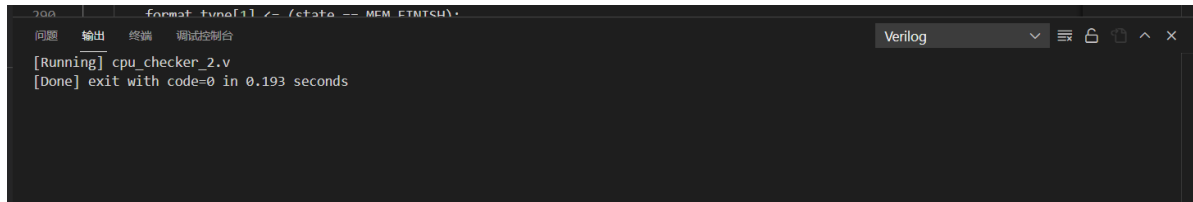
这时候你输入代码就可以实现代码补全功能了



第四步：运行Verilog程序

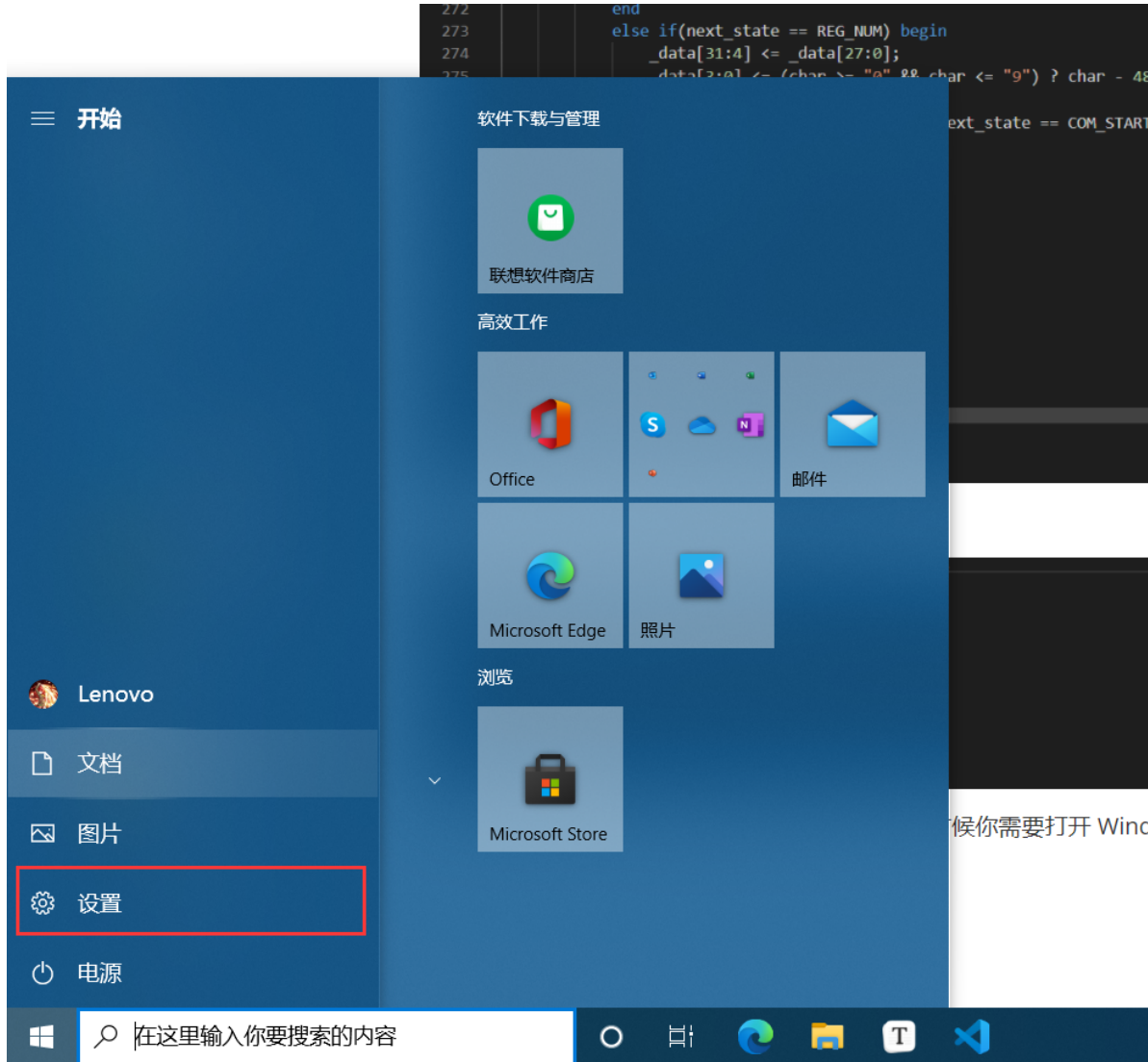


如果一切顺利，你在下面的输出窗口中将看到



