**实验二 组合电路设计实验报告**

**姓名**：丘绎楦  **学号**：1820221050

**班级**：07812201 **手机**：15010901562

注：黑色字体内容不能改动，蓝色字体内容（为示例或说明）需删除和修改。

1. **实验题目**

设计一个组合电路，输入一个3位的数字，输出一个六位的二进制数字，且输出数字的值等于输入数字值的平方。

1. **实验约束**

* 电路设计时只能使用或非门和非门进行实现。
* 采用Verilog实现时使用结构化描述方式。

1. **电路设计**
   1. **规范化**

输入：三个二进制变量S[2]、S[1]、S[0]

输出：六个二进制变量 Y[5]、Y[4]、Y[3]、Y[2]、Y[1]、Y[0]

电路中通过或非门和非门来对输入进行处理，通过对每一位二进制变量的运算，得到每一个二进制变量输出Y[i]

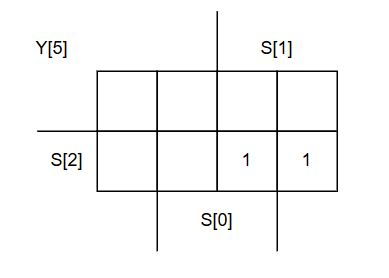
* 1. **形式化**

| **输入** | | | | **输出** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S[2] | S[1] | S[0] | 值 | Y[5] | Y[4] | Y[3] | Y[2] | Y[1] | Y[0] | 值 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 1 | 3 | 0 | 0 | 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 0 | 4 | 0 | 1 | 0 | 0 | 0 | 0 | 16 |
| 1 | 0 | 1 | 5 | 0 | 1 | 1 | 0 | 0 | 1 | 25 |
| 1 | 1 | 0 | 6 | 1 | 0 | 0 | 1 | 0 | 0 | 36 |
| 1 | 1 | 1 | 7 | 1 | 1 | 0 | 0 | 0 | 1 | 49 |

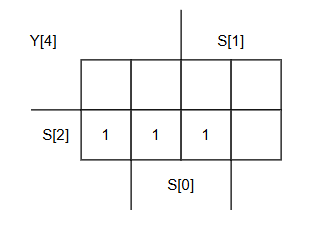
**真值表：**

* 1. **优化**

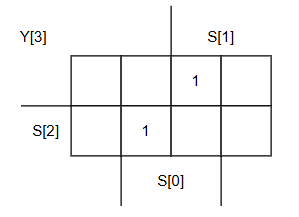
以下是每一个输出对应的卡诺图。



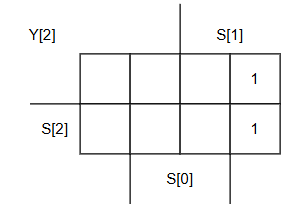
Y[5] = S[2]S[1]



Y[4] = S[2]S[0] + S[2] =

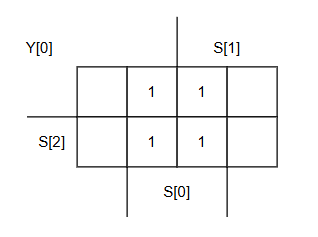


Y[3] = =



Y[2] =

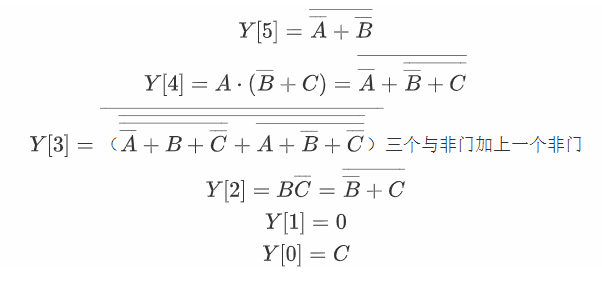
Y[1] = 0



Y[0] = S[0]

