

实验02 简单组合逻辑电路

【实验题目】

- 进一步使用Logisim设计组合逻辑电路，并用Verilog语言对设计的电路进行描述。

【实验目的】

- 熟练掌握Logisim的基本用法
- 进一步熟悉Logisim更多功能（利用真值表生成电路）
- 用Logisim设计组合逻辑电路并进行仿真
- 初步学习Verilog语法

【实验环境】

- PC机一台，能流畅地链接校园网
- Logisim仿真工具
- <www.vlab.ustc.edu.cn>

【实验过程】

Step 1 用真值表自动生成电路

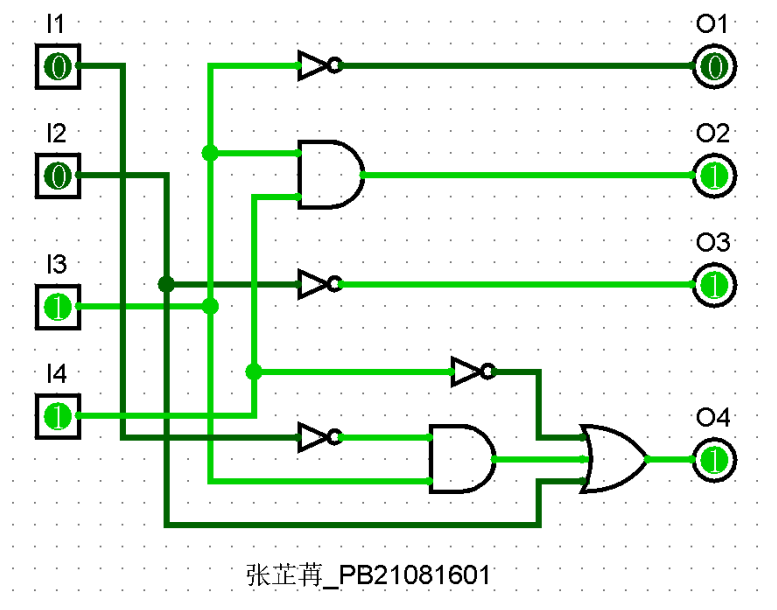
手工通过真值表生成电路的方法：

- 根据真值表画出各输出项的卡诺图
- 通过卡诺图写出各输出项的逻辑表达式
- 根据逻辑表达式画出电路图，完成电路设计

而Logisim可以帮助我们完成上述步骤中大部分工作。

1. 在Logisim中新建一个电路图，名为 lab2，新建电路名为 真值表生成电路 在电路图中根据输入输出数量放置引脚，并给其标号。
2. 在菜单栏的“Project”选项卡中找到“Analyze Circuit”选项，并选中。在弹出的窗口中选择“Table”选项，按照实验手册中所给的真值表修改输出值，最后点击“Build Circuit”便可生成电路。

