

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：____徐奥____
学生学号：____PB20061343____
完成日期：_2021 年 10 月 21 日_

计算机实验教学中心制

2020 年 09 月

【实验题目】

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

【实验目的】

1. 熟练掌握 Logisim 的基本用法
2. 进一步熟悉 Logisim 更多功能
3. 用 Logisim 设计组合逻辑电路并进行仿真
4. 初步学习 Verilog 语法

【实验环境】

1. PC 一台, 能流畅的连接校园网
2. Logisim 仿真工具
3. vlab.ustc.edu.cn (jre、Logisim 工具以及 Verilog 语法介绍都可在此网站获取)

【实验过程】

1. 用真值表自动生成电路。步骤为：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。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，按照前面的真值表修改输出值，最后点击“Build Circuit”便可生成电路。
2. 用表达式生成电路图。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。
3. Verilog HDL 语法入门。每个模块都是以关键字 module 开头，以 endmodule 结束。module 后面是模块名，括号内是输入输出信号的声明。如果模块功能较复杂的话，可能会用到一些中间信号，那就要在模块内部声明。逻辑描述部分是每个模块的主体，用于描述该电路的行为特性。模块调用在 Verilog 中也非常重要，在电路较复杂时，我们需要将其分解成若干个子电路，最后再将子电路整合，或者复用第三方以及自己之前设计的功能模块时，都需要用到模块调用。

【实验练习】

题目 1：用 Logisim 编辑真值表功能，完成电路设计。

首先，在“Analyze Circuit”功能中输入题目给出的真值表，点击“Build Circuit”即可生成电路。生成的电路如图 1。

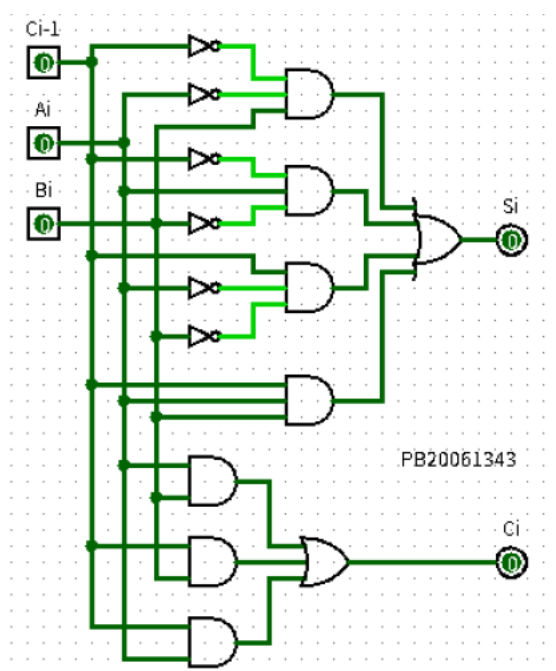


图 1

题目 2：用 Logisim 编辑表达式功能，完成电路设计。

分析真值表可知各输出项的表达式

$$\begin{aligned} Y7 &= \sim(G1 \& \sim G2 \& \sim G3 \& A2 \& A1 \& A0) \\ Y6 &= \sim(G1 \& \sim G2 \& \sim G3 \& A2 \& A1 \& \sim A0) \\ Y5 &= \sim(G1 \& \sim G2 \& \sim G3 \& A2 \& \sim A1 \& A0) \\ Y4 &= \sim(G1 \& \sim G2 \& \sim G3 \& A2 \& \sim A1 \& \sim A0) \\ Y3 &= \sim(G1 \& \sim G2 \& \sim G3 \& \sim A2 \& A1 \& A0) \\ Y2 &= \sim(G1 \& \sim G2 \& \sim G3 \& \sim A2 \& A1 \& \sim A0) \\ Y1 &= \sim(G1 \& \sim G2 \& \sim G3 \& \sim A2 \& \sim A1 \& A0) \\ Y0 &= \sim(G1 \& \sim G2 \& \sim G3 \& \sim A2 \& \sim A1 \& \sim A0) \end{aligned}$$

