

lab0 魏岚

Lab0-1

实验目的

- 了解基本元器件电压特性
- 掌握用于设计和仿真数字逻辑电路的 Logisim 的基本使用方法
- 理解硬件语言在电路级别的映射关系

实验过程

设计思路，实现步骤

- **思路：**我在lab0-1配置环境的基础上，按照[syslab指导](#)进行了基础的环境配置（反相器，logisim），并且绘制了相应的电路图。

debug

- Q1：如何通过linux bash语言进入所要指定进入的文件目录 path：

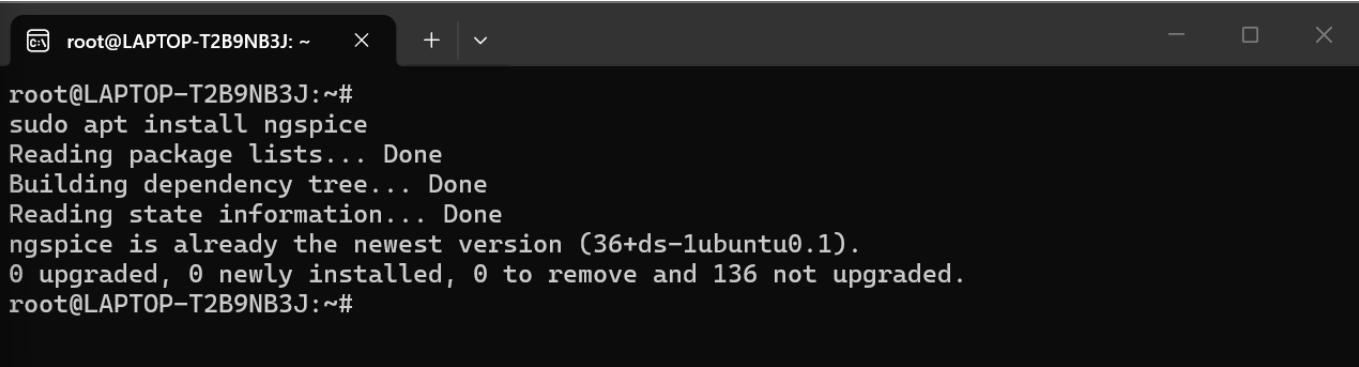
```
pwd                # 显示当前路径
ls -la             # 列出所有文件（包括隐藏文件）
cd /path/to/dir    # 切换目录
cd ~ 或 cd         # 返回家目录
cd -               # 返回上一个目录
```