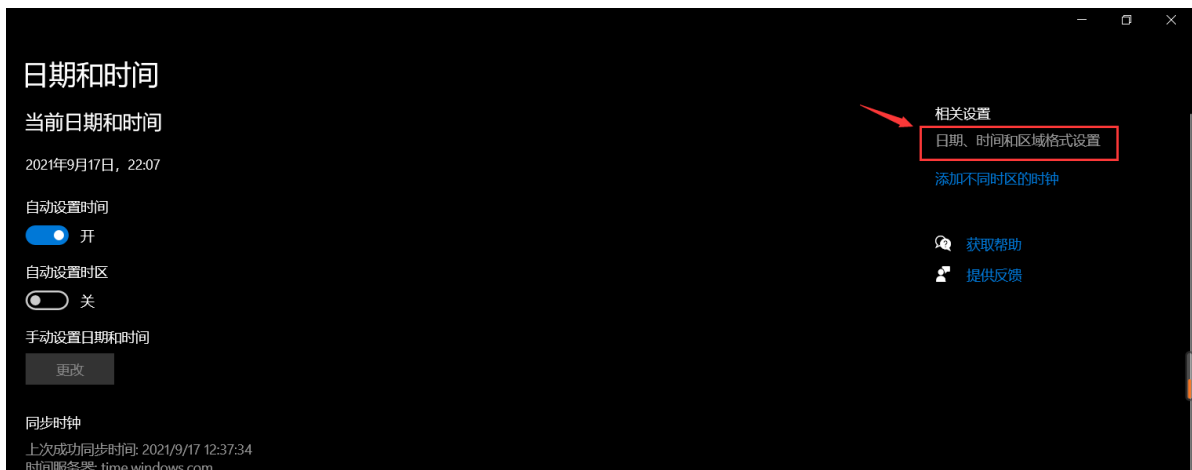
```
format_tune[1] /- (cstate == MEM_ETMISU).
问题 输出 终端 调试控制台
[Running] cpu_checker_2.v
[Done] exit with code=0 in 0.193 seconds
```