在“Analyze Circuit”功能中输入表达式，点击“Build Circuit”即可生成电路。生成的电路如图 2。

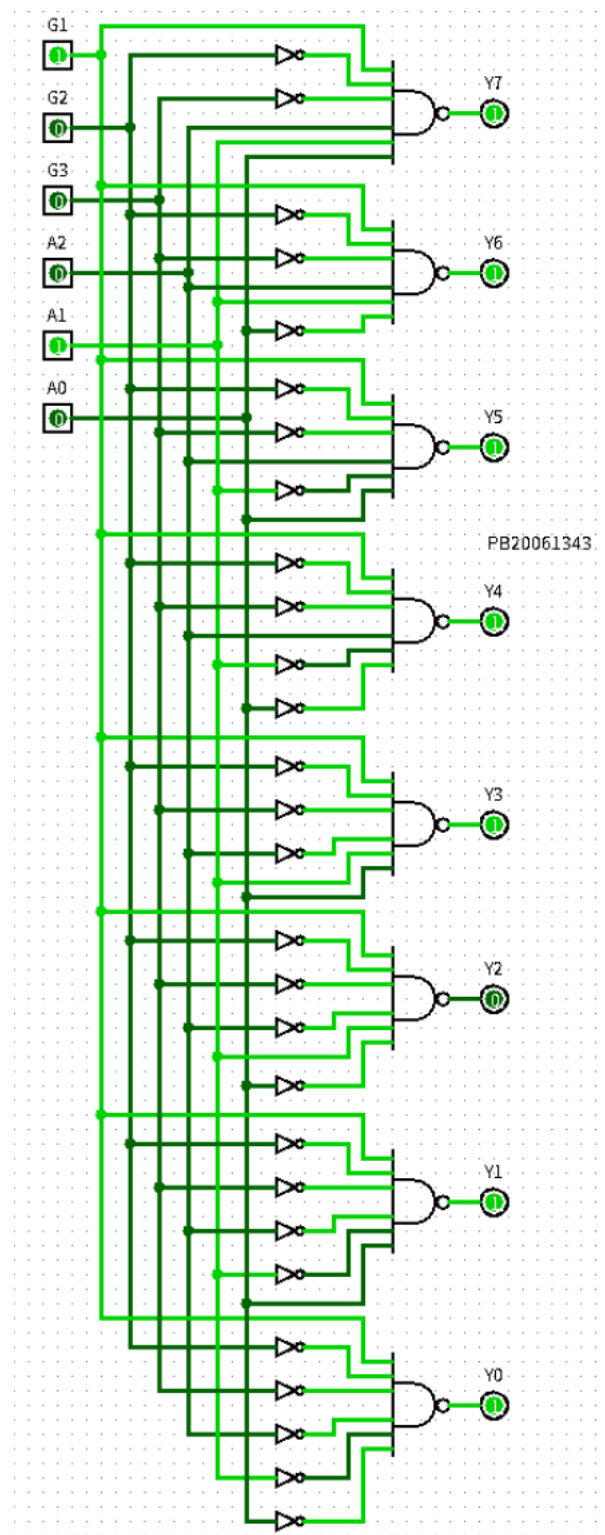


图 2

题目 3: 使用 Logisim 绘制 1bit 位宽的二选一选择器电路图，并根据生成的电路图编写 Verilog 代码。

首先，绘制 1bit 位宽的二选一选择器电路图，如图 3.

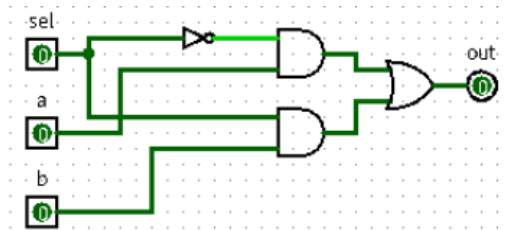


图 3

之后，编写 Verilog 代码，如图 4.

```
module mux2_1(
    input a,b,sel,
    output out);
    assign out = ~sel & a | sel & b;
endmodule
```

图 4

题目 4: 通过例化题目 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。

首先，编写 Verilog 代码，如图 5.

```
module mux2_1(
    input a,b,sel,
    output out);
    assign out = ~sel & a | sel & b;
endmodule

module mux4_1(
    input a,b,c,d,sel0,sel1,
    output out);
    wire y1,y2;
    mux2_1 a1(
        .a      (a      ),
        .b      (b      ),
        .sel     (sel0   ),
        .out     (y1     ));
    mux2_1 a2(
        .a      (c      ),
        .b      (d      ),
        .sel     (sel0   ),
        .out     (y2     ));
    mux2_1 a3(
        .a      (y1      ),
        .b      (y2      ),
        .sel     (sel1   ),
        .out     (out     ));
endmodule
```

图 5

封装二选一数据选择器，实现四选一数据选择器。如图 6.

