浙江大学

本科实验报告

课程名称：计算机逻辑设计基础

姓　　名：仇国智

学　　院：竺可桢学院

专　　业：计算机科学与技术

学　　号：3220102181

指导教师：董亚波

2023 年 10 月 16 日

一、 实验目的和要求

1　目的

1.1　掌握变量译码器的的逻辑构成和逻辑功能

1.2　用变量译码器实现组合函数

1.3　采用原理图设计电路模块

1.4　进一步熟悉 ISE 平台及下载实验平台物理验证

2　要求

2.1　原理图设计实现 74LS138 译码器模块

2.2　用 74LS138 译码器实现楼道灯控制，具体功能和操作同实验四任务 1

二、 实验内容和原理

1　内容

1.1　任务 1：原理图设计实现 74LS138 译码器模块

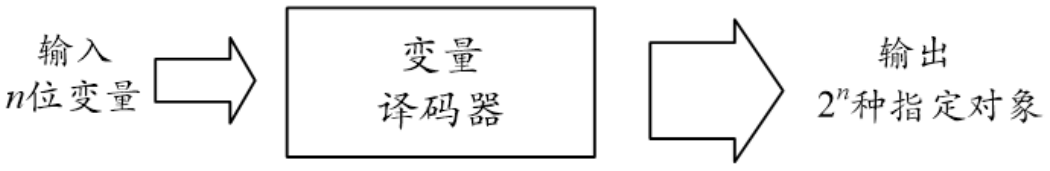
1.2　任务 2：用 74LS138 译码器实现楼道灯控制器

2　原理

2.1　译码器是将一种输入编码转换成另一种编码的电路，即将给定的代码进行“翻译”并转换成指定的状态或输出信号（脉冲或电平）。

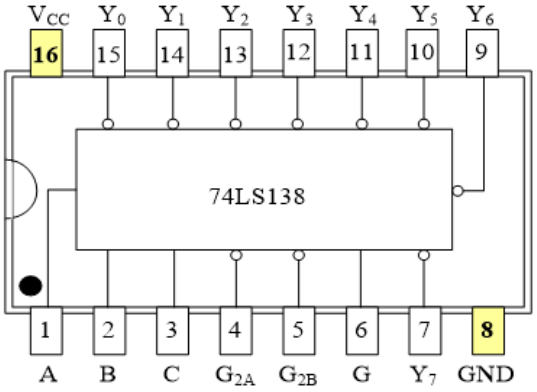
2.2　译码可分为：变量译码、显示译码。变量译码一般是将一种较少位输入变为较多位输出的器件，如 2n 译码和 8421BCD 码译码。显示译码主要进行 2 进制数显示成 10 进制或 16 进制数的转换，可分为驱动 LED 和 LCD 两类。

2.3　变量译码器是一个将 n 个输入变为 2n 个最小项输出的多输出端的组合逻辑电路。n 通常在 2~64 之间。



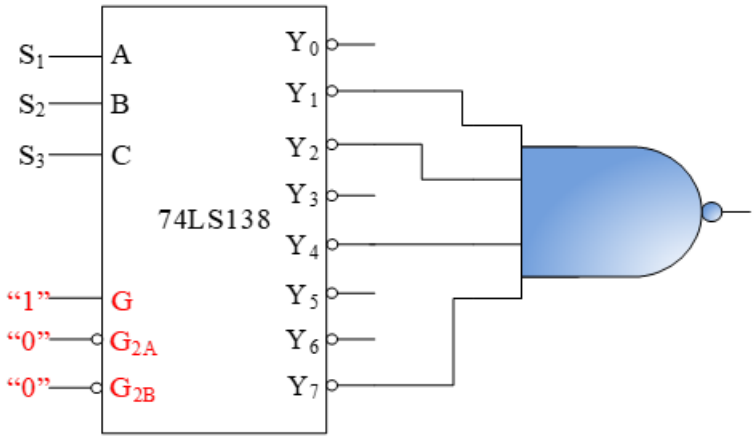
2.4 变量译码器—74LS138：带 3 个使能端的 3-8 译码器的逻辑结构由三级门电路构成，输出低电平有效。