实验结果

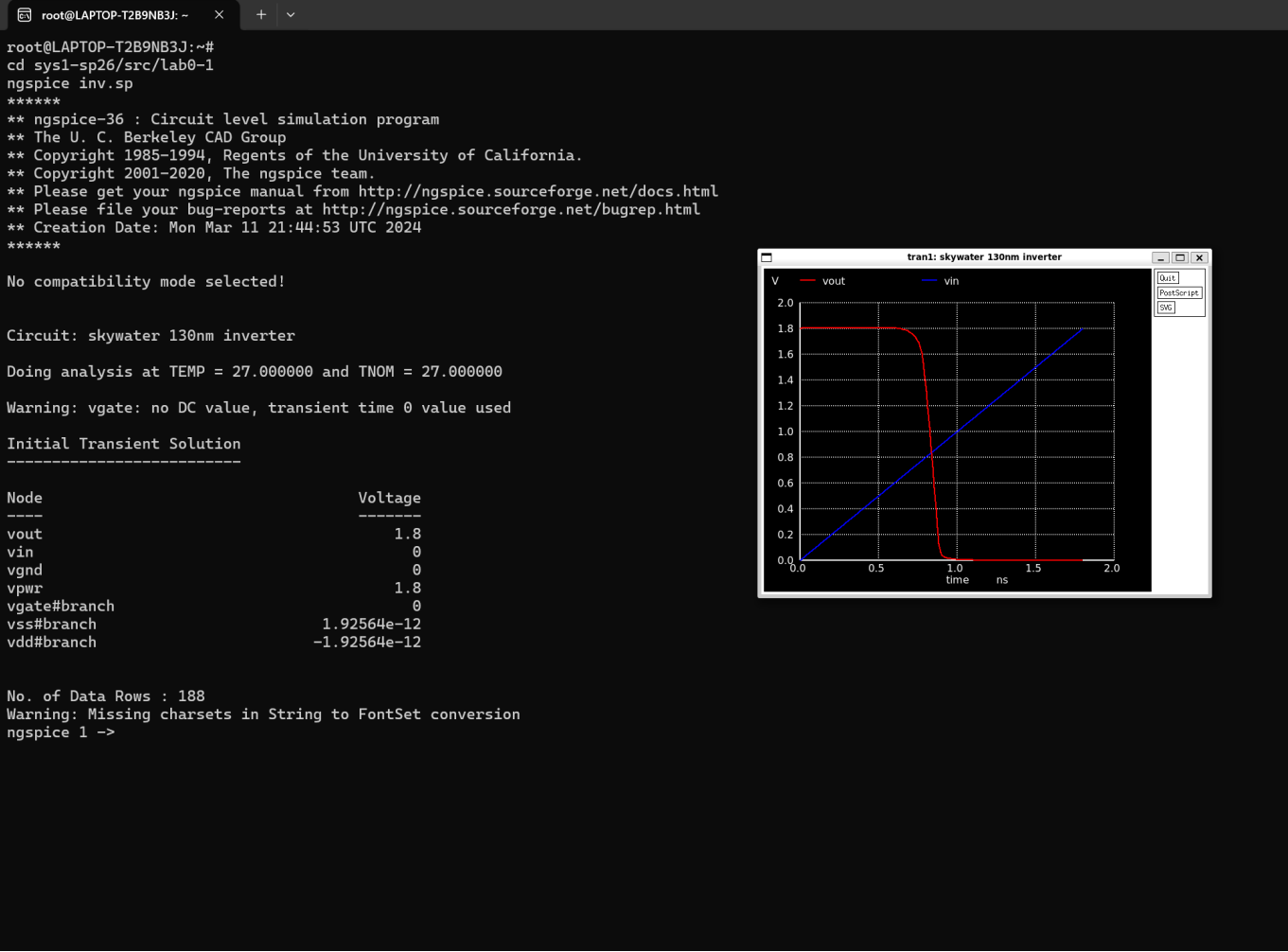
反相器的电压传输特性

截图除安装外的相关实验步骤

ngspice



ngspice仿真

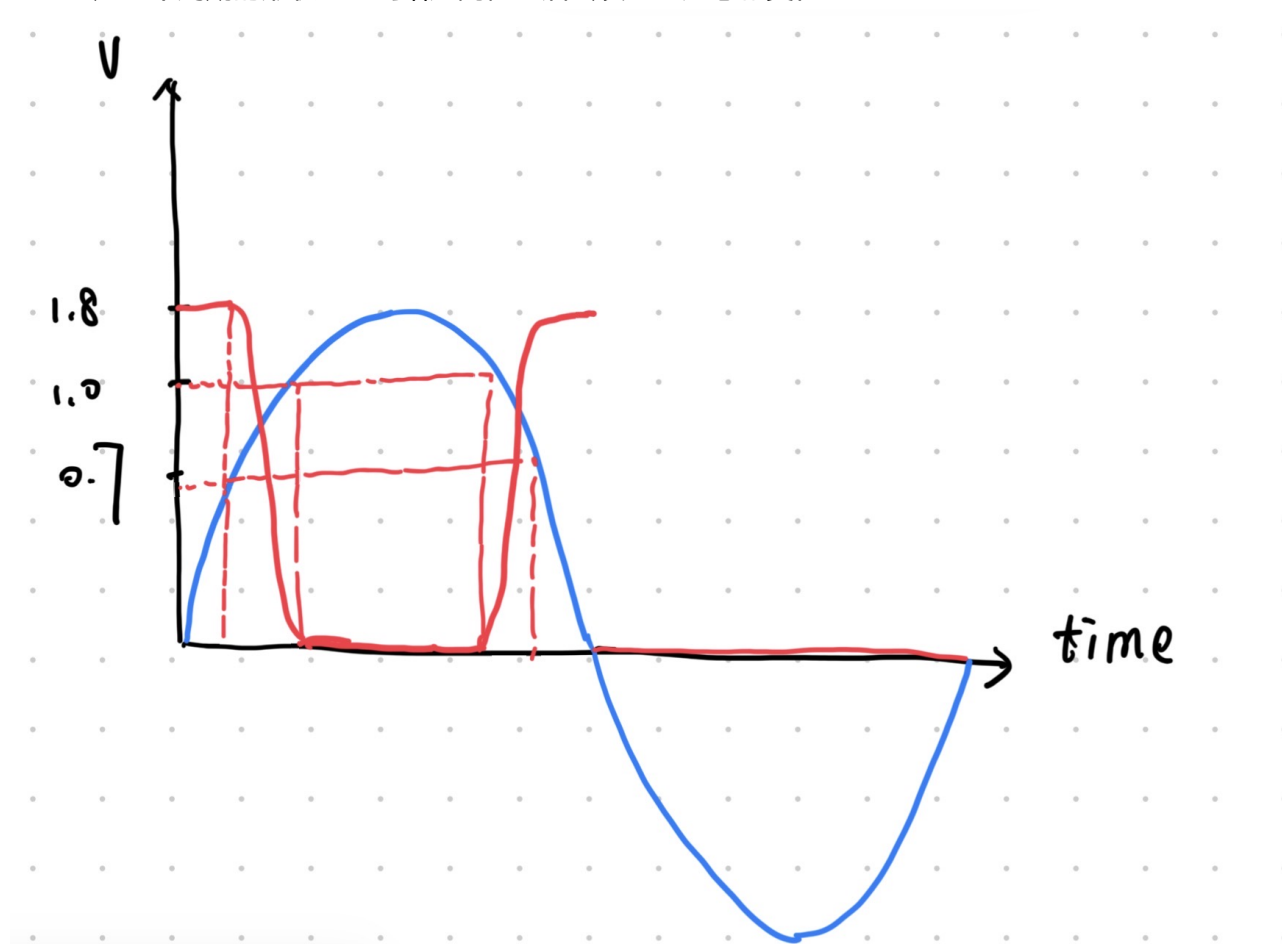


描述执行ngspice inv.sp得到的输入输出关系

- vin是以线性输入的
- vout在vin0.0-0.7时处于高电平1.8v，在vin0.7-1.0v时，vout以平滑且陡峭的路线达到0v低电平。之后vout全是0v

