**中国科学技术大学计算机学院**

**《数字电路实验》报告**



实验题目：简单组合逻辑电路

学生姓名：\_ \_李远航\_\_ \_\_\_

学生学号： \_PB20000137\_ \_

完成日期： 2021.10.24\_ \_

计算机实验教学中心制

2020年09月

【实验题目】

**简单组合逻辑电路**

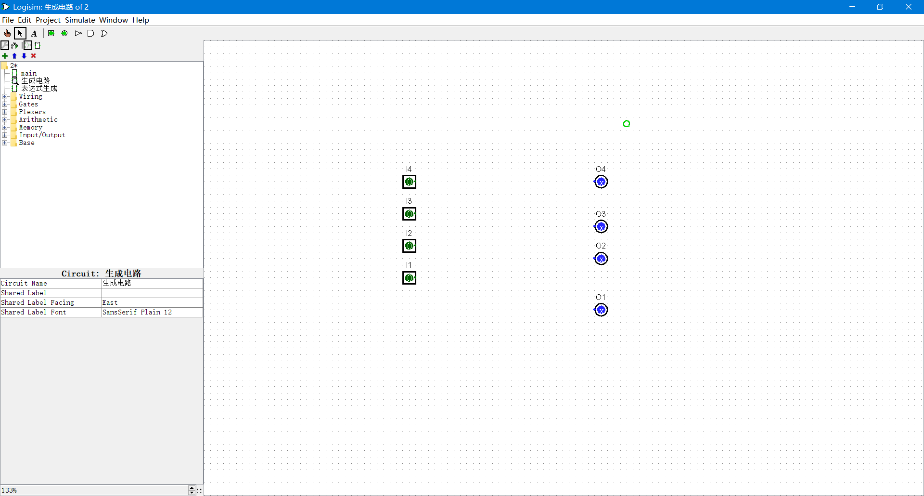
【实验目的】

* 熟悉Logisim的基本用法
* 进一步熟悉Logisim的更多功能
* 用Logisim设计组合逻辑电路并进行仿真
* 初步学习Verilog语法
* 学习使用Verilog OJ