输入		译码器输出（低电平有效）							
使能	变量								
$GG_{2A}G_{2B}$	ABC	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
×11	×××	1	1	1	1	1	1	1	1
0××	×××	1	1	1	1	1	1	1	1
100	000	0	1	1	1	1	1	1	1
100	001	1	0	1	1	1	1	1	1
100	010	1	1	0	1	1	1	1	1
100	011	1	1	1	0	1	1	1	1
100	100	1	1	1	1	0	1	1	1
100	101	1	1	1	1	1	0	1	1
100	110	1	1	1	1	1	1	0	1
100	111	1	1	1	1	1	1	1	0



2.5 变量译码器的输出对应所有输入变量的最小项组合，如果将函数转换成最小项和的形式，则可以用变量译码器实现函数的组合电路:

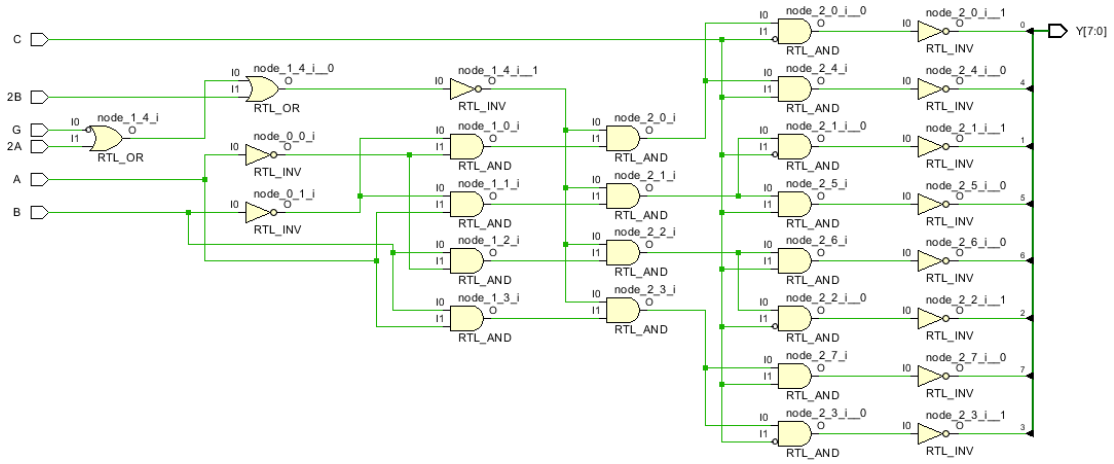
$$F = \overline{S3}\,\overline{S2}S1+\overline{S3}S2\overline{S1}+S3\overline{S2}\,\overline{S1}+S3S2S1$$