观察 Vout，思考如果给 Vin 接入一个正弦变化的电压，输出的 Vout 是什么样的，可以手绘 Vout-T 的结果加以文字描述

- v_{out} 也是一个周期的形状。达到给定阈值之后会发生电压急剧变化。



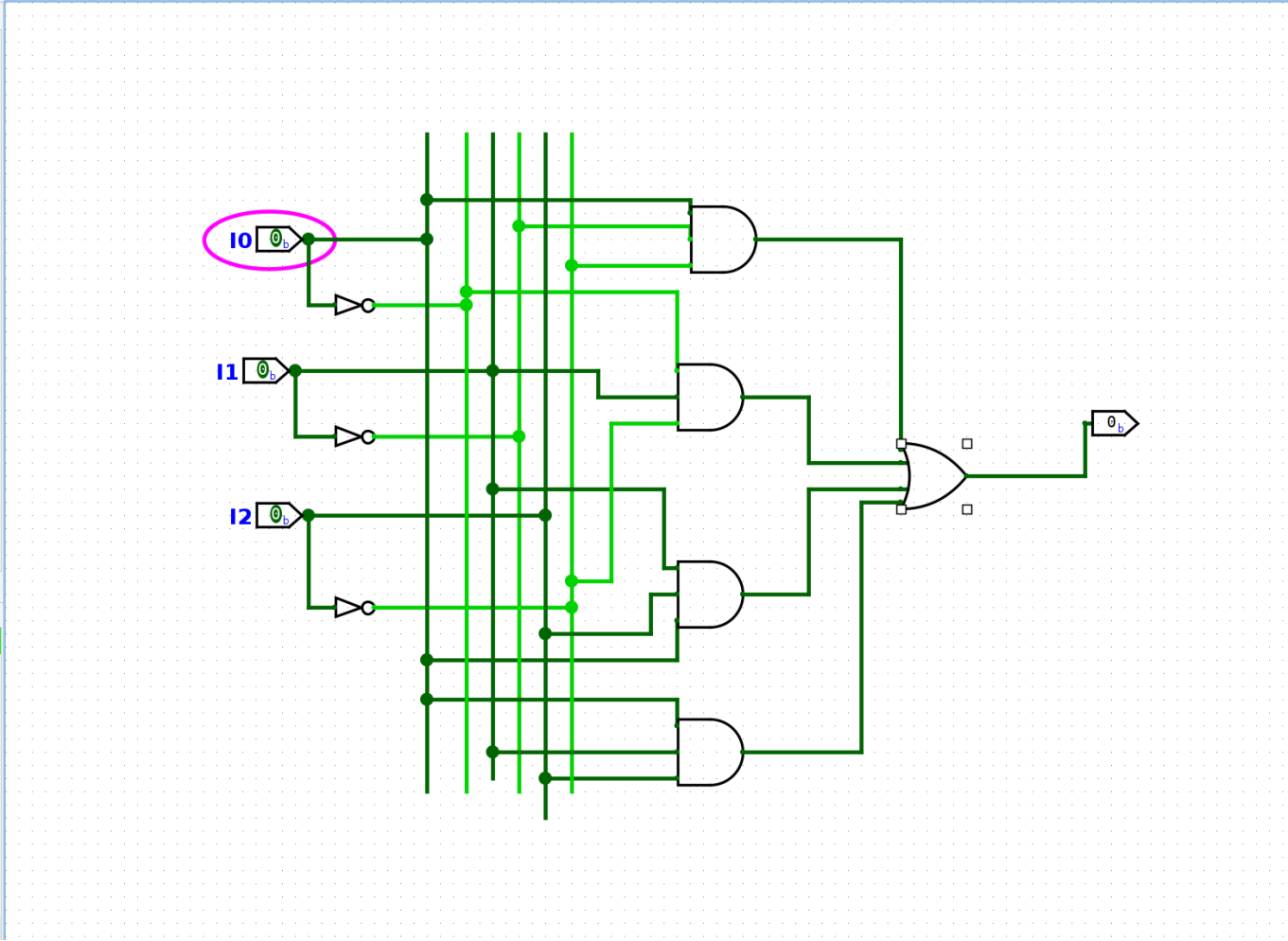
你认为对于该反相器，逻辑 0 和逻辑 1 的阈值应该设定为多少 V，即 V_{in} 输入的电压范围是多少可以认为输入的电压是 0/1 信号， V_{out} 输出的电压范围是多少可以认为输出的电压是 0/1 信号

- 我认为 0, 1 的阈值应该设定为 0 和 1.8V。其中 v_{in} 在小于 0.7 时应当为 0，大于 1.0 时为 1。 v_{out} 在小于 0.1V 时认为是 0，在大于 1.6V 时认为是 1