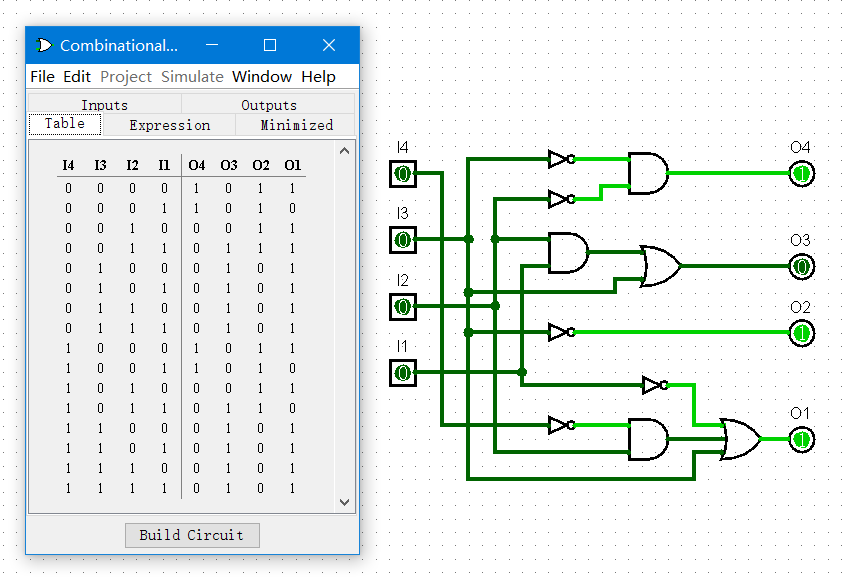
【实验环境】

* PC一台
* Windows 10 操作系统
* Java 8（64 bit）
* Logisim 仿真工具

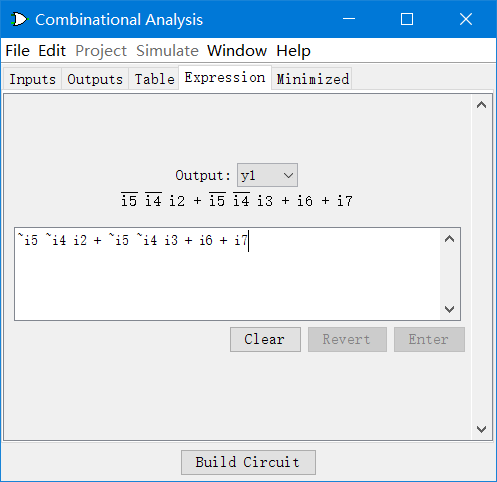
【实验过程】

1. 用真值表自动生成电路
   1. 在Logisim中放入输入输出引脚

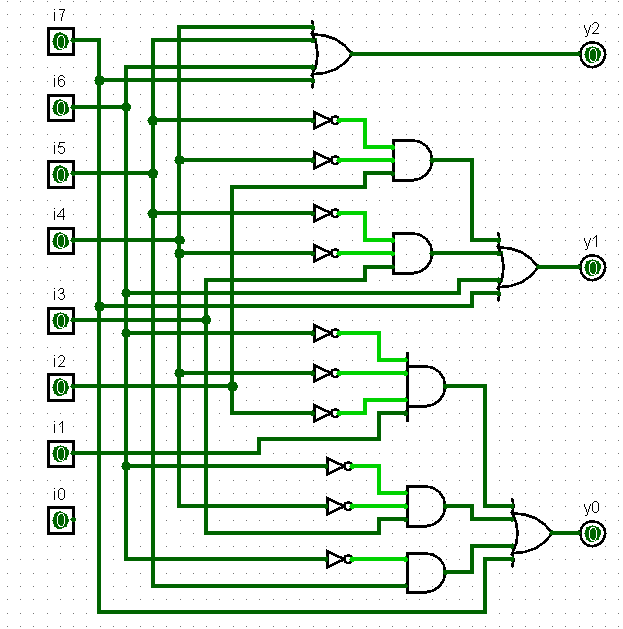
**图1.1**

* 1. 在菜单栏的“Project”选项卡中找到“Analyze Circuit”选项，选择“Table”选项，按照真值表填写，并生成电路

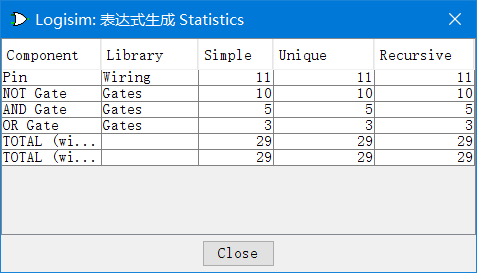
**图1.2**

1. 用表达式生成电路图
   1. 根据真值表写出各输出信号的表达式并借助“Minimized”选项卡化简

**图2.1**

* 1. 生成电路

**图2.2**

* 1. 通过“Project”--> “Get Circuit Statistics”选 项统计电路的基本信息

**图2.3**

1. Verilog HDL语法入门
   1. 例一：初步学习Verilog语法

module test (

    input in,

    output out,

    output out\_n);

  assign out=in;

  assign out\_n=~in;

endmodule

* 1. 两种不同的半加器电路

module add (

    input a , b ,

    output sum,cout);

    assign cout = a & b;

    assign sum = a ^ b;

endmodule

module add (

    input a,b,

    output sum,cout);

  assign {cout,sum} = a+b;

endmodule

* 1. 使用前面例子中的半加器，构造一个全加器

module full\_add(

    input a,b,cin,

    output sum,cout);

  wire s,carry1,carry2;

  full\_add add\_inst1(

             .a (a ),

             .b (b ),

             .sum (s ),

             .cout (carry1));

  full\_add add\_inst2(

             .a (s ),

             .b (cin ),

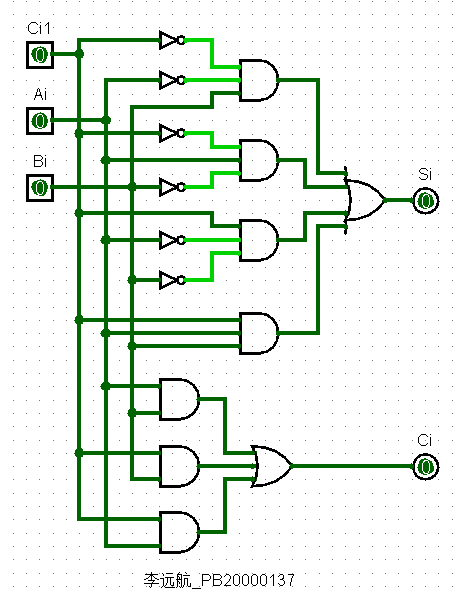
             .sum (sum ),

             .cout (carry2));

