

lab 0-1：数字电路仿真环境准备

1. 实验环境

- EDA 工具：[Ngspice](#), [Logisim-evolution](#)
- 操作系统：~~Windows 11 on arm 25H2 (Parallels Desktop), Ubuntu 24.04 with Rosetta (Parallels Desktop)~~
 - Windows 11 24H2, Ubuntu 24.04(WSL)

生命不息，折腾不止。

虚拟机已死，换回 Windows了。过一阵子试试docker。

先后尝试并搞崩了多种虚拟机软件和多台机器，最终决定使用M芯片的Mac，并找到了合适的环境(PD虚拟机，
下载链接：<https://macwk.cn/app/2941.html>，供学习使用)。该环境可以较流畅地在Windows系统运行x86转译的vivado且电源管理优秀，无需担心续航和发热问题。希望能给同样用类似huan'j的同学一些有用的参考。

其中markdown软件采用Typora+plugin (Windows) 。

- VHDL: Verilog

2. 实验报告

2.1 反相器的电压传输特性 20%

2.1.1 相关实验步骤 5%

工具安装

SPICE 仿真

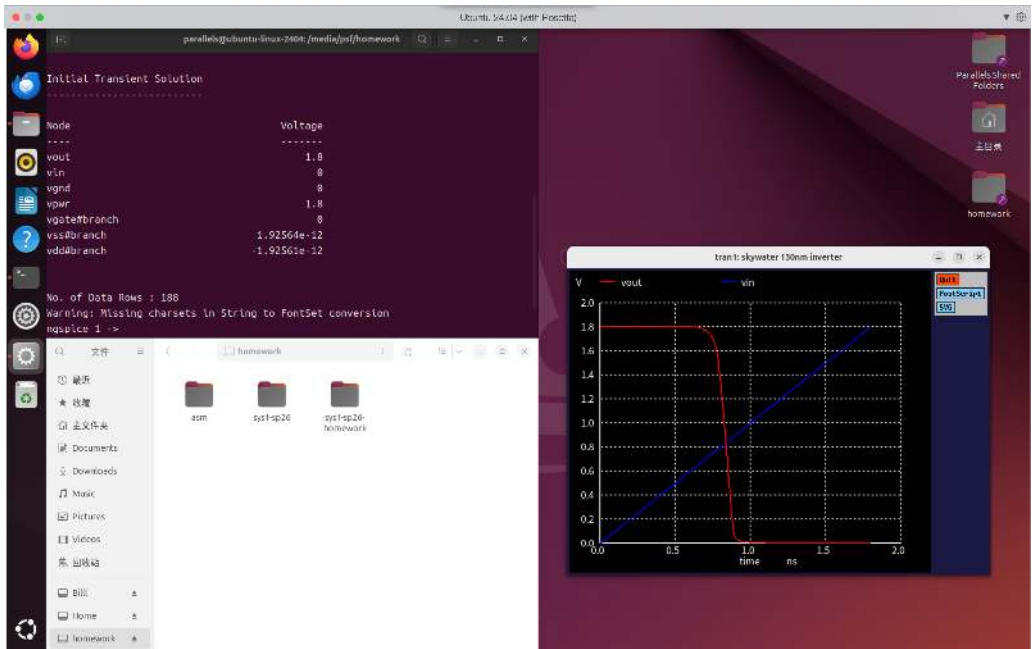


Image 1

2.1.2 描述执行 `ngspice inv.sp` 得到的输入输出关系 5%

- 总体上, v_{out} 越高, v_{in} 越低; 反之, v_{in} 越高, v_{out} 越低。
- 反相器实现取反逻辑。
- 观察 v_{out} , 在 v_{in} 较低时保持1.8V, v_{in} 超过约0.9V阈值电压后, v_{out} 迅速下降到0V。这种突变类似开-关的变化。

如果没有突变而是线性的 v_{out} , 其误差会累计, 抗干扰能力差。

2.1.3 观察 V_{out} , 思考如果给 V_{in} 接入一个正弦变化的电压, 输出的 V_{out} 是什么样的, 可以手绘 V_{out} - T 的结果加以文字描述 5%

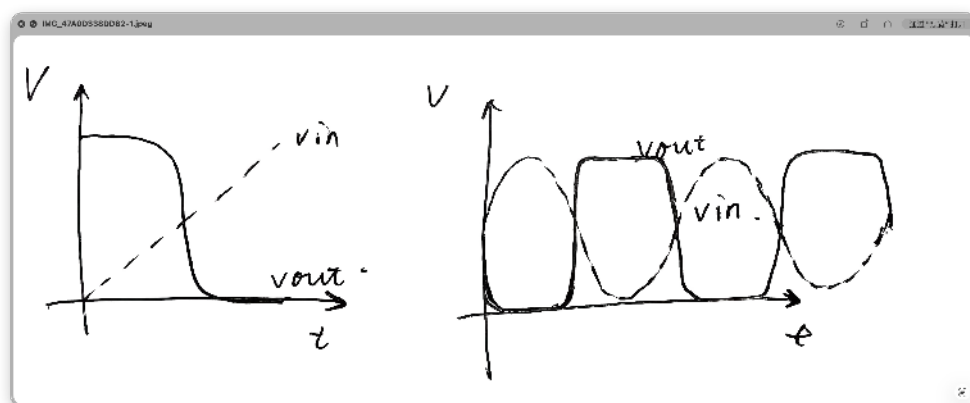


Image 2

- 正弦的 v_{in} , 整体上对应反相的 v_{out} 。但 v_{out} 有较大平台, 而不是正弦。

2.1.4 你认为对于该反相器, 逻辑 0 和逻辑 1 的阈值应该设定为多少 V, 即 V_{in} 输入的电压范围是多少可以认为输入的电压是 0/1 信号, V_{out} 输出的电压范围是多少可以认为输出的电压是 0/1 信号 5%