Logisim 电路仿真

截图除安装、保存外的相关实验步骤

logisim电路



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

I0	I1	I2	O
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

思考题

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

- 我认为在逻辑关系的运算之中，与门之间要使用逻辑乘法，而或门之间要使用逻辑加法。等式的左右两边分别是输入的逻辑运算，右边是输出的值。

心得体会

- 收获丰富，但有一个小小的建议，就是TA在实验课上可以使用话筒吗，坐在旁边会有杂音。

lab0-2 lab0-3

实验过程记录的是lab0-2，实验结果记录的是lab0-3的verilog实践编写。

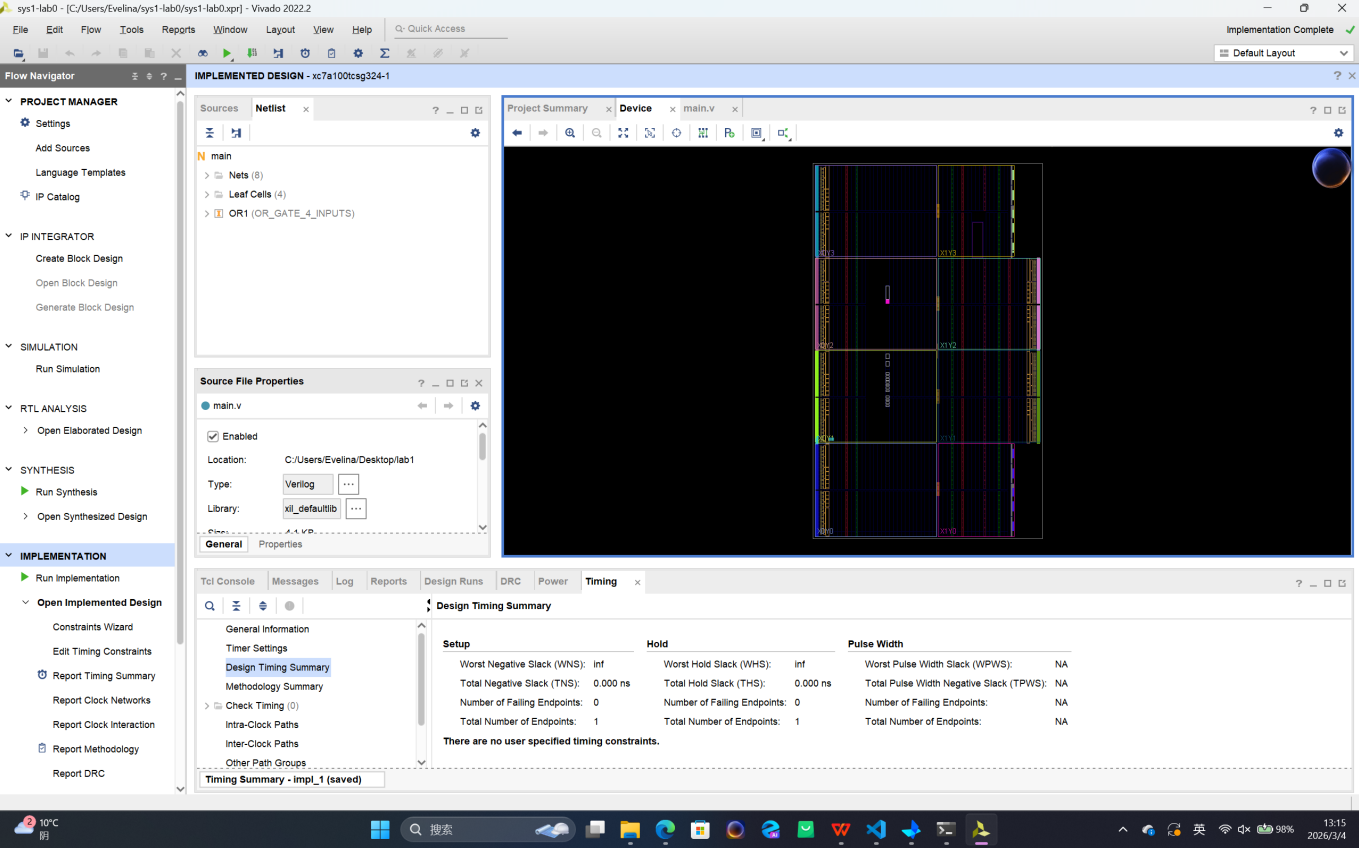
实验目的

- 掌握 FPGA 设计软件 Vivado 的基本使用方法
- 掌握 FPGA 和 Verilog 语言的最基础使用方法
- 学习布尔表达式和对应 Verilog 语言的转换

实验过程(仅仅是lab0-2的过程)

将原理图导出为 Verilog

vivado成功复制完内容，跑出电路图

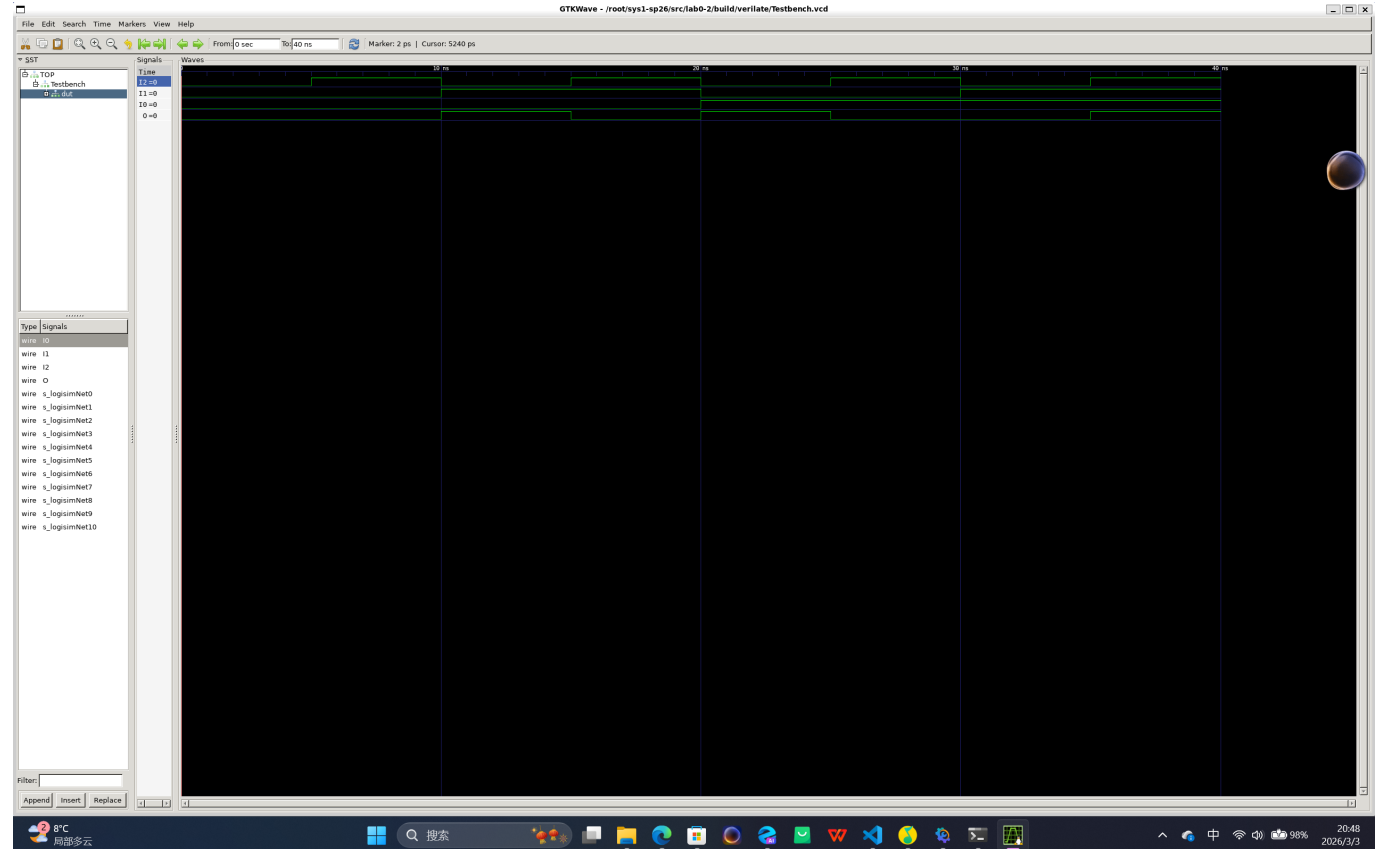


使用tree查看src文件

```
root@LAPTOP-T2B9NB3J:~/sys1-sp26/repo# tree
.
├── Makefile
├── opensbi
├── patch
│   ├── opensbi
│   │   └── 1.patch
│   └── riscv-isa-cosim
│       ├── 1.patch
│       └── 2.patch
├── riscv-isa-cosim
├── riscv-openocd
├── riscv-pk
├── sys-project
│   ├── lab0-2
│   │   ├── sim
│   │   │   └── testbench.v
│   │   ├── syn
│   │   │   ├── nexysa7.xdc
│   │   │   └── top.v
│   │   └── tcl
│   │       └── vivado.tcl
```

