实验二 组合电路设计实验报告

姓名 **学号**

班级 **手机**

1. 实验题目

设计一个组合电路，输入一个 4 位的数字，输出一个 3 位的二进制数字，且输出数字的值近似等于输入数字值的平方根。例如，如果平方根的值等于 3.5 或者更大的值，则四舍五入记为 4。如果平方根的值小于 3.5 大于等于 2.5，则记为 3。

2. 实验约束

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

3. 电路设计

a) 规范化

输入：0 到 15 的二进制编码表示，分别为 0000-1111

输出：输入数字的平方根的 3 位二进制编码近似表示

电路的行为：对输入的 4 位二进制编码数字，输出其平方根的 3 位二进制数字

b) 形式化

记输入为$X_3X_2X_1X_0$，输出为ABC，采用真值表来对电路行为进行建模。

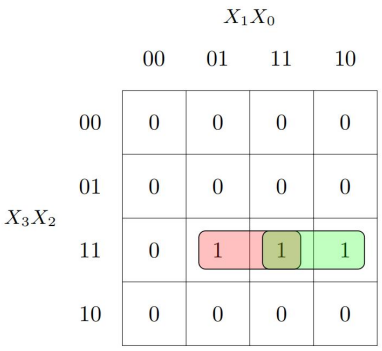
表 1 真值表

输入				输出		
X_3	X_2	X_1	X_0	A	B	C
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	0
0	1	0	1	0	1	0

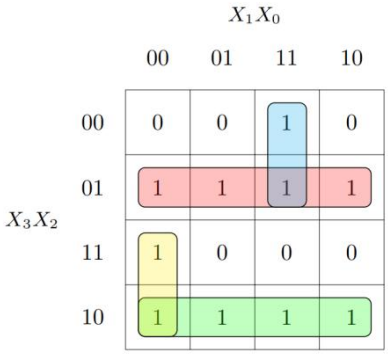
0	1	1	0	0	1	0
0	1	1	1	0	1	1
1	0	0	0	0	1	1
1	0	0	1	0	1	1
1	0	1	0	0	1	1
1	0	1	1	0	1	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0

c) 优化

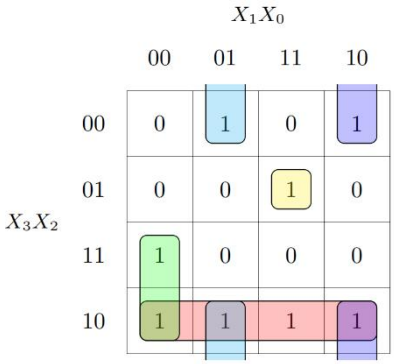
根据表 1 绘制出A、B、C的卡诺图，化简结果如图 1 所示。



(a) $A = X_3X_2X_0 + X_3X_2X_1$



(b) $B = X_3\overline{X_2} + \overline{X_3}X_2 + X_3\overline{X_1X_0} + \overline{X_3}X_1X_0$



(c) $C = X_3\overline{X_2} + X_3\overline{X_1X_0} + \overline{X_2X_1}X_0 + \overline{X_2}X_1\overline{X_0} + \overline{X_3}X_2X_1X_0$

图 1 卡诺图化简过程

提取公因子，整理电路，可得

$$T = \overline{X_2} + \overline{X_1X_0}$$
$$A = X_3X_2(X_0 + X_1)$$
$$B = X_3T + \overline{X_3}(X_2 + X_1X_0)$$

$$C = X_3T + \overline{X_2}\left(\overline{X_1}X_0 + X_1\overline{X_0}\right) + \overline{X_3}X_2X_1X_0$$