这是按照指导生成的电路：

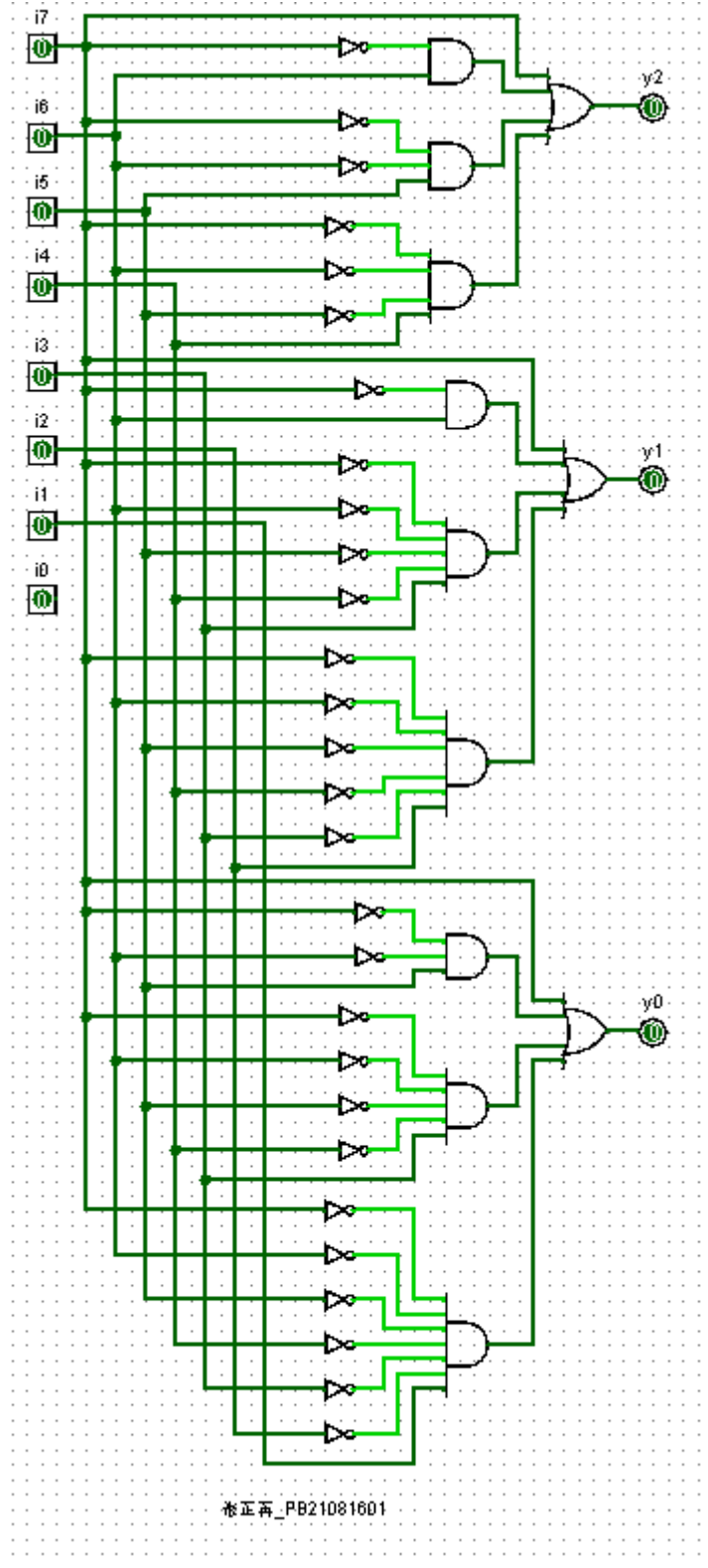


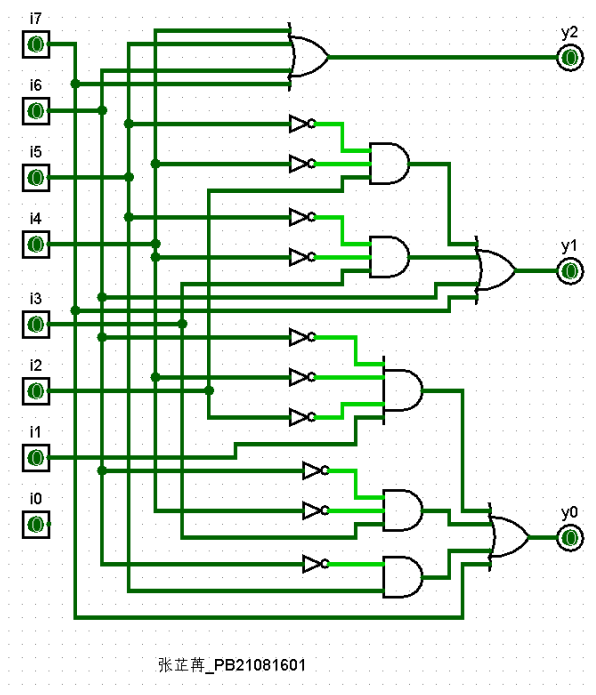
Step 2 用表达式生成电路图

通过真值表生成电路确实能为我们减少工作，但是也存在不足之处，真值表条目数与输入项个数呈指数相关，当输入信号数量较多时，编辑真值表也是一项非常繁重的工作。此时，需要寻找一个更为便捷的方式。