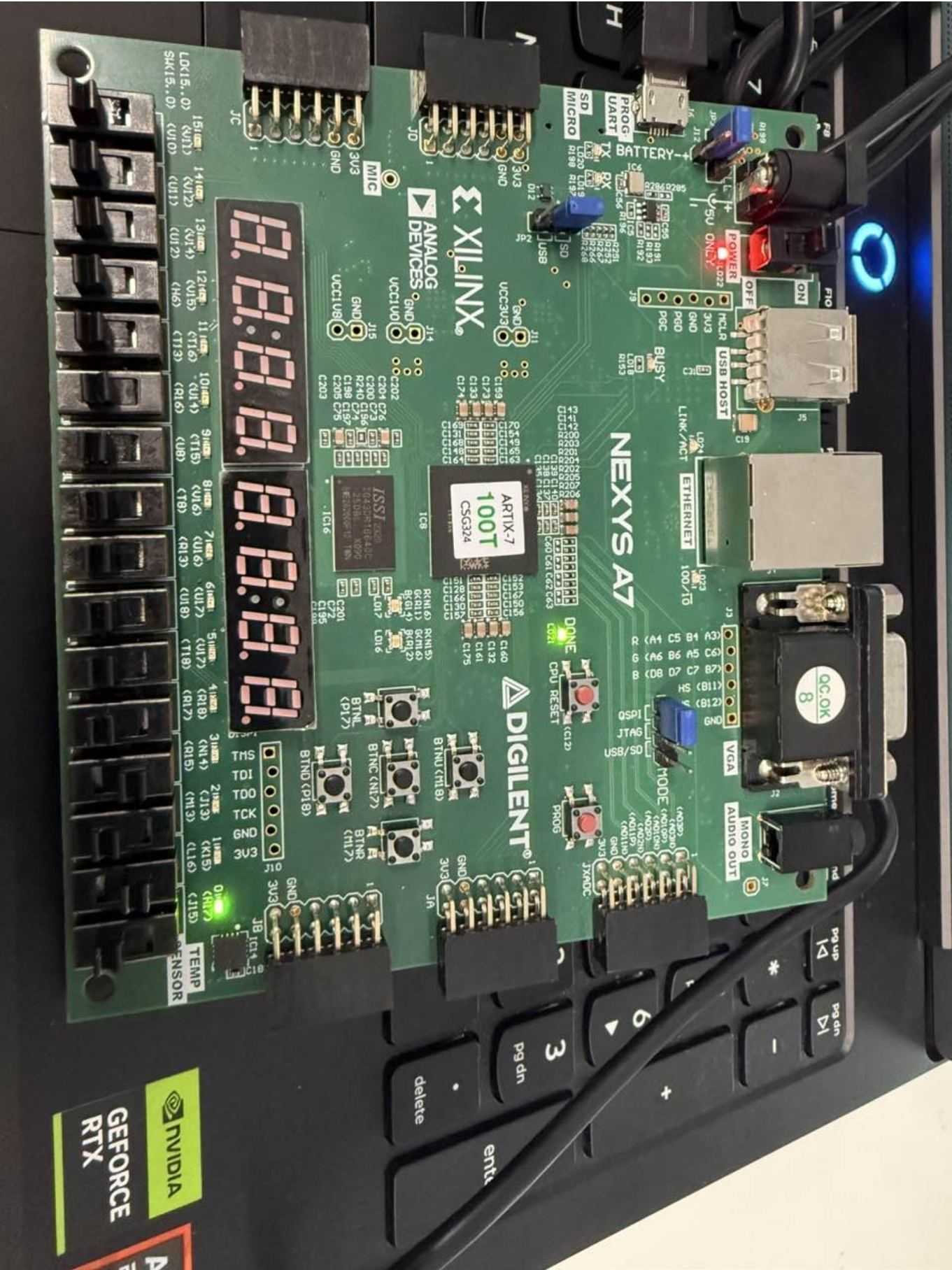
Verilator 仿真测试

使用gtkwave查看波形图



Vivado下板

Vivado 下板



过程中所遇到的bug

- Q1:网络连接的问题;

```
fatal: unable to access 'https://github.com/verilator/verilator.git/': GnuTLS recv error (-110): The TLS connection was non-properly terminated.
fatal: clone of 'https://github.com/verilator/verilator.git' into submodule path '/root/sys1-sp26/repo/verilator' failed
Failed to clone 'repo/verilator'. Retry scheduled
Cloning into '/root/sys1-sp26/repo/verilator'...

fatal: unable to access 'https://github.com/verilator/verilator.git/': Failed to connect to github.com port 443 after 134356 ms: Connection timed out
fatal: clone of 'https://github.com/verilator/verilator.git' into submodule path '/root/sys1-sp26/repo/verilator' failed
Failed to clone 'repo/verilator' a second time, aborting
```

解决方法: 上网寻找科学合法的vpn。设置为局域网连接, 再将wsl的网络连接配置更改到本机所在的ip地址, 再通过这个vpn连接到境外的git库。

- Q2: 在wsl中使用makewave语句之后无法对lab中的v文件进行了连接编译。

```
verilator -Wno-STMTDLY --timescale 1ns/10ps --trace --cc --exe --main --timing --Mdir /root/sys1-sp26/src/lab0-2/build/verilate --top-module Testbench -D
DEBUG -DTOP=Testbench -std=c++17 /root/sys1-sp26/src/lab0-2/../../repo/sys-project/lab0-2/sim/testbench.v /root/sys1-sp26/src/lab0-2/submit/OR_GATE_4_IN
c/lab0-2/submit/main.v /root/sys1-sp26/src/lab0-2/submit/AND_GATE_3_INPUTS.v +define+TOP_DIR="/root/sys1-sp26/src/lab0-2/build/verilate/" +define+VERILAT
%Warning-PINMISSING: /root/sys1-sp26/src/lab0-2/../../repo/sys-project/lab0-2/sim/testbench.v:26:10: Cell has missing pin: 'o'
26 | main dut(
    | ^
    | ... For warning description see https://verilator.org/warn/PINMISSING?v=5.021
    | ... Use "/* verilator lint_off PINMISSING */" and lint_on around source to disable this message.
%Error-PINNOTFOUND: /root/sys1-sp26/src/lab0-2/../../repo/sys-project/lab0-2/sim/testbench.v:30:10: Pin not found: 'O'
30 | .O(O)
    | ^
%Error: Exiting due to 1 error(s), 1 warning(s)
make: *** [Makefile:28: Testbench] Error 1
root@LAPTOP-T2B9NB3J:~/sys1-sp26/src/lab0-2# make wave
gtkwave /root/sys1-sp26/src/lab0-2/build/verilate/Testbench.vcd

GTKWave Analyzer v3.3.104 (w)1999-2020 BSI

(gtkwave:19024): dconf-WARNING **: 20:28:58.840: failed to commit changes to dconf: Could not connect: No such file or directory
Error opening .vcd file '/root/sys1-sp26/src/lab0-2/build/verilate/Testbench.vcd'.
Why: No such file or directory
make: *** [Makefile:32: wave] Error 255
root@LAPTOP-T2B9NB3J:~/sys1-sp26/src/lab0-2#
```