三、 实验过程和数据记录

1 原理图设计实现 74LS138 译码器模块

1.1 新建工程，工程名称用 D_74LS138_SCH。新建 Schematic 源文件，文件名称用 D_74LS138。原理图方式进行设计。



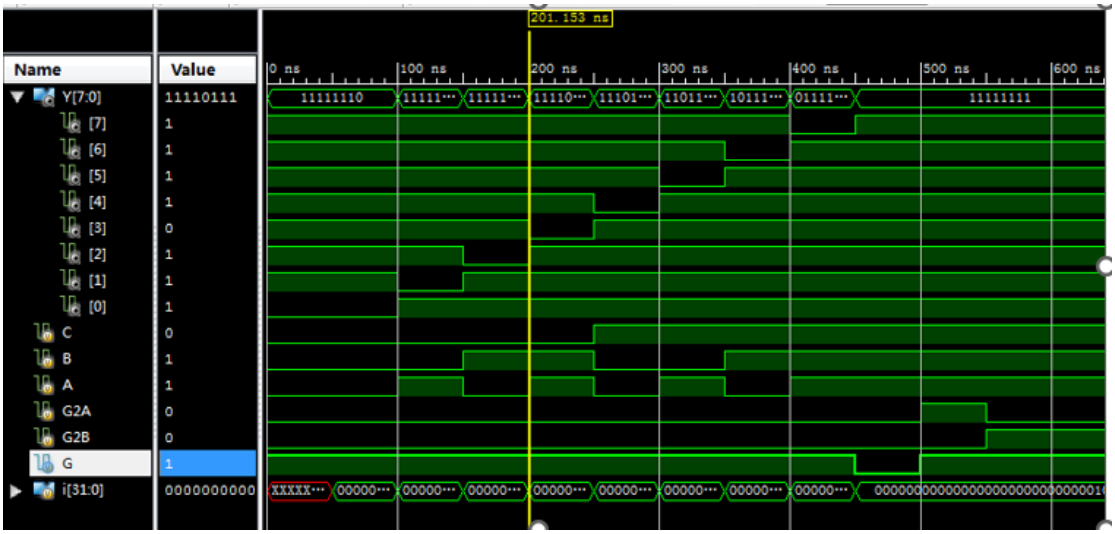

```
`timescale 1ns / 1ps
module decoder_3_8 (
    A,
    B,
    C,
    G,
    G2A,
    G2B,
    Y
);
  input wire A, B, C, G, G2A, G2B;
  output wire [7:0] Y;
  not node_0_0 (A_n, A), node_0_1 (B_n, B), node_0_2 (C_n, C), node_0_3
(G_n, G);
  and node_1_0 (D0, B_n, A_n), node_1_1 (D1, B_n, A), node_1_2 (D2, B,
A_n), node_1_3 (D3, B, A);
  nor node_1_4 (EN, G_n, G2A, G2B);
  nand node_2_0 (
      Y[0], EN, D0, C_n
  ), node_2_1 (
      Y[1], EN, D1, C_n
  ), node_2_2 (
      Y[2], EN, D2, C_n
  ), node_2_3 (
      Y[3], EN, D3, C_n
  ), node_2_4 (
      Y[4], EN, D0, C
  ), node_2_5 (
      Y[5], EN, D1, C
  ), node_2_6 (
      Y[6], EN, D2, C
  ), node_2_7 (
      Y[7], EN, D3, C
```

```
  );
endmodule
```

1.2 Check Design Rules，检查错误 View HDL Functional Model，查看并学习 Verilog HDL 代码。对 D_74LS138 模块进行仿真，激励代码如下：

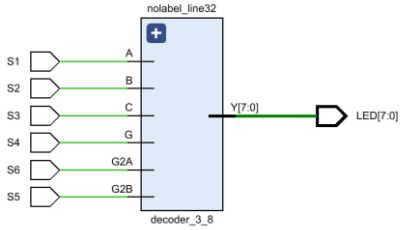
```
`timescale 1ns / 1ps
module test_tb ();
  integer i;
  reg A, B, C, G, G2A, G2B;
  wire[7:0] Y;
  decoder_3_8 UUT (
      .A  (A),
      .B  (B),
      .C  (C),
      .G  (G),
      .G2A(G2A),
      .G2B(G2B),
      .Y  (Y)
  );
  initial begin
    A   = 0;
    B   = 0;
    C   = 0;

    G   = 1;
    G2A = 0;
    G2B = 0;
    #50;

    for (i = 0; i <= 7; i = i + 1) begin
      {C,B,A} = i;
      #50;
    end
    assign G = 0;
    assign G2A = 0;
    assign G2B = 0;
    #50;

    assign G = 1;
    assign G2A = 1;
    assign G2B = 0;
```

```
    #50;

    assign G = 1;
    assign G2A = 0;
    assign G2B = 1;
    #50;
  end
endmodule
```