可以在 Logisim 中直接输入表达式生成电路，在“Project” --> “Analyze Circuit”的弹出窗口中选择“Expression”选项，填入每个输出信号的表达式。最后点击“Build Circuit”生成电路。有时候手动输入的表达式并不是最简形式，最终生成的电路也会占用较多的逻辑门，此时可以借助“Minimized”选项卡对表达式进行简化，进而减少电路使用的逻辑门数量，电路输入信号不多的情况下，该窗口还能显示卡诺图。

下图分别是按照指导生成的化简前与化简后的电路：





通过“Project” --> “Get Circuit Statistics”选项统计电路的基本信息：

Logisim: 表达式生成电路 Statistics

Component	Library	Si...	Un...	Re...
Pin	Wiring	11	11	11
NOT Gate	Gates	10	10	10
AND Gate	Gates	5	5	5
OR Gate	Gates	3	3	3
Label	Base	1	1	1
TOTAL (without proj...		30	30	30
TOTAL (with subcirc...		30	30	30

Close

Step 3 Verilog HDL 语法入门

【实验练习】

题目 1：依据如下真值表，通过 Logisim 编辑真值表功能，完成电路设计。

输入			输出	
Ci-1	Ai	Bi	Si	Ci
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

真值表：

Combinational Analysis

—□×

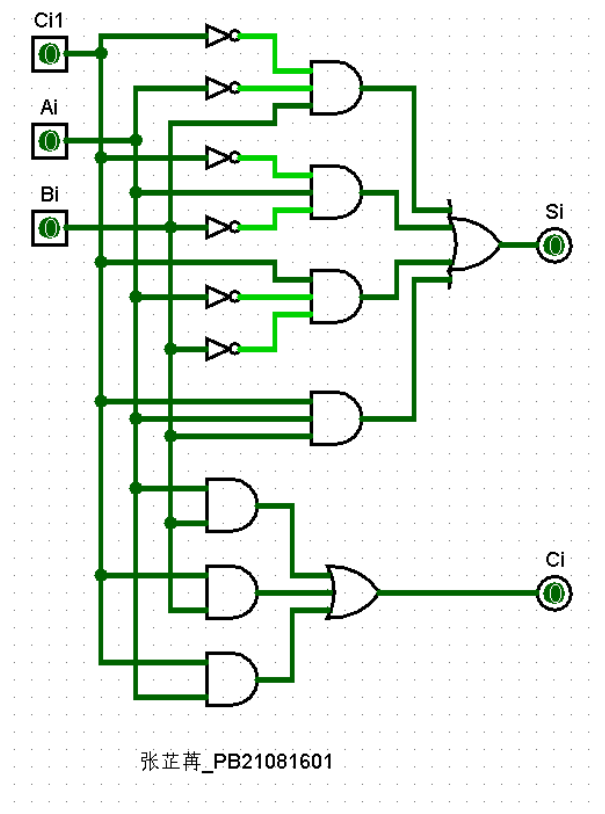
File Edit Project Simulate Window Help

InputsOutputsTableExpressionMinimized

Ci-1	Ai	Bi	Si	Ci
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Build Circuit

有真值表自动生成电路图：



题目 2：根据下列真值表，通过 Logisim 的编辑表达式功能完成电路设计。

输入						输出							
G1	G2	G3	A2	A1	A0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	1	X	X	X	1	1	1	1	1	1	1	1
0	X	X	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