- v_{in} : 0.0-0.4对应0, 1.4-1.8对应1。(不宜采用0.9, 以提高容错性)
- v_{out} 0.0-0.2对应0, 1.6-1.8对应1。

2.2 Logisim 电路仿真 30%

2.2.1 截图除安装、保存外的相关实验步骤 10%

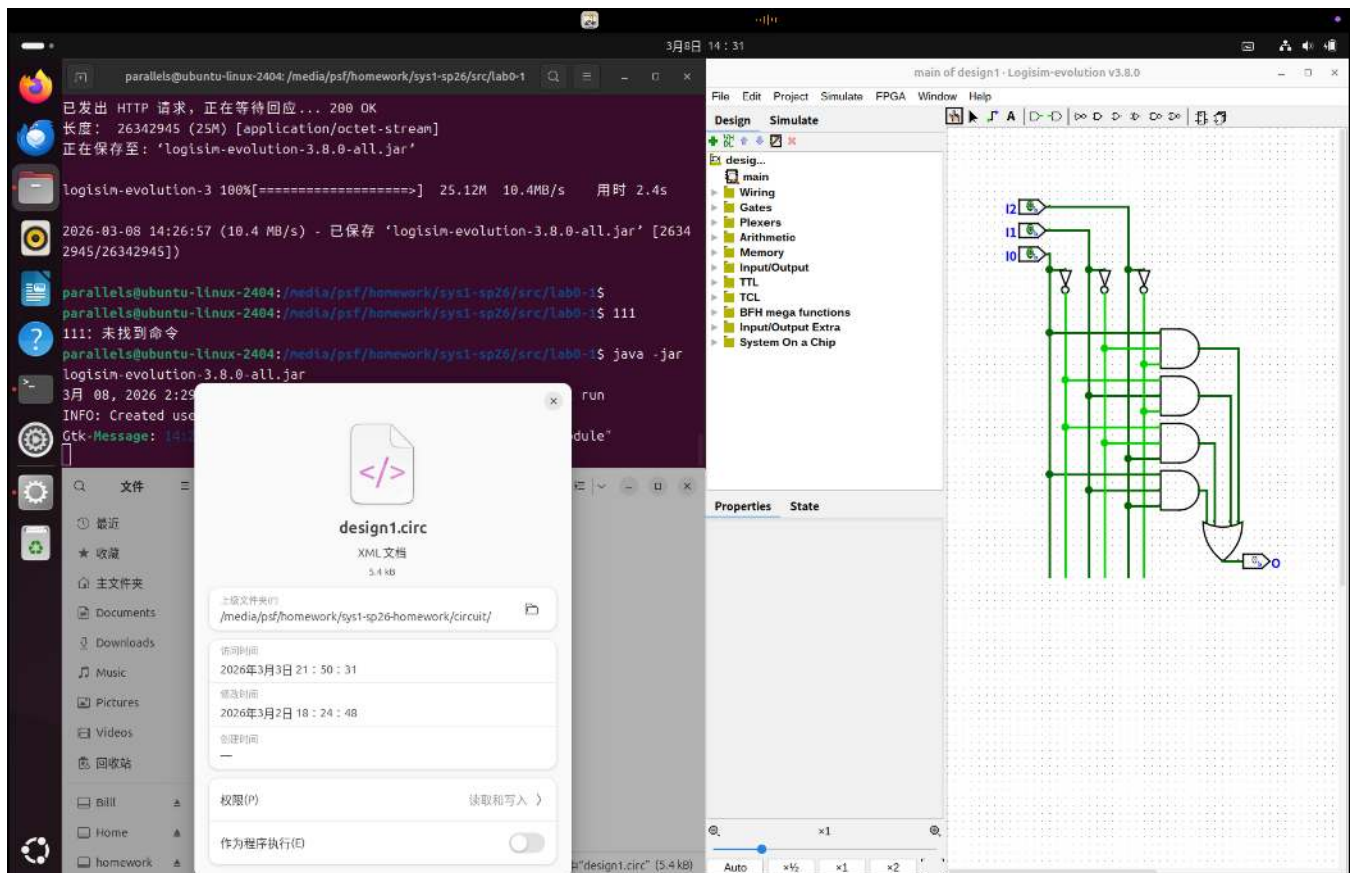


Image 3

2.2.2 遍历全部可能的输入，填写以下表格并总结该电路实现的功能 20%

Input			Output
I0	I1	I2	O
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

2.2.3 尝试总结输入输出关系和电路组成结构之间存在什么内在联系（选做）

- O与I的关系：

$$O = \overline{I_0} \overline{I_1} I_2 + \overline{I_0} I_1 \overline{I_2} + I_0 \overline{I_1} \overline{I_2}$$

- 奇数个信号时，输出为高电平；反之输出为低电平。

- 在上述电路图中，从上向下前三个AND是用来分别判断是否只有一个I是1，后一个AND来判断是否都是1。

3. 验收和代码提交 40%

3.1 验收检查点

- 反相器的截图 20%

截图见上

- Logisim 电路图和生成的工程文件 20%

电路图见上

3.2 提交文件

- 提交 [Logisim 保存设计](#)一节保存下来的 circuit 文件

lab 0-2: FPGA 实验环境准备

1. 实验环境

- 同上

2. 实验过程

2.1 导出之后的tree结构：

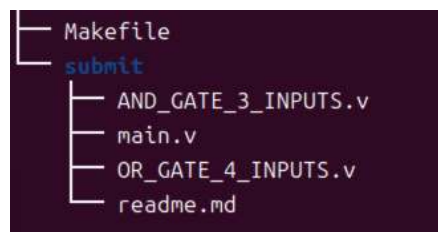


Image 4

2.2 仿真波形:

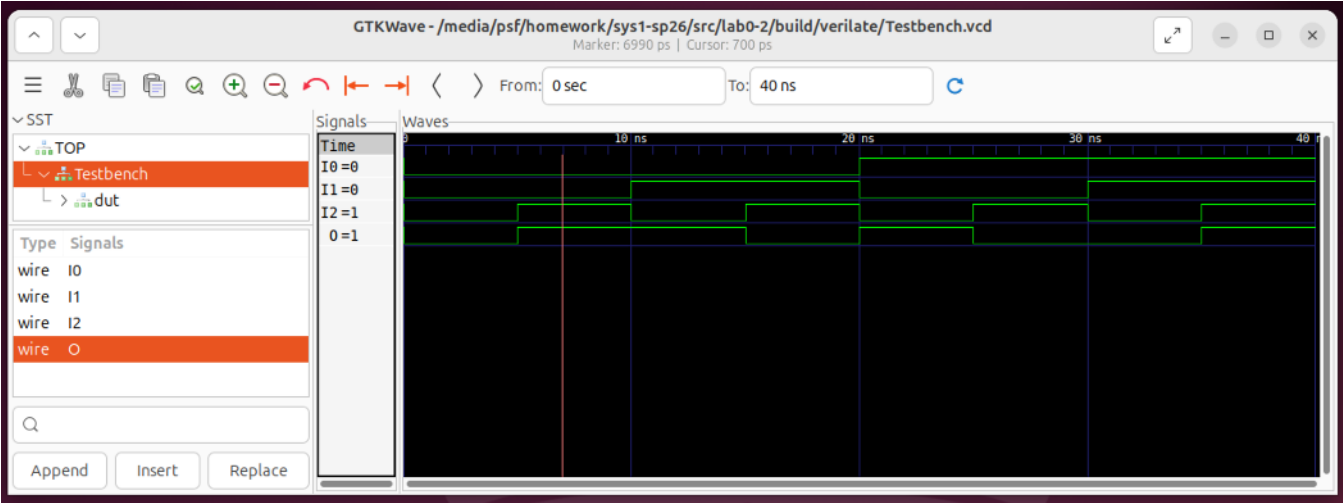


Image 5

2.3 综合:

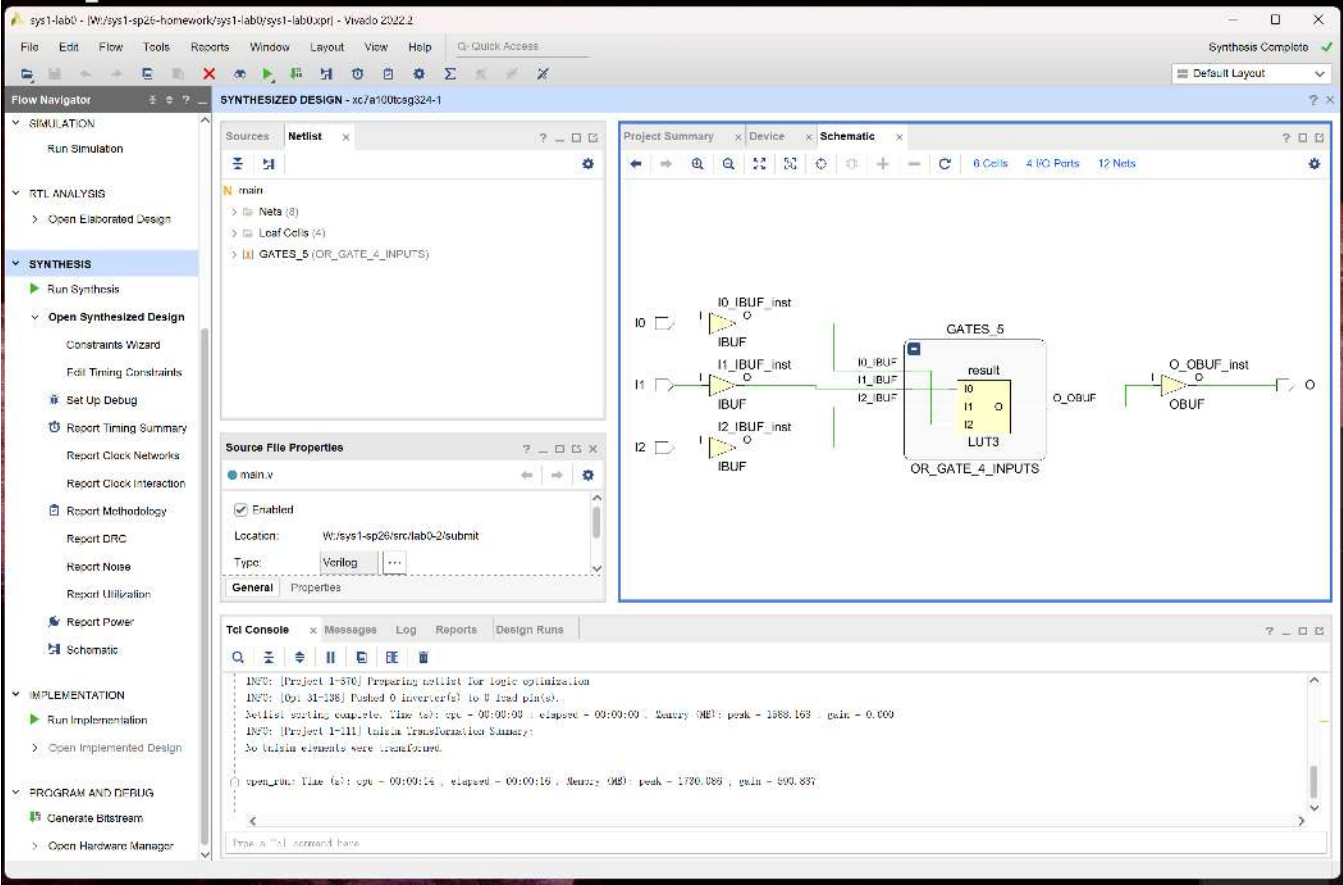


Image 6

2.4 构建:

```
Tcl Console  x Messages  Serial I/O Links  Serial I/O Scans

start_gui
create_project sys1-lab0 W:/sys1-sp26-homework/sys1-lab0 -part xc7a100tcsq324-1
add_files -norecurse {W:/sys1-sp26/src/lab0-2/submit/AND_GATE_3_INPUTS.v W:/sys1-sp26/src/lab0-2/submit/OR_GATE_4_INPUTS.v W:/sys1-sp26/src/lab0-2/submit/main.v}
update_compile_order -fileset sources_1
launch_runs synth_1 -jobs 8
open_run synth_1 -name synth_1
add_files -norecurse W:/sys1-sp26/repo/sys-project/lab0-2/syn/top.v
update_compile_order -fileset sources_1
update_compile_order -fileset sources_1
add_files -fileset constrs_1 -norecurse W:/sys1-sp26/repo/sys-project/lab0-2/syn/nexysa7.xdc
reset_run synth_1
launch_runs synth_1 -jobs 8
launch_runs impl_1 -jobs 8
close_design
open_run impl_1
launch_runs impl_1 -to_step write_bitstream -jobs 8
open_hw_manager
connect_hw_server -allow_non_jtag
open_hw_target
set_property PROGRAM.FILE {W:/sys1-sp26-homework/sys1-lab0/sys1-lab0.runs/impl_1/top.bit} [get_hw_devices xc7a100t_0]
current_hw_device [get_hw_devices xc7a100t_0]
refresh_hw_device -update_hw_probes false [lindex [get_hw_devices xc7a100t_0] 0]
```

Image 7

2.5 生成比特流:

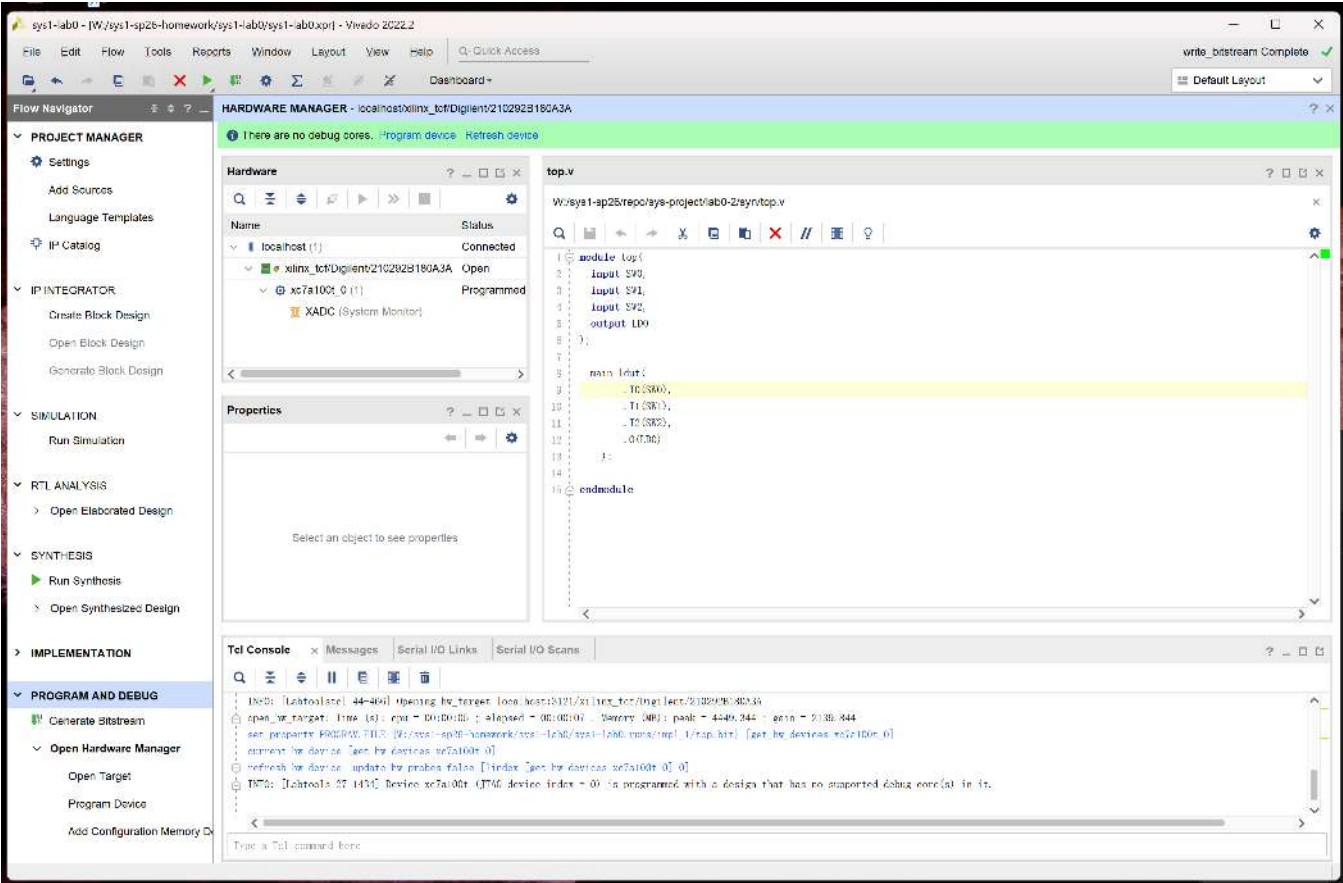


Image 8

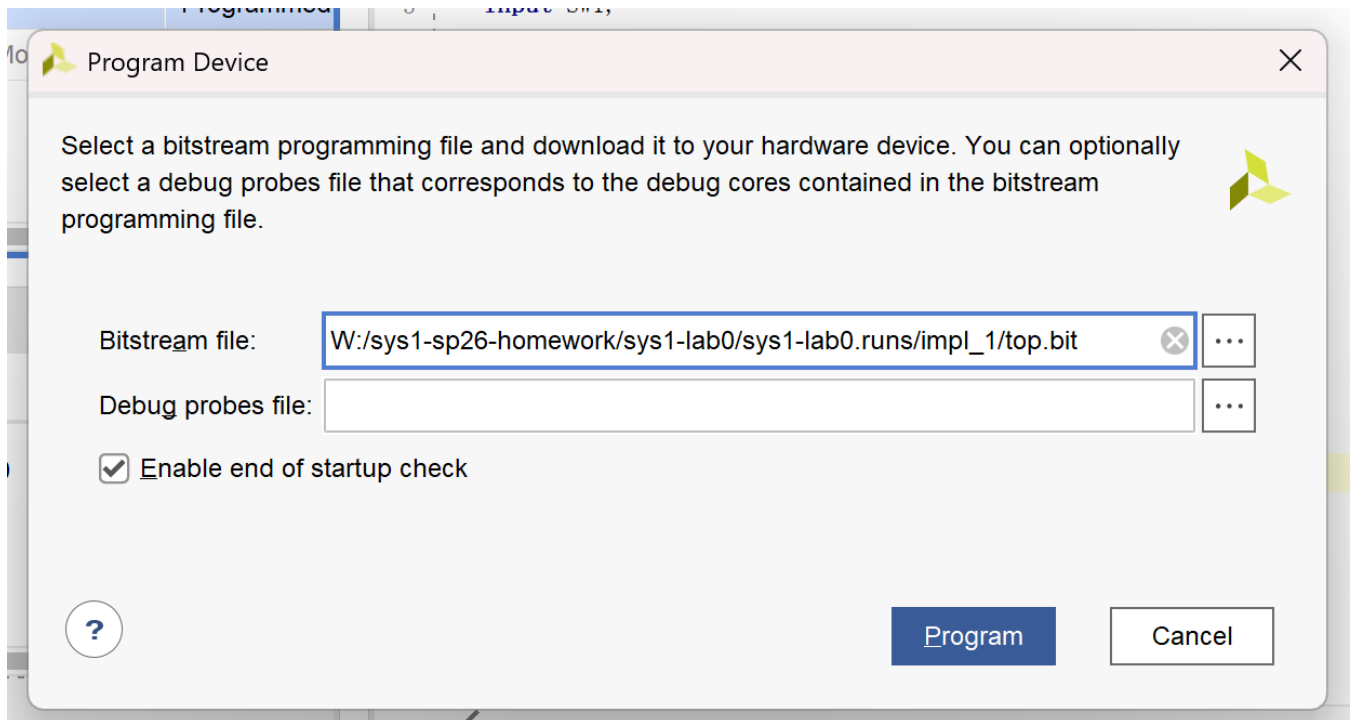


Image 9

2.6 下板验证:

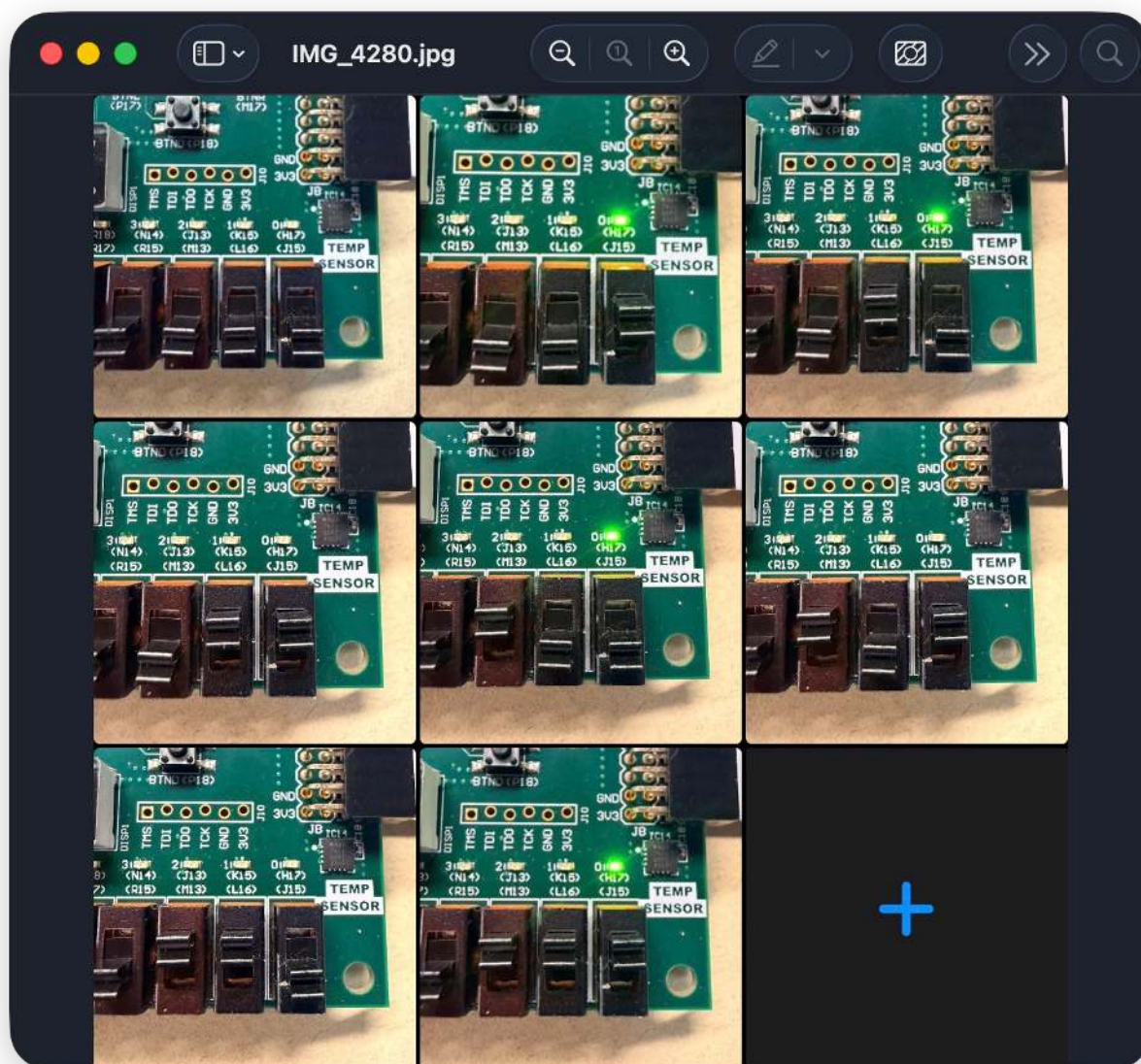


Image 10

2.7 vivado仿真：

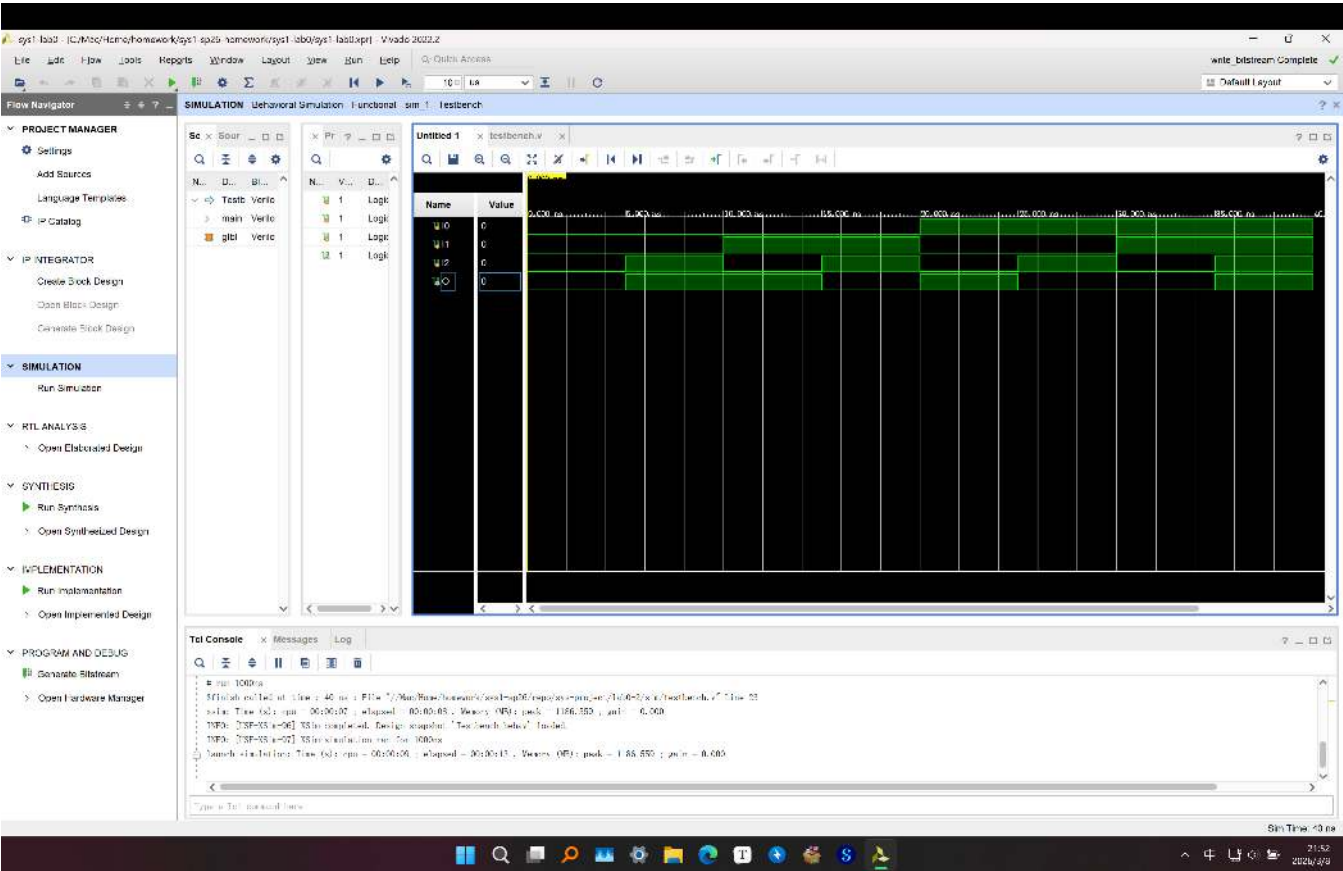


Image 11

由于对各种环境进行尝试，我客观上完成了很多遍构建等下板流程，初步熟悉了软件和命令，这未尝不是一件好事（误）

3. 实验报告

3.1 Verilog 练习 30%

- 写出如下表达式对应的 Verilog 代码（补全 src/lab0-3/syn 10%、src/lab0-3/submit 10%）
- 编写引脚约束文件并进行上板验证，其中将开关 R15、M13、L16、J15 作为输入，LED 灯 H17 作为输出（给出上板结果 10%）：

$$F(A,B,C,D) = \overline{B}\overline{C}\overline{D} + \overline{A}C\overline{B} + \overline{A}BD + CD + AC$$