1.3 Create Schematic Symbol，系统生成 D_74LS138 模块的逻辑符号图文件，文件后缀.sym。符号图位于工程根目录。自动生成的符号可修改: 可以用 Tools 菜单的 Symbol Wizard，也可以打开.sym 文件直接修改。在新工程中使用时，把.sym 和.sch 复制到对应工程目录

1.4 新建工程“D_74LS138_Test”。新建 Schematic 文件“D_74LS138_Test”。复制 D_74LS138.sym 和.sch 到工程目录。在 symbols 框里的第一个元件，就是 D_74LS138。用拨盘开关控制模块的输入，用 LED(7:0)作为模块的输出，验证模块的功能。




```
`timescale 1ns / 1ps
```

```
module test(S1,
    S2,
    S3,
    S4,
    S5,
    S6,
    LED);
  input wire S1,S2,S3,S4,S5,S6;
  output wire [7:0] LED;
  decoder_3_8(S1,S2,S3,S4,S6,S5,LED);
endmodule
```

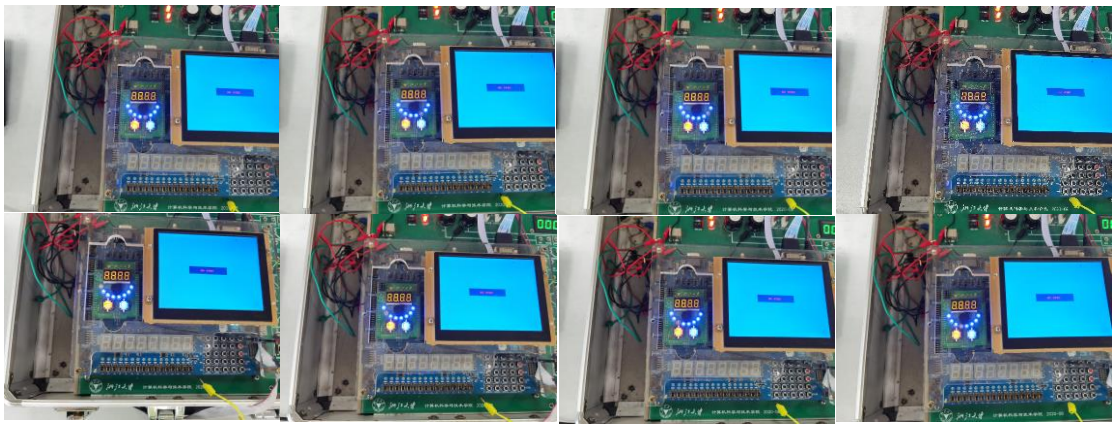
1.5 建立引脚文件，下载验证，根据真值表，操作实验板，验证功能。

```
set_property IOSTANDARD LVCMOS33 [get_ports {LED[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED[0]}]
set_property PACKAGE_PIN W23 [get_ports {LED[0]}]
set_property PACKAGE_PIN AB26 [get_ports {LED[1]}]
set_property PACKAGE_PIN Y25 [get_ports {LED[2]}]
set_property PACKAGE_PIN AA23 [get_ports {LED[3]}]
set_property PACKAGE_PIN Y23 [get_ports {LED[4]}]
set_property PACKAGE_PIN Y22 [get_ports {LED[5]}]
set_property PACKAGE_PIN AE21 [get_ports {LED[6]}]
set_property PACKAGE_PIN AF24 [get_ports {LED[7]}]
set_property IOSTANDARD LVCMOS15 [get_ports S1]
set_property IOSTANDARD LVCMOS15 [get_ports S2]
set_property IOSTANDARD LVCMOS15 [get_ports S3]
set_property PACKAGE_PIN AA10 [get_ports S1]
set_property PACKAGE_PIN AB10 [get_ports S2]
set_property PACKAGE_PIN AA13 [get_ports S3]
#set_property PACKAGE_PIN AF24 [get_ports F]
#set_property IOSTANDARD LVCMOS33 [get_ports F]
set_property IOSTANDARD LVCMOS15 [get_ports S4]
set_property IOSTANDARD LVCMOS15 [get_ports S5]
set_property IOSTANDARD LVCMOS15 [get_ports S6]
set_property PACKAGE_PIN AA12 [get_ports S4]
set_property PACKAGE_PIN Y13 [get_ports S5]
```

```
set_property PACKAGE_PIN Y12 [get_ports S6]
```