输入表达式自动生成电路：

Output:

$$\overline{G1} + \overline{A2} + \overline{A1} + \overline{A0} + G3 + G2$$

Output:

$$\overline{G1} + \overline{A1} + \overline{A0} + A2 + G3 + G2$$

Output:

$$\overline{G1} + \overline{A2} + \overline{A1} + A0 + G3 + G2$$

Output:

$$\overline{G1} + \overline{A1} + A0 + A2 + G3 + G2$$

Output:

$$\overline{G1} + \overline{A2} + \overline{A0} + A1 + G3 + G2$$

Output:

$$\overline{G1} + \overline{A0} + A1 + A2 + G3 + G2$$

Output:

$$\overline{G1} + \overline{A2} + A0 + A1 + G3 + G2$$

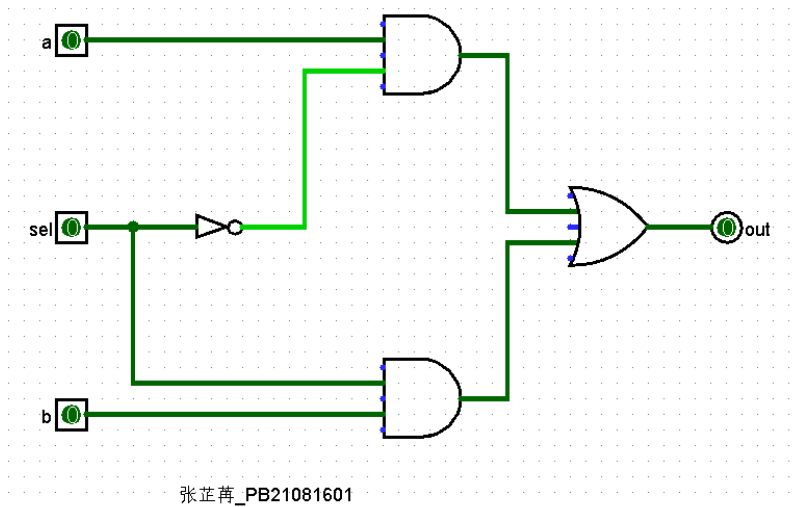
Output:

$$\overline{G1} + A0 + A1 + A2 + G3 + G2$$

由于电路图太长，而电脑不能实现连续截长图，故将电路源码放在 repository https://github.com/StellaVesperi/Public-for_public_use/blob/master/lab2.circ lab 2 文件的 T2 circuit 中，便于助教检查。

题目 3：使用 Logisim 绘制 1bit 位宽的二选一选择器电路图，并编写 Verilog 代码。输入信号a, b, sel，输出信号为 out, sel 为 0 时选通 a 信号。