但是你也很有可能很不幸的遇到**输出乱码**的情况，这时候你需要打开 Windows设置

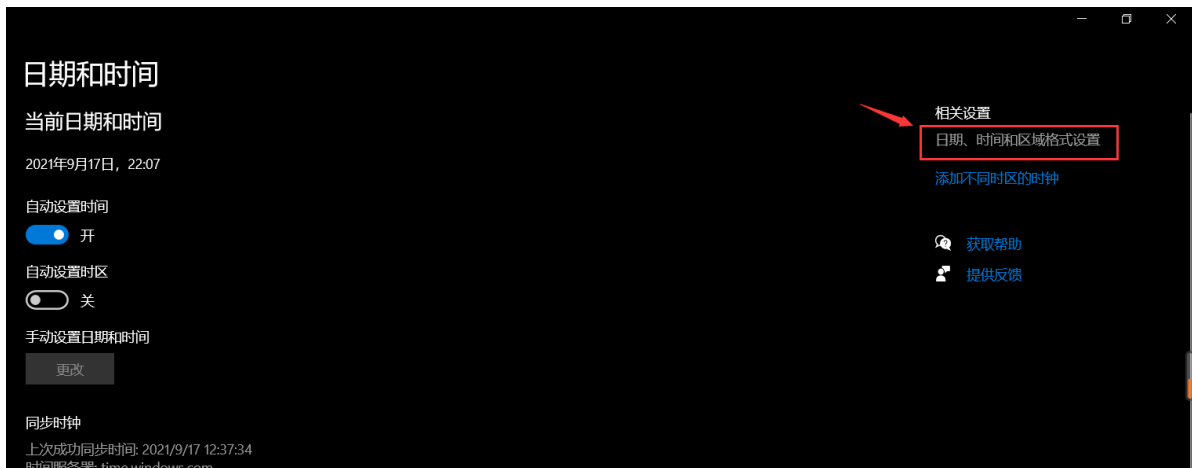


选择时间和语言





再点击这个日期、时间和区域格式设置



再点击这个其他设置

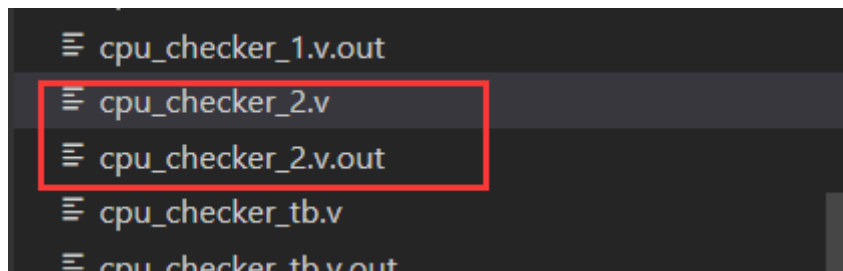


再这样操作



然后重启计算机就可以了

如果编译顺利，你在VSCode左侧的文件管理窗口中将看到生成了一个*.v.out 文件



第四步：查看仿真波形

有两种方法，但是任何一种方法都需要你手动自行编写TestBench测试文件

假定我们已经写好了 my_cpu_checker_tb.v

```
`timescale 1ns / 1ps
`include "cpu_checker_2.v"
```

首先，文件头部要有控制仿真时间粒度的 ``timescale` 语句，同时要包含你想测试的模块文件，在本例中是 `cpu_checker_2.v`

接着要实例化你所写的模块，如下图所示

```
// Instantiate the Unit Under Test (UUT)
cpu_checker uut (
    .clk(clk),
    .reset(reset),
    .char(char),
    .freq(16'd256),
    .format_type(format_type)
);
```

然后在 `initail` 语句块中加入下面的语句

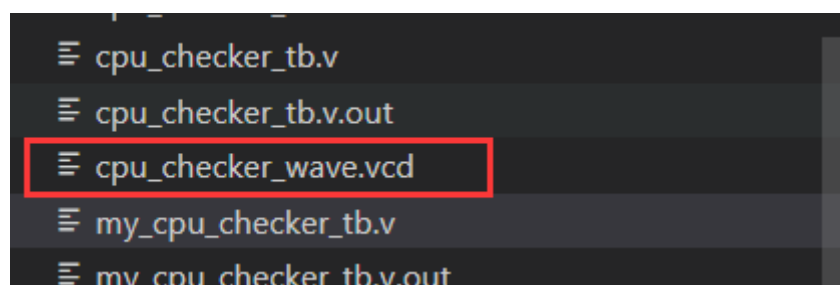
```
32     initial begin
33         $dumpfile("cpu_checker_wave.vcd");
34         $dumpvars;
35         // Initialize Inputs
36         clk = 0;
37         reset = 1;
38         char = 0;
39         freq = 2;
40         finish = 0;
41
42         #10 reset = 0;
43     #2 char = "^";
44     #2 char = "2";
45     #2 char = "4";
46     #2 char = "2";
47     #2 char = "2";
48     #2 char = "@";
49     #2 char = "0";
50     #2 char = "0";
51     #2 char = "0";
52     #2 char = "0";
53     #2 char = "3";
54     #2 char = "0";
```