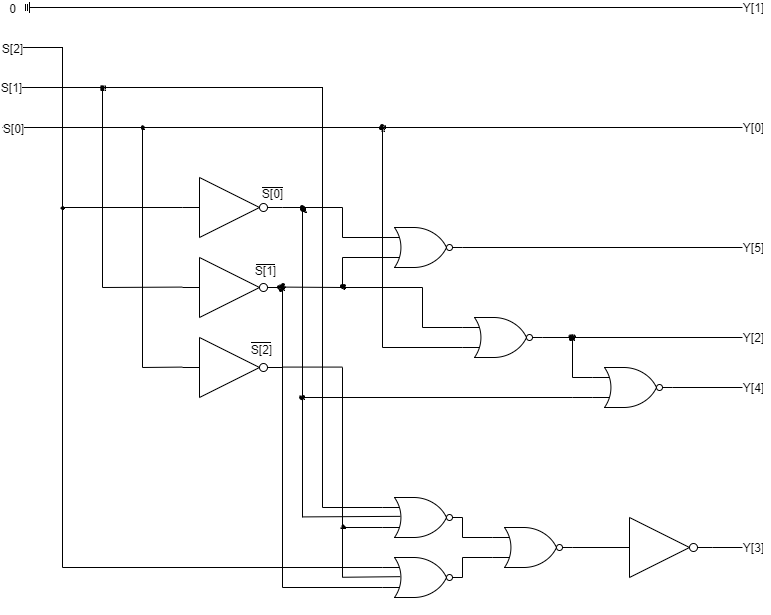
综上，每一个变量化成**或非门**和**非门**的组合：

下表中令A=S[2], B=S[1], C=S[0]，用来便于表示：

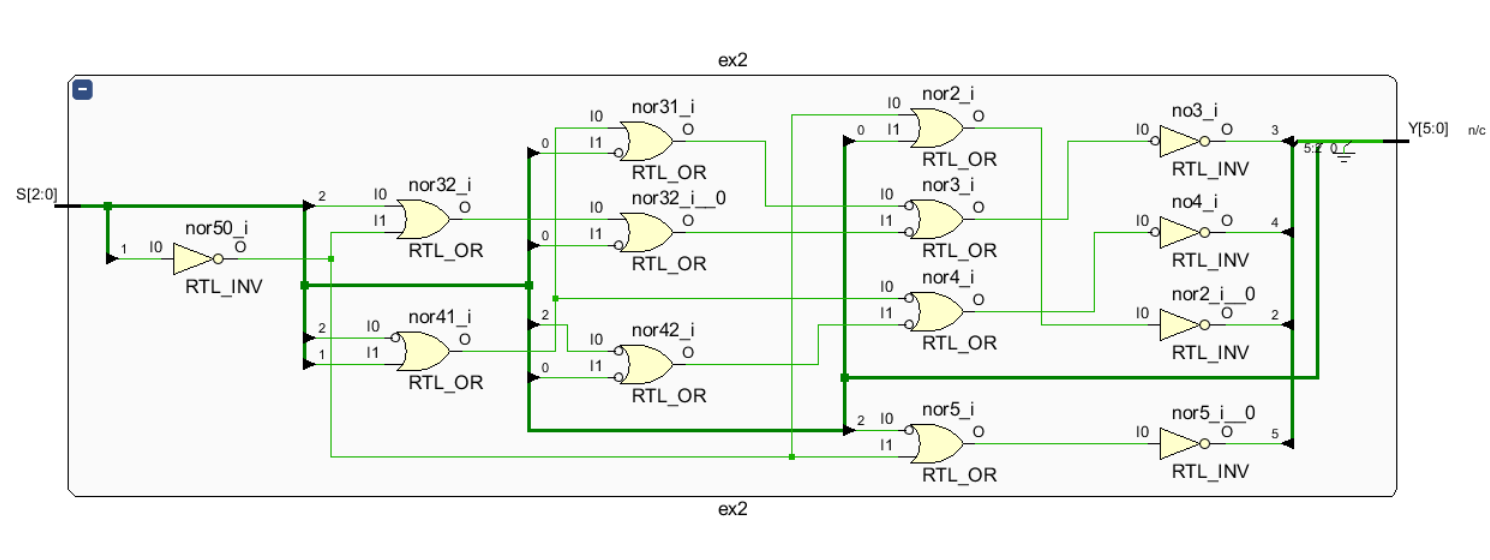


* 1. **工艺映射**

**工业映射前：**

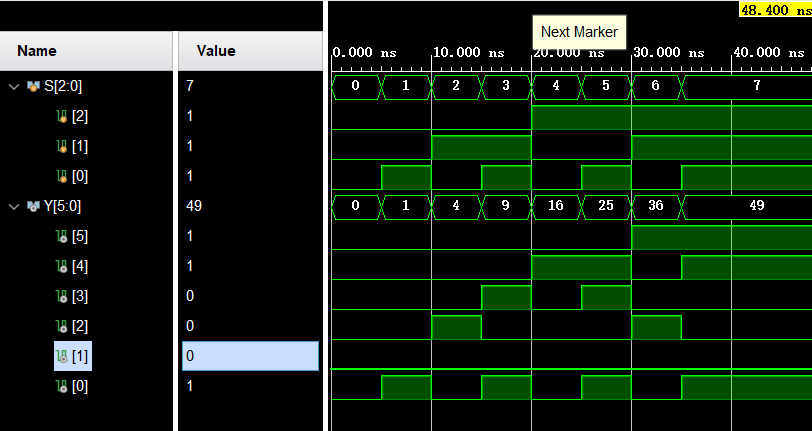


**工业映射后：**



1. **电路实现**
2. `timescale 1ns/1ps
3. module ex2(S,Y);
4. input [2:0] S;
5. output [5:0] Y;
6. wire temp\_41, temp\_42;  *// 中间变量*
7. wire temp\_31, temp\_32;  *// 中间变量*
8. wire not3, not4;  *// 中间变量,~Y[3],~Y[4]*
10. nor  *// 或非门运算*
11. nor5(Y[5], ~S[2], ~S[1]),  *// Y[5]*
12. nor2(Y[2], ~S[1], S[0]),  *// Y[2]*
14. nor41(temp\_41, ~S[2], S[1]),
15. nor42(temp\_42, ~S[2], ~S[0]),
16. nor4(not4, temp\_41, temp\_42),
18. nor31(temp\_31, ~S[2], S[1], ~S[0]),
19. nor32(temp\_32, S[2], ~S[1], ~S[0]),
20. nor3(not3, temp\_31, temp\_32);
22. not  *// 非门运算*
23. no3(Y[3], not3),  *// Y[3]*
24. no4(Y[4], not4);  *// Y[4]*
26. assign Y[1]=0;  *// Y[1]*
27. assign Y[0]=S[0];  *// Y[0]*
28. endmodule
29. **电路验证**
    1. **TestBench**
30. `timescale 1ns / 1ps
31. module testbench(
32. );
33. reg [2:0] S; //对应ex2 模块的输入
34. wire [5:0] Y; //对应ex2 模块的输出