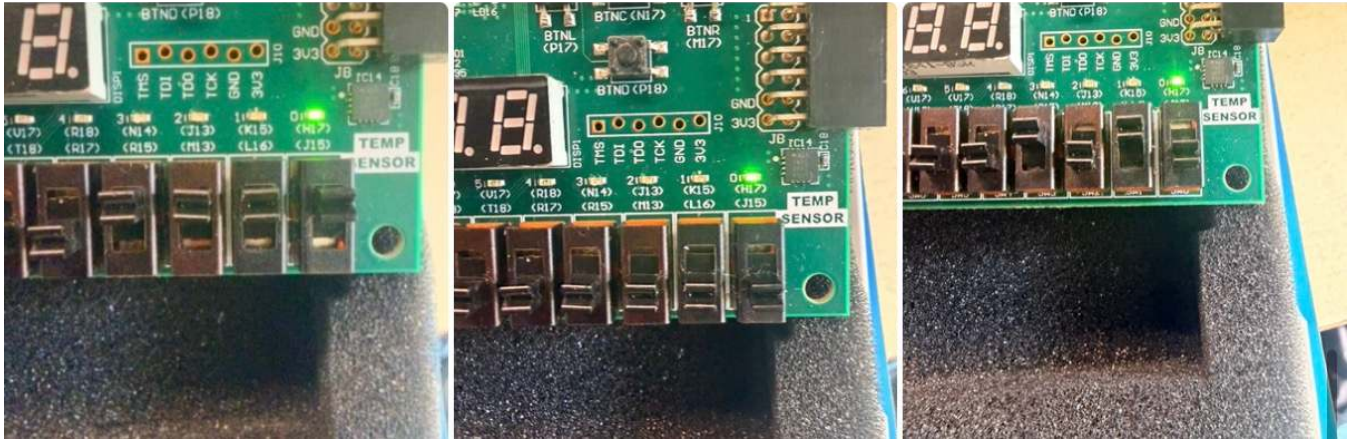


Image 12

3.2 仿真练习 20%

- 为该模块设计仿真激励文件（补全 src/lab0-3/sim 10%），分别用 Verilator 和 Vivado 进行仿真，解释仿真设计的思路，并提供仿真的波形截图，两部分思路截图各5%，共10%