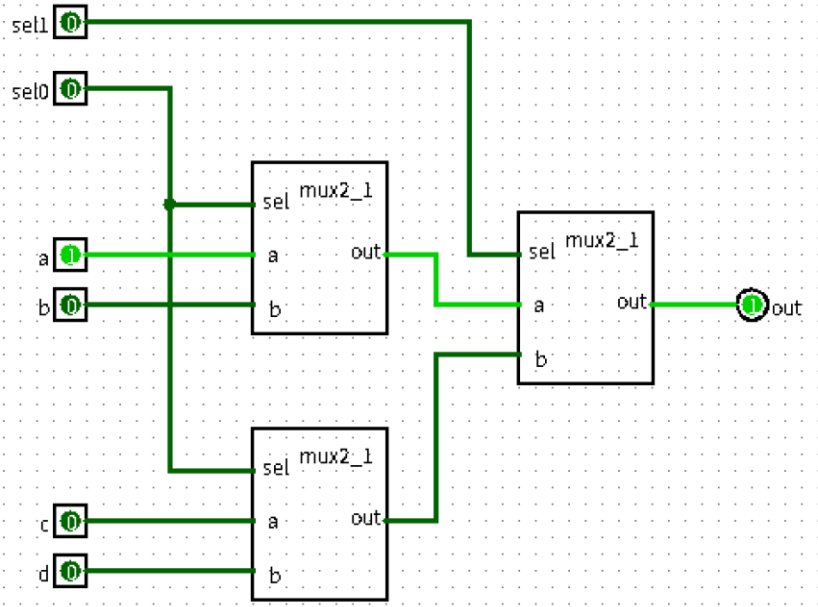


图 6

题目 5：根据已知的八位优先编码器真值表，编写 verilog 代码。

代码如图 7.

```

module sel(
  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

```

图 7

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

根据代码可知, 当输入的 a, b, c 中有奇数个为 1 时, s1 为 1, s2 为 0; 当 a, b, c 有偶数个为 1 时, s1 为 0, s2 为 1. 所以这个电路实现了判断三个输入中 1 的奇偶性。

电路图如图 8.

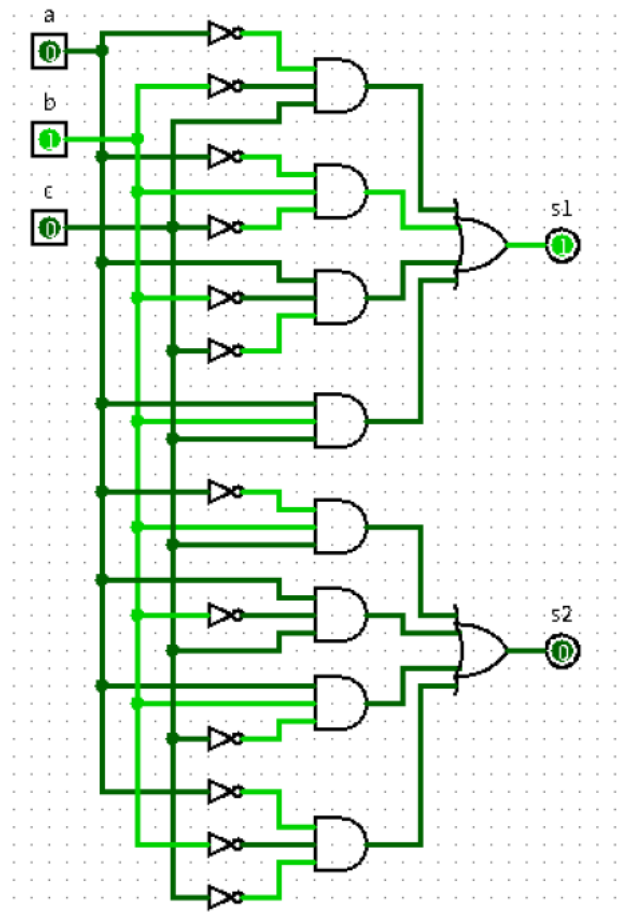


图 8

【总结与思考】

1. 通过本次实验的练习，学会了在 Logisim 使用真值表和表达式设计电路，相比较于之前创建电路时大量的拖拽、布局、连线，本次实验的两种方法十分便捷高效。
2. 这次实验后，我对 Logisim 的基本用法更加熟悉，逐渐意识到这个软件的强大功能，它不仅可以让们清晰地看到电路图，还提供了方便的生成电路的方法。
3. 初步学习了 Verilog 语法。这个语言和 C 语言有很多相似之处，但是它的代码执行不是串行的，而是并行执行。这实际上是由于这个电路是直接描述电路的，当电路建立好后，各个逻辑函数都同时会执行。
4. 学会了 Verilog 的模块调用功能，这个功能可以大量节省代码量，而且只要保证已经写好的模块内部没有 BUG，后续在调用它时，不需要担心它会出现问题，这也大大节省了 Debug 的时间。这种模块调用功能于 C 语言中的函数功能相似。
5. 本次实验内容充实，难度适中，相比较于第一次实验，干货更多，重复性的工作减少了，建议保持。