35. initial begin
36. S[2]=1'b0; //初始化
37. #20 S[2]=1'b1; //之后每20ns S[2]变化一次，第一位变化一次
38. end
39. initial begin
40. S[1]=1'b0;
41. #10 S[1]=1'b1;
42. #10 S[1]=1'b0;
43. #10 S[1]=1'b1;//每10ns S[1]变化一次，第二位变化一次
44. end
45. initial begin
46. S[0]=1'b0;
47. #5 S[0]=1'b1;
48. #5 S[0]=1'b0;
49. #5 S[0]=1'b1;
50. #5 S[0]=1'b0;
51. #5 S[0]=1'b1;
52. #5 S[0]=1'b0;
53. #5 S[0]=1'b1;//每 5ns S[0]变化一次，第三位变化一次
54. end
55. ex2 ex2 (
56. .S(S),
57. .Y(Y)
58. );
59. endmodule
    1. **仿真结果**

该部分通过Vivado仿真。需要给出仿真得到的Wave，并进行验证解释。



从仿真得出的波形可以看出，当输入为0，1，2，3，4，5，6，7时，输出为0，1，4，9，16，25，36，49 ，输出值恰好为输入值的平方， 对应的六位二进制波形也和真值表相对应。

仿真结果正确。

1. **实验心得**

在这次数字逻辑课程的实验中，我深入探索了组合逻辑电路的设计原理和实现方法。通过使用或非门和非门，我不仅掌握了如何构建基本的逻辑门，还学习了如何将它们组合成复杂的逻辑电路。

实验的第一步是理解给定逻辑的真值表，这是分析任何逻辑电路的基础。通过真值表，我能够清晰地看到不同输入组合下输出的状态，这对于后续的电路设计至关重要。接着，我利用卡诺图来简化逻辑表达式，这是一种非常有效的工具，可以帮助我减少电路中的门数量，从而降低复杂性和成本

在设计过程中，我特别关注了如何使用或非门和非门来替代传统的与门、或门和非门。这不仅是一种技术上的挑战，也是对现有知识的一种扩展。通过这个过程，我学会了如何灵活运用不同的逻辑门来实现相同的功能，这增强了我的问题解决能力。

最终，我成功地将设计转化为实际的逻辑图，并进行了功能测试。这个过程不仅加深了我对数字逻辑的理解，也提高了我的实践技能。通过这次实验，我更加确信，理论知识与实践操作的结合是掌握任何技术的关键。