Tcl Console | Messages | Log | Reports | Design Runs | I/O Ports

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (14)											
LED (8)	OUT			☑	12	LVCMOS33*	3.300		12	SLOW	NONE
LE...	OUT		AF24	☑	12	LVCMOS33*	3.300		12	SLOW	NONE
LE...	OUT		AE21	☑	12	LVCMOS33*	3.300		12	SLOW	NONE
LE...	OUT		Y22	☑	12	LVCMOS33*	3.300		12	SLOW	NONE
LE...	OUT		Y23	☑	12	LVCMOS33*	3.300		12	SLOW	NONE
LE...	OUT		AA23	☑	12	LVCMOS33*	3.300		12	SLOW	NONE
LE...	OUT		Y25	☑	12	LVCMOS33*	3.300		12	SLOW	NONE
LE...	OUT		AB26	☑	12	LVCMOS33*	3.300		12	SLOW	NONE
LE...	OUT		W23	☑	12	LVCMOS33*	3.300		12	SLOW	NONE
Scalar ports (6)											
S1	IN		AA10	☑	33	LVCMOS15*	1.500				NONE
S2	IN		AB10	☑	33	LVCMOS15*	1.500				NONE
S3	IN		AA13	☑	33	LVCMOS15*	1.500				NONE
S4	IN		AA12	☑	33	LVCMOS15*	1.500				NONE
S5	IN		Y13	☑	33	LVCMOS15*	1.500				NONE
S6	IN		Y12	☑	33	LVCMOS15*	1.500				NONE

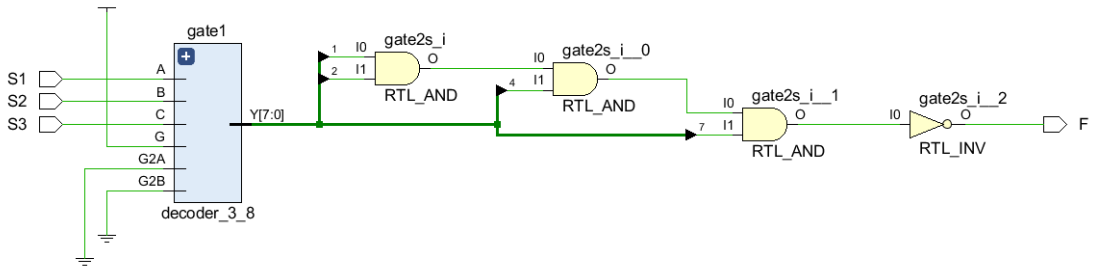


2 用 74LS138 译码器实现楼道灯控制器

2.1 新建工程 LampCtrl138。复制 D_74LS138.sym 和.sch 文件到工程目录。在 symbols 框里的第一个元件，就是 D_74LS138。根据前面原理，用原理图方式输入。1 用 VCC，0 用 GND。

```
`timescale 1ns / 1ps
module LampCtrl (
    S1,
    S2,
    S3,
    F
);
  input wire S1, S2, S3;
  output wire F;
  wire[7:0] out;
  wire G,G2A,G2B;
  VCC vcc1(G);
```