发现我的main.v文件中的输入和输出的文件名称和top.v中的端口名称不一致, 所以无法在这两个文件之间产生连接。

- Q3: wsl连接vivado的时候会产生连接错误。

```
root@LAPTOP-T2B9NB3J:~/sys1-sp26/src/lab0-2# make bitstream
mkdir -p /root/sys1-sp26/src/lab0-2/build/project
cd /root/sys1-sp26/src/lab0-2/build/project; cp /root/sys1-sp26/src/lab0-2/../../repo/sys-project/lab0-2/tcl/vivado.tcl .; cmd.exe /c "call D:/Vivado/2022.2/settings64.bat && set
DIR_SRC 65 \
vivado -mode batch -nojournal -source vivado.tcl -tclargs -top-module top -board xc7a100tcs9324-1"
用作当前目录的以上路径启动了 CMD.EXE.
UNC 路径不受支持。默认值设为 Windows 目录。
DIR_SRC=\\wsl.localhost\Ubuntu-22.04\root\sys1-sp26\src\lab0-2\submit
CRITICAL WARNING: [Common 17-183] Failed to open handle vivado.log. Please check access permission of directory 'C:\Windows'. You should restart the application from a writable wo
rking directory.

***** Vivado v2022.2 (64-bit)
**** SW Build 3671981 on Fri Oct 14 05:00:03 MDT 2022
**** IP Build 3669848 on Fri Oct 14 08:30:02 MDT 2022
** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

source vivado.tcl
couldn't read file "vivado.tcl": no such file or directory
INFO: [Common 17-206] Exiting Vivado at Wed Mar 4 10:36:25 2026...
make: *** [Makefile:62: bitstream] Error 1
```

原因: 在wsl中的内存是和windows里面互相独立的, 在wsl中无法连接到下载在windows的vivado。

实验结果 Verilog 练习

补全 src/lab0-3/syn 10%、src/lab0-3/submit 10%

代码如下:

top.v

```
module top(
    input SW0,
    input SW1,
    input SW2,
    input SW3,
    output LD0
);
main Idut(
    .I0(SW0),
    .I1(SW1),
    .I2(SW2),
```

```
        .I3(SW3),  
        .O(LDO)  
    );  
  
endmodule
```

main.v

```
module main(  
    input I0,  
    input I1,  
    input I2,  
    input I3,  
    output O  
);  
  
    wire net1;  
    wire net2;  
    wire net3;  
    wire net4;  
    wire net5;  
    wire net6;  
    wire net7;  
    wire net8;  
    wire net9;  
    wire net10;  
    wire net11;  
    wire net12;  
    wire net13;  
    wire netoutput1;  
  
    assign O=netoutput1;  
    assign net1=I0;  
    assign net3=I1;  
    assign net5=I2;  
    assign net7=I3;  
    assign net2=~net1;  
    assign net4=~net3;  
    assign net6=~net5;  
    assign net8=~net7;  
  
    AND_GATE_3_INPUTS #(BubblesMask(3'b000))  
        AND1(.input1(net4),  
            .input2(net6),  
            .input3(net8),  
            .result(net9));  
  
    AND_GATE_3_INPUTS #(BubblesMask(3'b000))  
        AND2(.input1(net2),  
            .input2(net5),  
            .input3(net4),  
            .result(net10));
```

```
AND_GATE_3_INPUTS #(.BubblesMask(3'b000))
  AND3(.input1(net2),
        .input2(net3),
        .input3(net7),
        .result(net11));

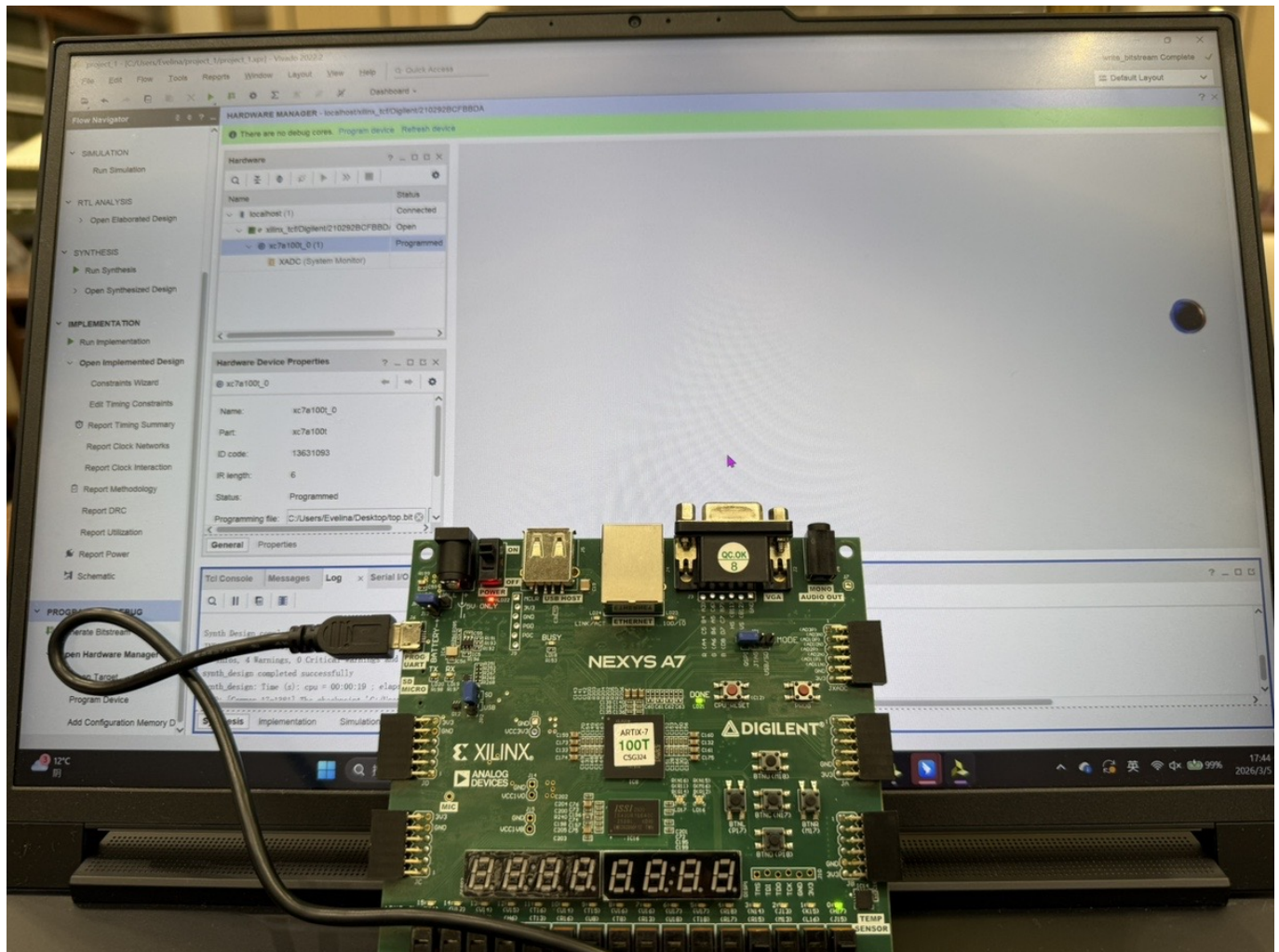
AND_GATE_2_INPUTS #(.BubblesMask(2'b00))
  AND4(.input1(net5),
        .input2(net7),
        .result(net12));
AND_GATE_2_INPUTS #(.BubblesMask(2'b00))
  AND5(.input1(net1),
        .input2(net5),
        .result(net13));

OR_GATE_5_INPUTS #(.BubblesMask(5'b00000))
  OR1 (.input1(net9),
        .input2(net10),
        .input3(net11),
        .input4(net12),
        .input5(net13),
        .result(netoutput1));

endmodule
```