电路图：

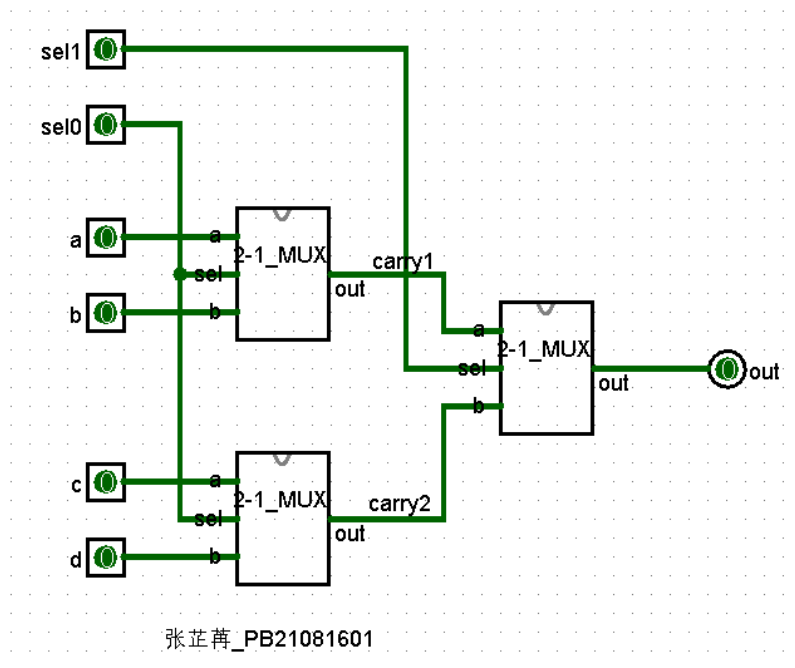


Verilog 代码：

```
module 2-1_MUX(  
    input a, b, sel,  
    output out);  
  
    assign out = (a & ~sel) | (b & sel);  
endmodule
```

题目 4：通过实例化题目 3 中的二选一选择器，用 Verilog 实现一个四选一选择器，并画出对应的电路图。输入信号为a, b, c, d, sel1, sel0, out，sel1 和 sel0 都为 0 时选中 a 信号。

电路图：



Verilog 代码：

```
module 4-1_MUX(  
    input sel0, sel1, a, b, c, d,  
    output out);  
    wire carry1, carry2;  
    2-1_MUX mux1(  
        .a    (a    ),  
        .b    (b    ),  
        .sel   (sel0 ),  
        .out   (carry1));  
    2-1_MUX mux2(  
        .a    (c    ),  
        .b    (d    ),  
        .sel   (sel0 ),  
        .out   (carry2));  
    2-1_MUX mux3(  
        .a    (carry1),  
        .b    (carry2),  
        .sel   (sel1  ),  
        .out   (out   ));  
endmodule
```


题目 5：根据前面用到的八位优先编码器真值表，编写 verilog 代码。

输入								输出		
i7	i6	i5	i4	i3	i2	i1	i0	y2	y1	y0
1	x	x	x	x	x	x	x	1	1	1
0	1	x	x	x	x	x	x	1	1	0
0	0	1	x	x	x	x	x	1	0	1
0	0	0	1	x	x	x	x	1	0	0
0	0	0	0	1	x	x	x	0	1	1
0	0	0	0	0	1	x	x	0	1	0
0	0	0	0	0	0	1	x	0	0	1
0	0	0	0	0	0	0	1	0	0	0

根据真值表列出各输出项的逻辑表达式：

```
y2 = i7 + ~i7&i6 + ~i7&~i6&i5 + ~i7&~i6&~i5&i4
y1 = i7 + ~i7&i6 + ~i7&~i6&~i5&~i4&i3 + ~i7&~i6&~i5&~i4&~i3&i2
y0 = i7 + ~i7&~i6&i5 + ~i7&~i6&~i5&~i4&i3 + ~i7&~i6&~i5&~i4&~i3&i2
```

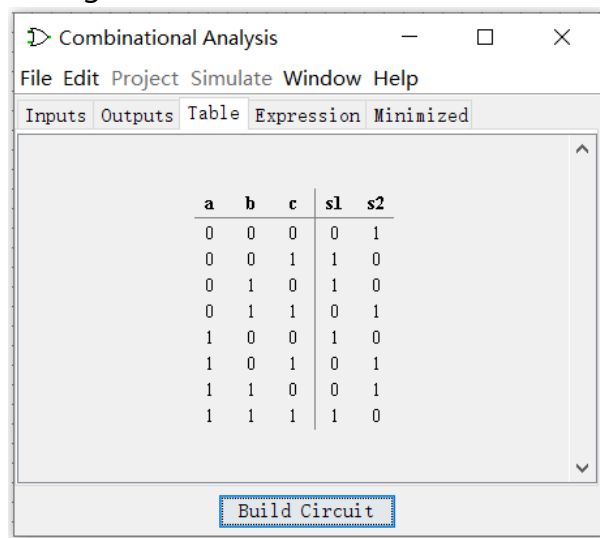
Verilog 代码：

```
module t5(
    input i7, i6, i5, i4, i3, i2, i1, i0,
    output y2, y1, y0);
    assign y2 = i7 + ~i7&i6 + ~i7&~i6&i5 + ~i7&~i6&~i5&i4;
    assign y1 = i7 + ~i7&i6 + ~i7&~i6&~i5&~i4&i3 +
~i7&~i6&~i5&~i4&~i3&i2;
    assign y0 = i7 + ~i7&~i6&i5 + ~i7&~i6&~i5&~i4&i3 +
~i7&~i6&~i5&~i4&~i3&i2;
endmodule
```