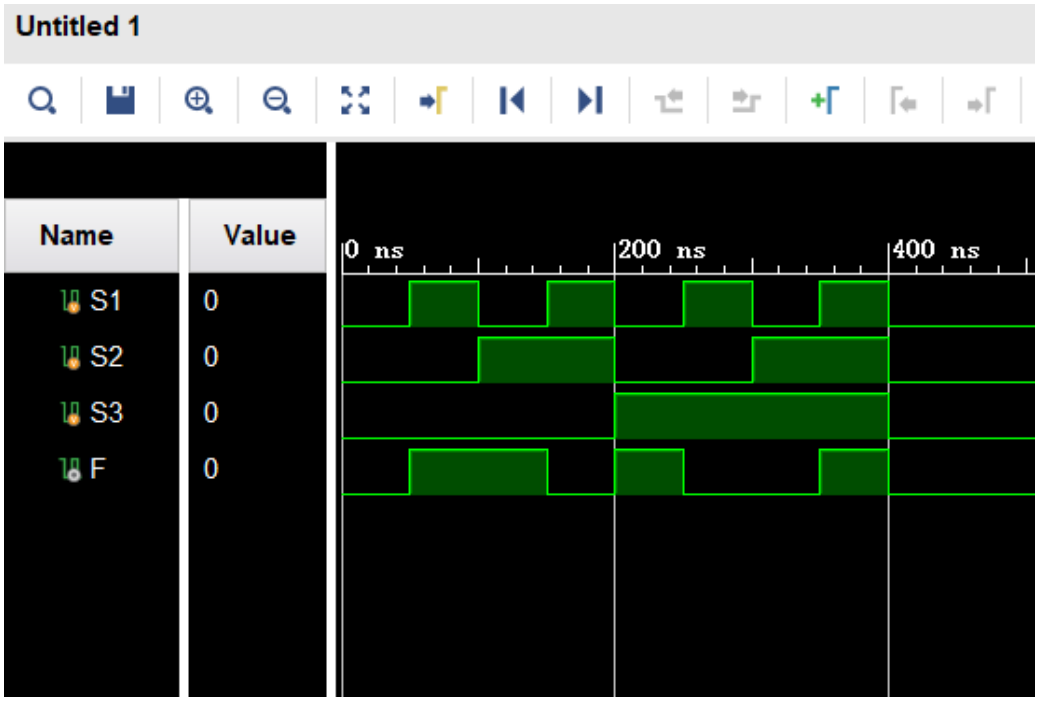
```
  GND gnd1(G2A), gnd2(G2B);
  decoder_3_8 gate1(S1,S2,S3,G,G2A,G2B,out);
  nand gate2s(F,out[1],out[2],out[4],out[7]);
endmodul
```