上版验证结果

成功!



仿真练习

激励文件

```
module Testbench;

    reg I0,I1,I2,I3;
    wire O;

    initial begin
        $display("Time\t I0 I1 I2 I3 | O");
        $display("-----");
        I0 = 0; I1 = 0; I2 = 0; I3 = 0; #10;
        I0 = 0; I1 = 0; I2 = 0; I3 = 1; #10;
        I0 = 0; I1 = 0; I2 = 1; I3 = 0; #10;
        I0 = 0; I1 = 0; I2 = 1; I3 = 1; #10;
        I0 = 0; I1 = 1; I2 = 0; I3 = 0; #10;
        I0 = 0; I1 = 1; I2 = 0; I3 = 1; #10;
        I0 = 0; I1 = 1; I2 = 1; I3 = 0; #10;
        I0 = 0; I1 = 1; I2 = 1; I3 = 1; #10;
        I0 = 1; I1 = 0; I2 = 0; I3 = 0; #10;
        I0 = 1; I1 = 0; I2 = 0; I3 = 1; #10;
        I0 = 1; I1 = 0; I2 = 1; I3 = 0; #10;
```

```
I0 = 1; I1 = 0; I2 = 1; I3 = 1; #10;
I0 = 1; I1 = 1; I2 = 0; I3 = 0; #10;
I0 = 1; I1 = 1; I2 = 0; I3 = 1; #10;
I0 = 1; I1 = 1; I2 = 1; I3 = 0; #10;
I0 = 1; I1 = 1; I2 = 1; I3 = 1; #10;

$display("测试完成");
$finish;

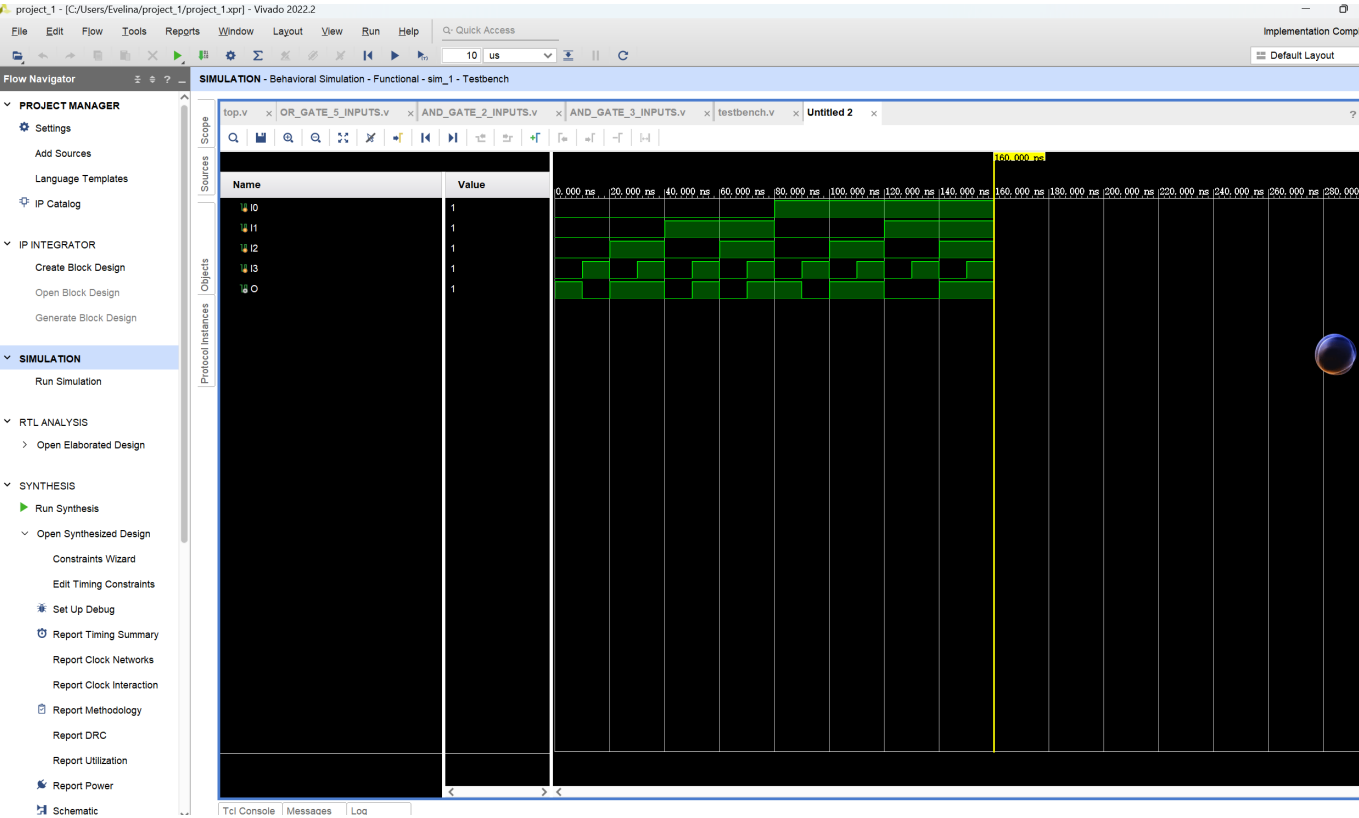
end

main dut(
    .I0(I0),
    .I1(I1),
    .I2(I2),
    .I3(I3),
    .O(O)
);

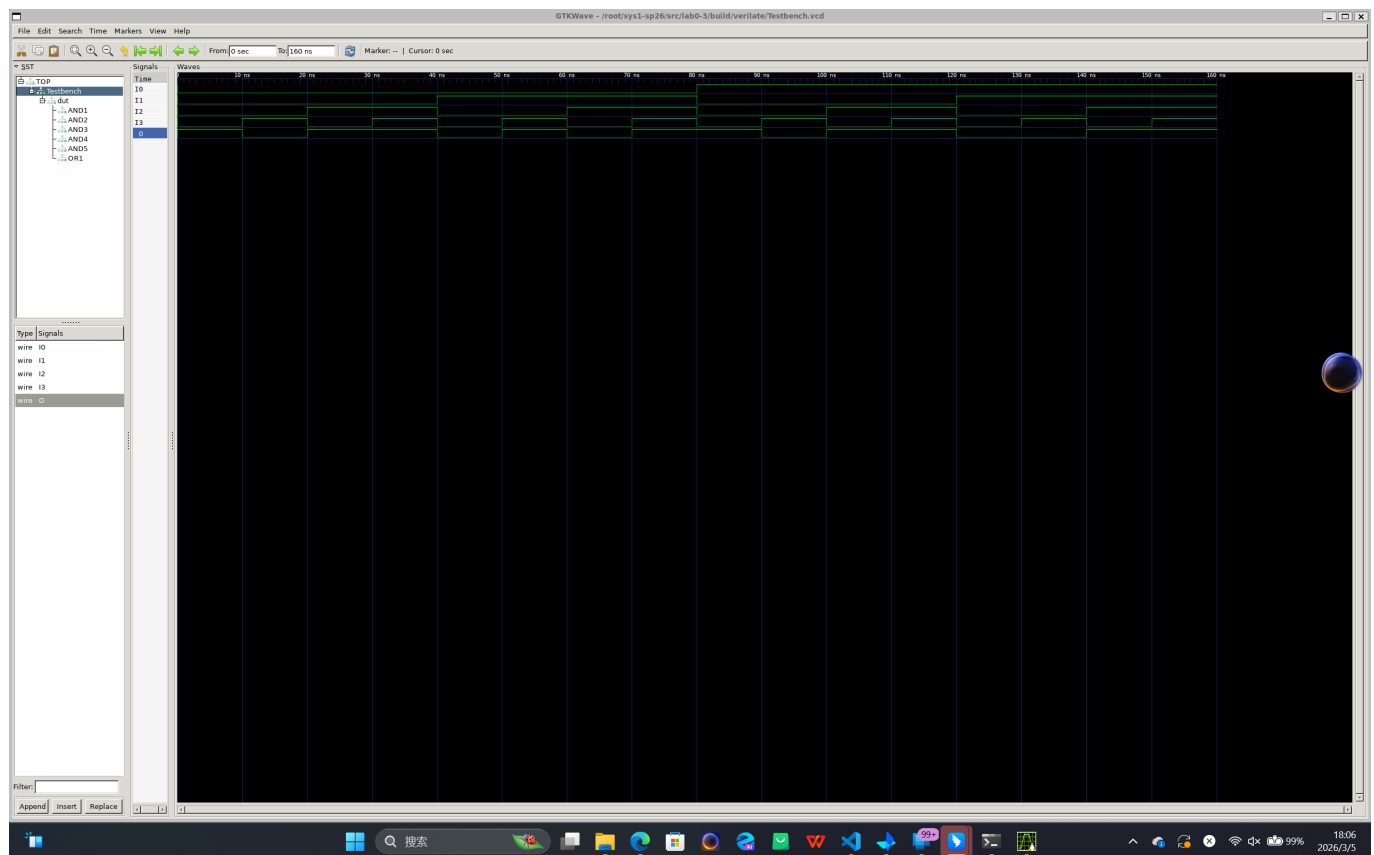
`ifdef VERILATE
    initial begin
        $dumpfile({`TOP_DIR,"/Testbench.vcd"});
        $dumpvars(0,dut);
        $dumpon;
    end
`endif

endmodule
```

vivado仿真



verilator仿真



思路

- 我采用的是ipad上手绘出大致的电路图，然后对照着手绘图进行编程。我先对其进行了top.v的补全，目的是从上层到下层进行编写。再编写main.v，main.v先编写了input和线的连接以及加了一个非门之后和线的连接。再加入与门，或门，——assign。思路也是从想top.v那样思考再细化main.v。

心得体会

- 我还是要多多思考，多多试错，最后的lab就可以做的出来！！