题目 6：阅读如下 Verilog 代码，描述其功能，并画出其对应的电路图。

```
module test(
input a,b,c,
output s1,s2);
assign s1= ~a &~b & c | ~a & b &~c | a &~b &~c | a & b & c;
assign s2= ~a & b & c | a &~b & c | a & b &~c | ~a &~b &~c;
endmodule
```

用 Logisim 列出真值表如下图：

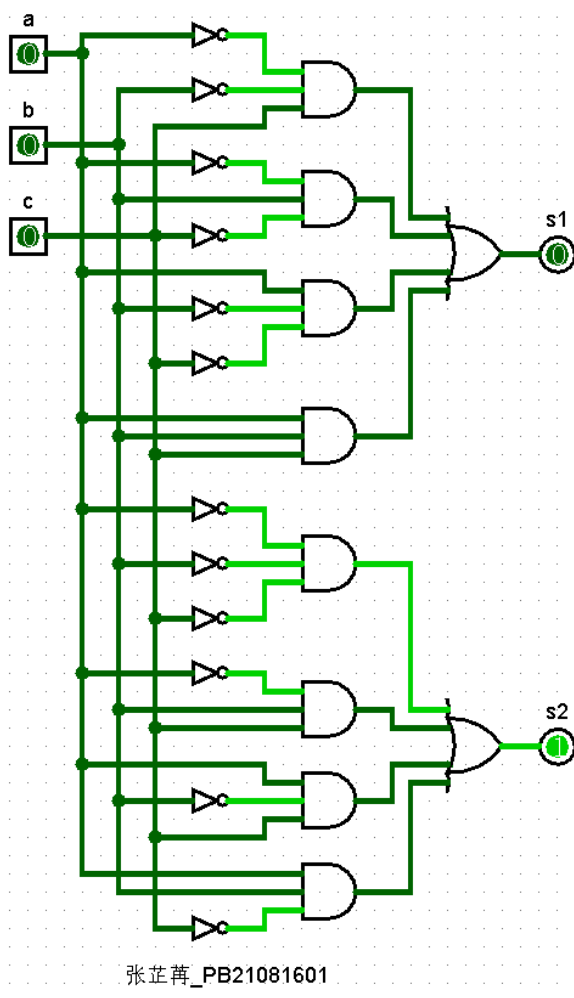


a	b	c	s1	s2
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Build Circuit

由真值表可知，当输入有0个或2个为1时，s1为0，s2为1；而当输入有1个或3个为1时，s1为1，s2为0。从而得知该电路功能为判断输入为高电平信号的个数（或奇偶性）。

由真值表生成电路图：



【总结与思考】

1. 通过本次实验我学会了在 Logisim 软件中：

- 用真值表生成电路
- 输入表达式生成电路

- 查看电路的各种信息，自动化简电路

以上这些手段使得我在运用 Logisim 创建电路图的过程中省去了不少时间。

2. 同时我还学会了一些基本的 verilog 语法，能用 verilog 语言描述一些简单的电路。
3. 在经历了第一次实验之后，我对各种工具的使用都更加熟练了，所以我认为本次实验任务量适中，难度恰到好处。
4. 希望以后能多介绍一些详细的 verilog 语法，并配上例题，如，用不同的方式描述同一个电路，这样会对 verilog 有更深刻的认识体会。