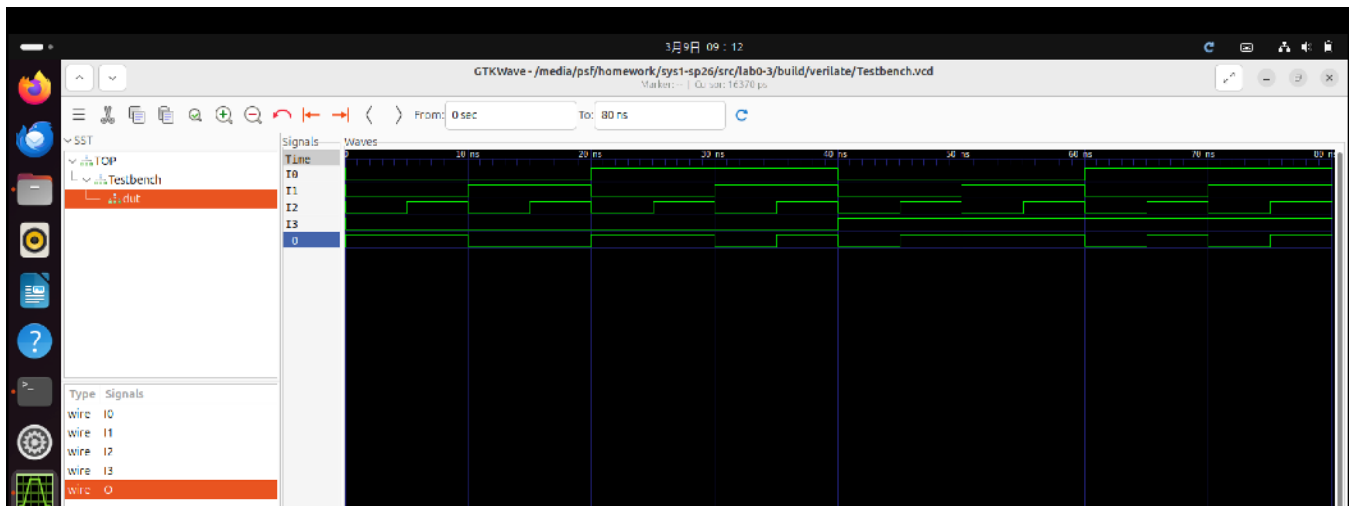


Image 13

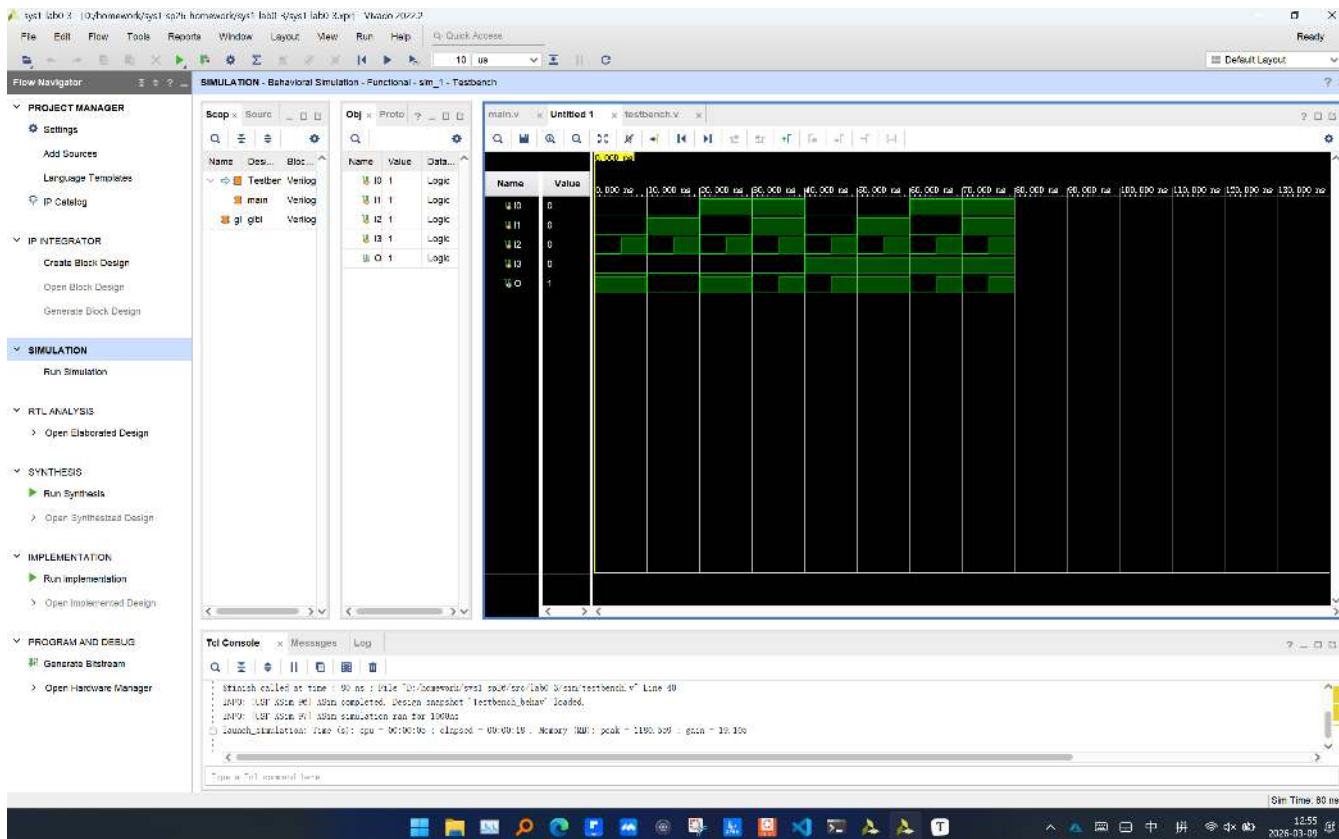


Image 14

- 尝试 Verilog、Vivado 波形窗口的各个按键，尝试理解每个按键的功能

4. 验收和代码提交 60%

4.1 验收检查点

- 仿真代码解释和仿真波形展示 20%
- 代码解释 20%
- 下板验证 20%

组合数比较多，给出了其中几个图片

4.2 提交文件

- 5.1 练习编写的源代码文件和引脚约束文件
- 5.2 练习编写的仿真文件

已提交

5. 心得体会

- 配环境和电脑是个折腾的工作，不过我很感兴趣！

- Verilog刚上手感觉不简单。
- 电路图的实现跟我之前玩的一款游戏（《Turing Complete》）有点像hhh

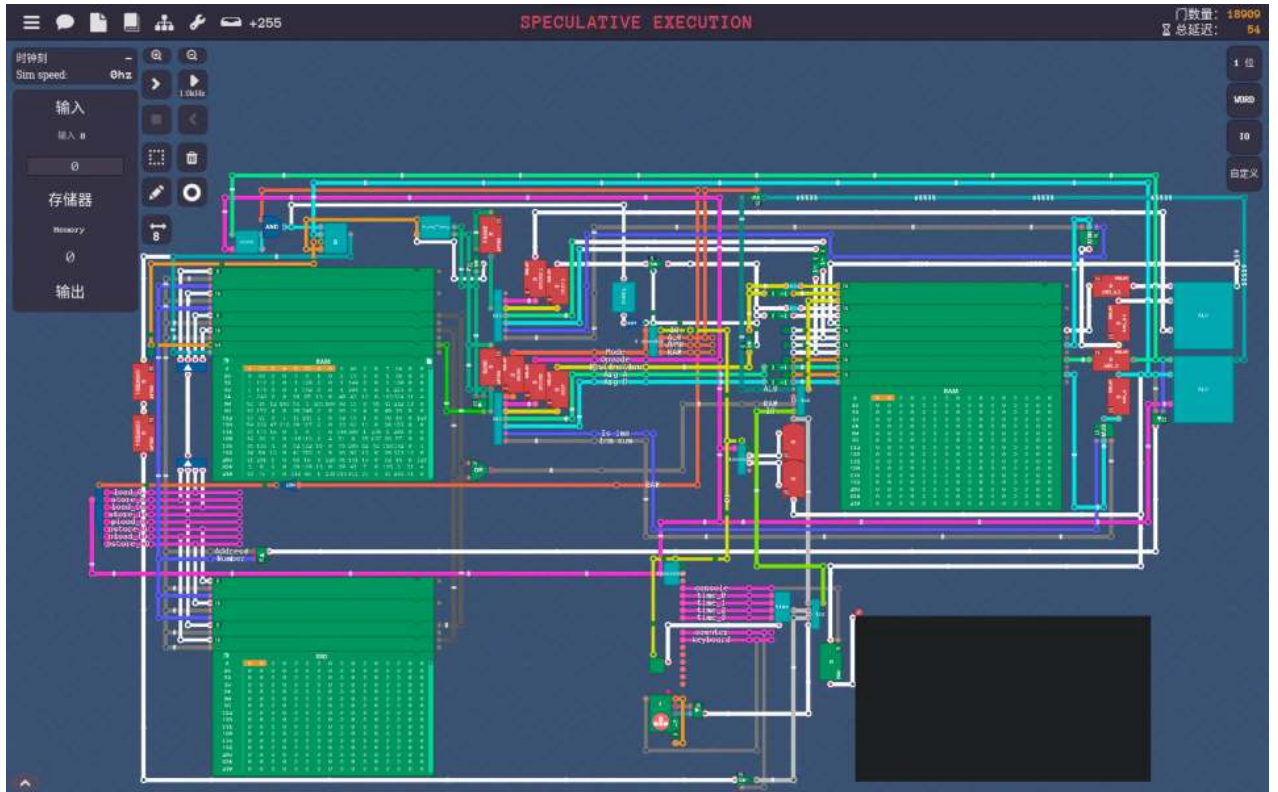


Image 15