2.2 建立基准测试波形文件，输出仿真波形图。

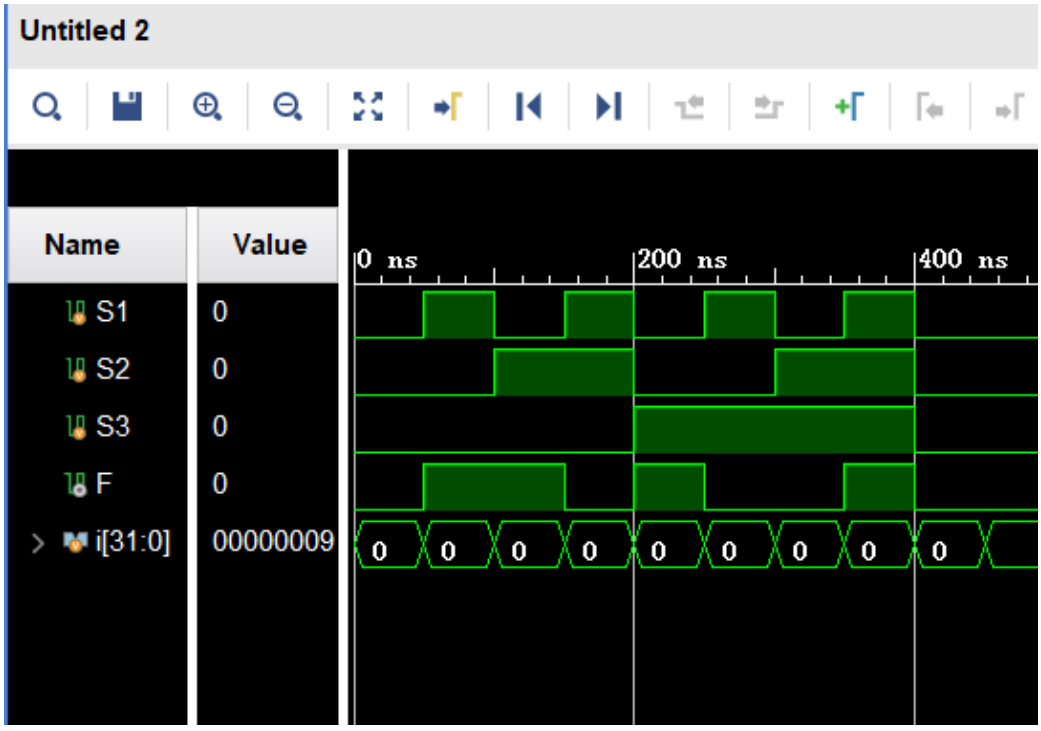
```
`timescale 1ns / 1ps
module tb_lampctrl_1 ();
  // Inputs
  reg  S1;
  reg  S2;
  reg  S3;

  // Output
  wire F;

  // Bidirs

  // Instantiate the UUT
  LampCtrl UUT1 (
      .S1(S1),
      .S2(S2),
      .S3(S3),
      .F (F)
  );
  // Initialize Inputs
  //   `ifdef auto_init
  initial begin
    S1 = 0;
    S2 = 0;
    S3 = 0;
    #50 S1 = 1;
    #50 S1 = 0;
```

```
    S2 = 1;
    #50 S1 = 1;
    #50 S1 = 0;
    S2 = 0;
    S3 = 1;
    #50 S1 = 1;
    #50 S1 = 0;
    S2 = 1;
    #50 S1 = 1;
    #50 S1 = 0;
    S2 = 0;
    S3 = 0;
  end
  //  `endif

endmodule
```




```
`timescale 1ns / 1ps
module tb_lampctrl_2 ();
  // Inputs
  reg  S1;
  reg  S2;
  reg  S3;
  // Output
```

```
  wire F;

  // Bidirs

  // Instantiate the UUT
  LampCtrl UUT2 (
      .S1(S1),
      .S2(S2),
      .S3(S3),
      .F (F)
  );
  // Initialize Inputs
  //   `ifdef auto_init
  integer i;
  initial begin
    for (i = 0; i <= 8; i = i + 1) begin
      {S3, S2, S1} <= i;
      #50;
    end
  end
  //  `endif

endmodule
```