其中 `$dumpfile` 指定波形文件名, `$dumpvars` 表示所有定义的 `wire` 和 `reg` 都参与仿真

下面的 `#2 char = "..."` 都是要进行测试的数据, 可以自行设计

然后运行 `my_cpu_checker_tb.v` 程序, 还是点击右上角的绿色图标

如果顺利的话, 你将在右侧文件管理器中看到



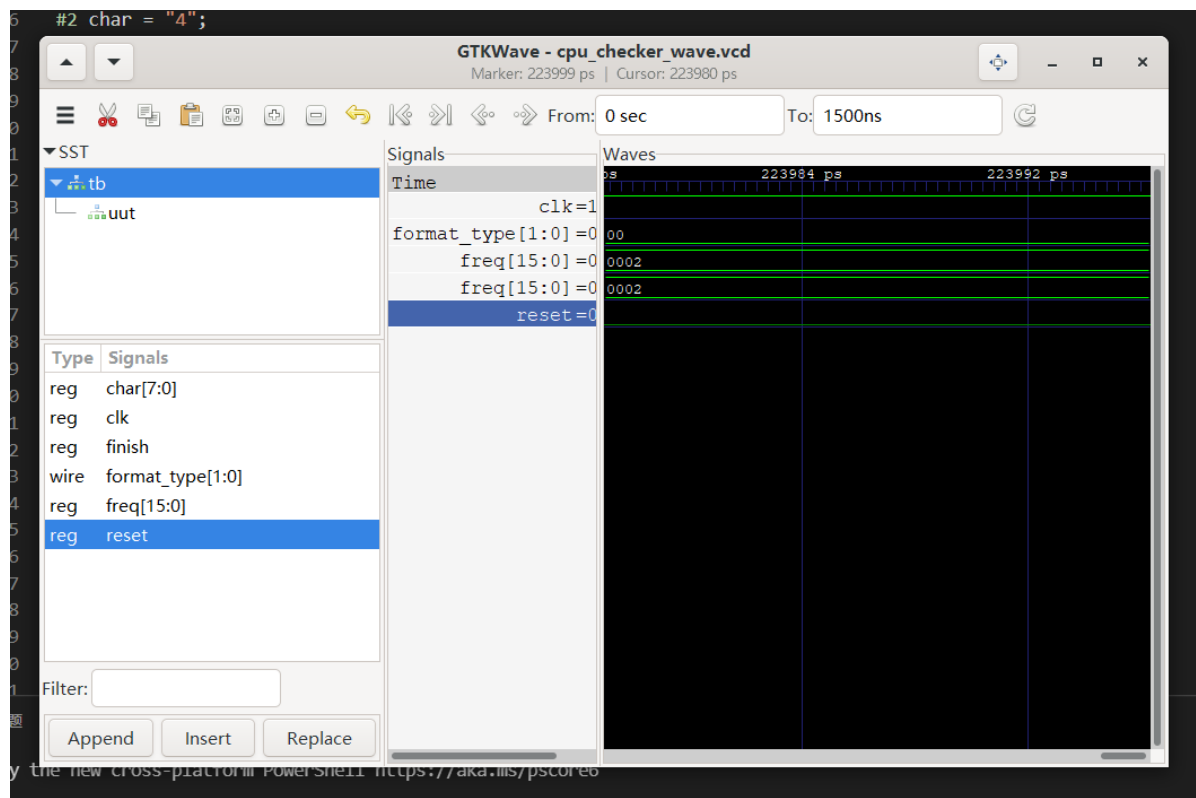