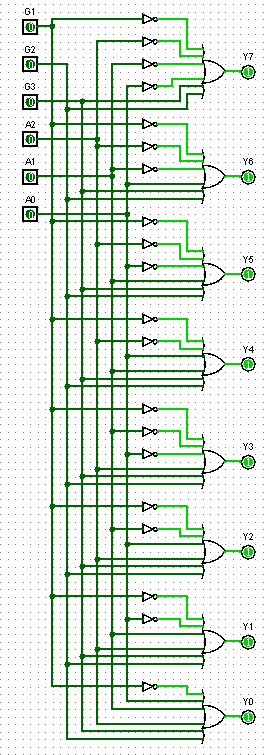
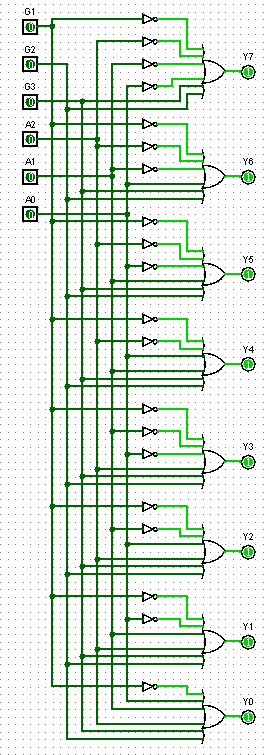
  assign cout = carry1 | carry2;

endmodule

【实验练习】

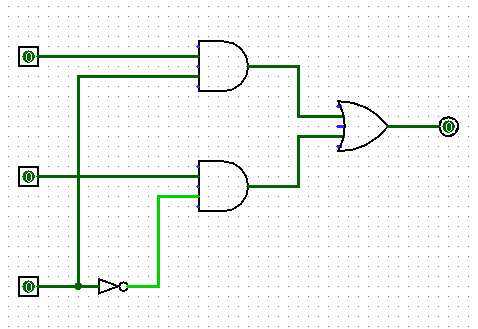
1. 根据真值表，完成电路设计

**题图1.1**

1. 通过编辑表达式功能完成电路设计，注意先化简电路可以让电路图更简单



**题图2.1**

1.  电路图和代码如下：

**题图3.1**

module MUX (

    input a,b,sel,

    output out);

  assign out=(a&~sel)|(b&sel);

endmodule

1. 电路图和代码如下：

module Fourtoone (

    input a,b,c,d,sel0,sel1,

    output out);

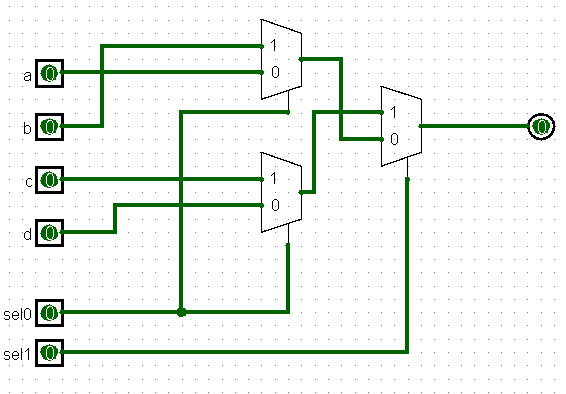
  wire carry1,carry2;

  MUX mux1(.a(a),.b(b),.sel(sel0),.out(carry1));

  MUX mux2(.a(c),.b(d),.sel(sel0),.out(carry2));

  MUX mux3(.a(carry1),.b(carry2),.sel(sel1),.out(out));

endmodule



**题图4.1**

1. 代码如下：

module test (

    input i7,i6,i5,i4,i3,i2,i1,i0,

    output y2,y1,y0

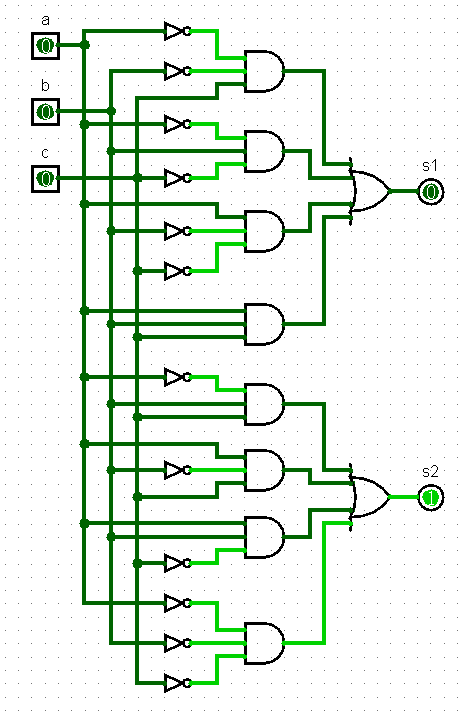
  );

  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&i1);

endmodule

1. 观察s1和s2的输出 可知 s1是检测输入中是否有奇数个1 ，若有则输出出1；s2是检测输入中是否有偶数个1，若有则输出1，电路图如下：

**题图6.1**

【总结与思考】

* 学会使用真值表和表达式生成电路
* 初步学习了Verilog的语法
* 本次实验难易程度适中，任务量适中
* 实验软件稳定性不佳，经常出现崩溃的情况
* 可以增加一些对于Verilog有关语法的描述以及如何编译运行Verilog语言