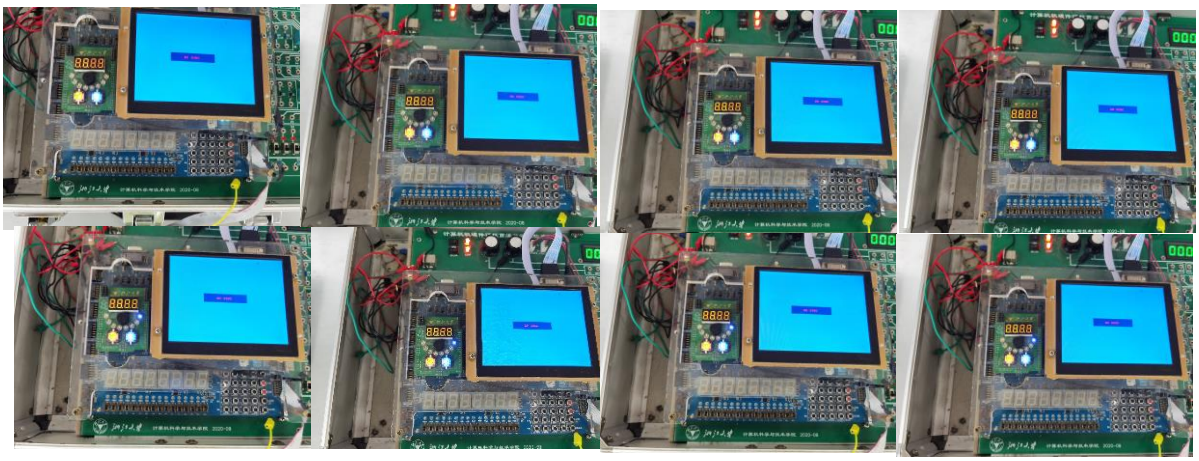


2.3 在 ucf 文件里输入必要的引脚约束。下载验证，使用 SW[1]-SW[3]控制 1 个 LED 开关。

```
# set_property IOSTANDARD LVCMOS33 [get_ports {LED[7]}]
# set_property IOSTANDARD LVCMOS33 [get_ports {LED[6]}]
# set_property IOSTANDARD LVCMOS33 [get_ports {LED[5]}]
# set_property IOSTANDARD LVCMOS33 [get_ports {LED[4]}]
# set_property IOSTANDARD LVCMOS33 [get_ports {LED[3]}]
# set_property IOSTANDARD LVCMOS33 [get_ports {LED[2]}]
# set_property IOSTANDARD LVCMOS33 [get_ports {LED[1]}]
# set_property IOSTANDARD LVCMOS33 [get_ports {LED[0]}]
# set_property PACKAGE_PIN W23 [get_ports {LED[0]}]
# set_property PACKAGE_PIN AB26 [get_ports {LED[1]}]
# set_property PACKAGE_PIN Y25 [get_ports {LED[2]}]
# set_property PACKAGE_PIN AA23 [get_ports {LED[3]}]
# set_property PACKAGE_PIN Y23 [get_ports {LED[4]}]
# set_property PACKAGE_PIN Y22 [get_ports {LED[5]}]
# set_property PACKAGE_PIN AE21 [get_ports {LED[6]}]
# set_property PACKAGE_PIN AF24 [get_ports {LED[7]}]
set_property IOSTANDARD LVCMOS15 [get_ports S1]
set_property IOSTANDARD LVCMOS15 [get_ports S2]
set_property IOSTANDARD LVCMOS15 [get_ports S3]
set_property PACKAGE_PIN AA10 [get_ports S1]
set_property PACKAGE_PIN AB10 [get_ports S2]
set_property PACKAGE_PIN AA13 [get_ports S3]
set_property PACKAGE_PIN AF24 [get_ports F]
set_property IOSTANDARD LVCMOS33 [get_ports F]
# set_property IOSTANDARD LVCMOS15 [get_ports S4]
# set_property IOSTANDARD LVCMOS15 [get_ports S5]
# set_property IOSTANDARD LVCMOS15 [get_ports S6]
# set_property PACKAGE_PIN AA12 [get_ports S4]
# set_property PACKAGE_PIN Y13 [get_ports S5]
# set_property PACKAGE_PIN Y12 [get_ports S6]
3
```

Tcl Console | Messages | Log | Reports | Design Runs | I/O Ports

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM
All ports (4)													
Scalar ports (4)													
	OUT		AF24	☑	12	...	3.300		12	SLOW	NONE	FP_VTT_50	
	IN		AA10	☑	33	...	1.500				NONE	NONE	
	IN		AB10	☑	33	...	1.500				NONE	NONE	
	IN		AA13	☑	33	...	1.500				NONE	NONE	



四、 实验结果分析

仿真波形与 ppt 一致，任务一下载验证时，按二进制码的顺序拨动 SW 开关，显示灯依次熄灭，任务二下载验证时，只有当奇数个开关开启时，显示灯才亮。

五、 讨论心得

我们每一个文件都相当于是一个元器件，括号里面的内容是此元器件的输入输出接口。 对于 top module 来说，接口用于在引脚约束以后，对应于开发板上一个个按钮、LED 灯等 器件，供我们在开发板上手动操作并观察效果。对于其他元器件来说，接口就是供被调用使用的。