这就是我们需要的波形文件

方法一：使用GTKWave

如下图操作

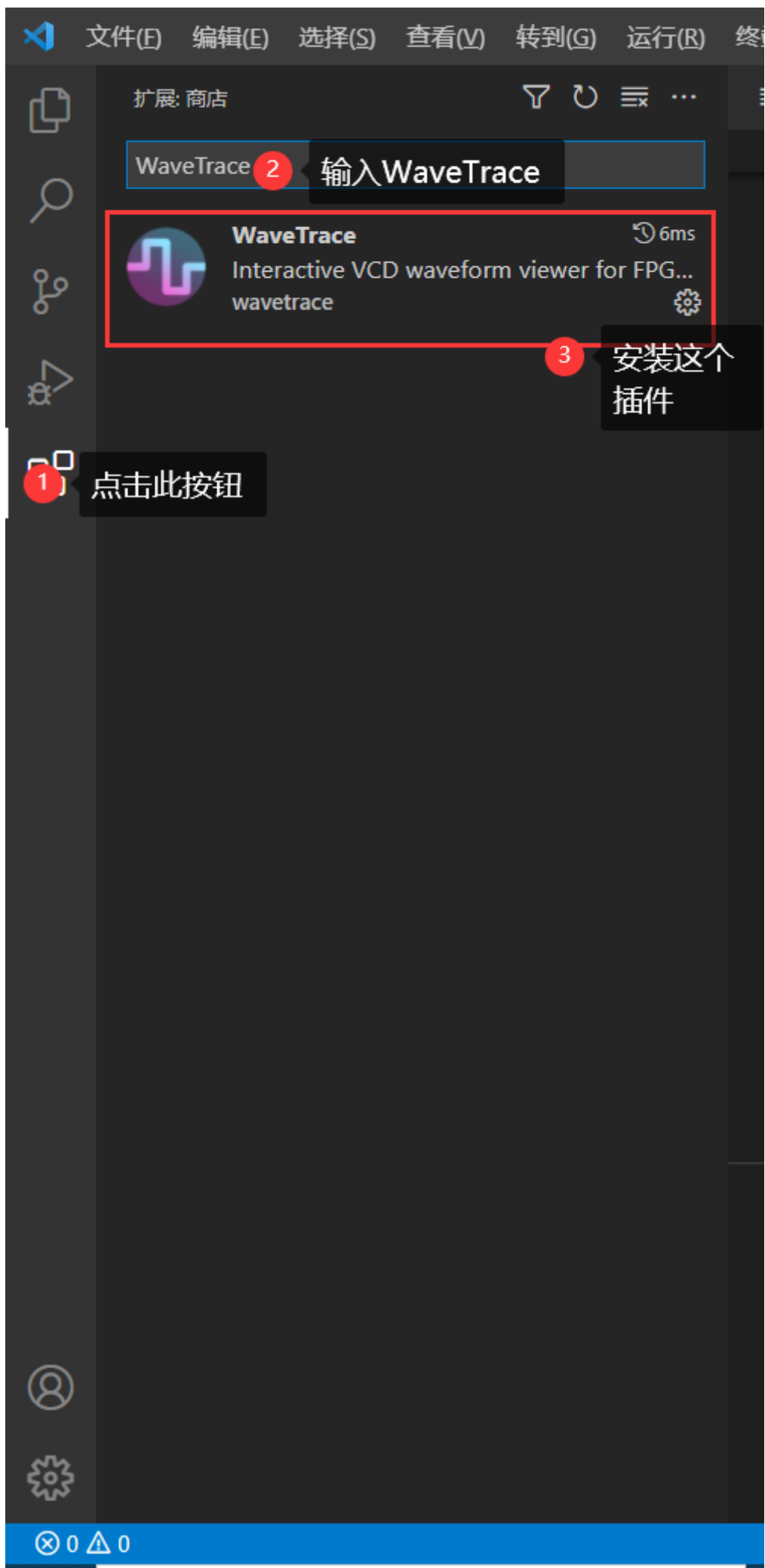


然后可以自行探索此软件的功能, 反正我也没用过



方法二：利用VSCode插件

如下图操作



然后我们可以直接在VSCode中点击左侧文件管理器中的 *.vcd 文件打开，如图所示，就可以看到炫酷的波形图像了