d) 工艺映射

工艺映射前的电路如下：

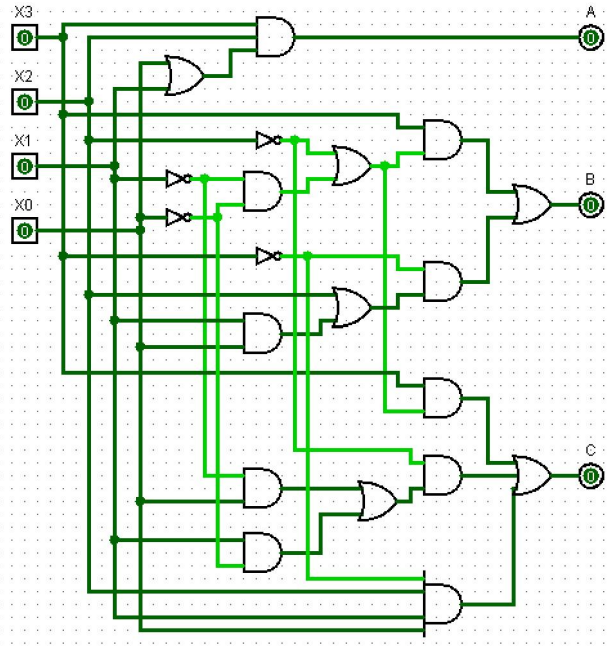


图 2 工艺映射前的电路

工艺映射后：

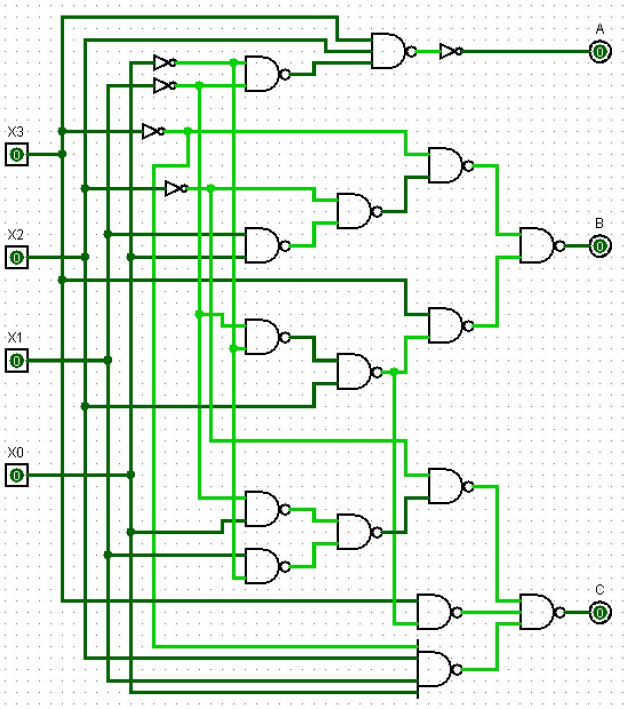


图 3 工艺映射后的电路

4. 电路实现

采用 Verilog 语言对电路进行结构化描述实现如下：

```
`timescale 1ns / 1ps

module calculate(X,A,B,C);
    //输入：0 到 15 的二进制表示，对应 X3X2X1X0
    input wire[3:0] X;
    //输出：输入数字平方根的 3 位二进制编码近似表示，对应 ABC
    output wire A,B,C;

    wire
    X3_n,X2_n,X1_n,X0_n,//输入数字的非
    A_1,A_2,//计算 A 的值所产生的中间变量
    B_1,B_2,B_3,B_4,B_5,B_6,
    C_1,C_2,C_3,C_4,C_5,C_6;

    //根据工艺映射后的电路，写出所需要用的非门和与非门
    not
    not3(X3_n,X[3]),//与 X[3]连接的非门
    not2(X2_n,X[2]),//与 X[2]连接的非门
    not1(X1_n,X[1]),//与 X[1]连接的非门
    not0(X0_n,X[0]),//与 X[0]连接的非门
    not3_o(A,A_2);//输出到 A 的非门

    nand
    nandA_1(A_1,X0_n,X1_n),
    nandA_2(A_2,X[3],X[2],A_1),

    nandB_1(B_1,X[1],X[0]),
    nandB_2(B_2,X2_n,B_1),
    nandB_3(B_3,X3_n,B_2),
    nandB_4(B_4,X1_n,X0_n),
    nandB_5(B_5,B_4,X[2]),
    nandB_6(B_6,X[3],B_5),
    nandB_o(B,B_3,B_6),//输出到 B 的与非门

    nandC_1(C_1,X1_n,X[0]),
    nandC_2(C_2,X[1],X0_n),
    nandC_3(C_3,C_1,C_2),
```

```
    nandC_4(C_4,X2_n,C_3),
    nandC_5(C_5,X[3],B_5),
    nandC_6(C_6,X3_n,X[2],X[1],X[0]),
    nandC_7(C,C_4,C_5,C_6);//输出到 C 的与非门

endmodule
```

---

5. 电路验证

a) TestBench

通过编写 Verilog TestBench，对实现的电路模块进行功能验证。

```
`timescale 1ns / 1ps

module testbench();
    reg[3:0] x;
    wire a,b,c;

    initial
    begin
        x = 4'b0000;     //输入 0，输出 0
        #10 x = 4'b0001; //输入 1，输出 1
        #10 x = 4'b0010; //输入 2，输出 1
        #10 x = 4'b0011; //输入 3，输出 2
        #10 x = 4'b0100; //输入 4，输出 2
        #10 x = 4'b0101; //输入 5，输出 2
        #10 x = 4'b0110; //输入 6，输出 2
        #10 x = 4'b0111; //输入 7，输出 3
        #10 x = 4'b1000; //输入 8，输出 3
        #10 x = 4'b1001; //输入 9，输出 3
        #10 x = 4'b1010; //输入 10，输出 3
        #10 x = 4'b1011; //输入 11，输出 3
        #10 x = 4'b1100; //输入 12，输出 3
        #10 x = 4'b1101; //输入 13，输出 4
        #10 x = 4'b1110; //输入 14，输出 4
        #10 x = 4'b1111; //输入 15，输出 4
    end
```

```
    calculate cal(
        .X(x),
        .A(a),
        .B(b),
        .C(c)
    );

Endmodule
```

---

b) 仿真结果

利用上一小节的 Verilog TestBench 进行 Vivado 仿真，得到 Wave 如下：

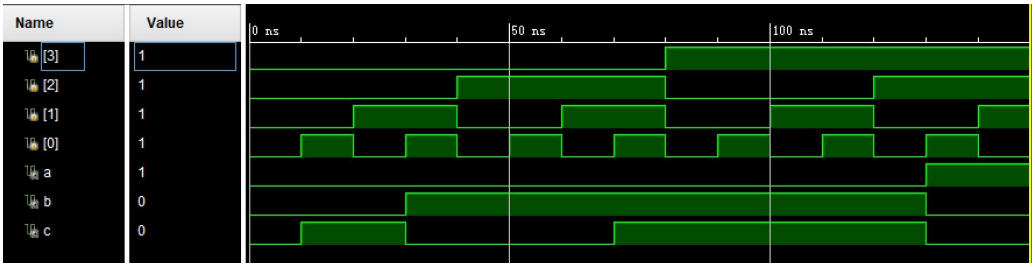


图 4 电路的仿真结果

输入为 0 到 15 的二进制表示 X[3]X[2]X[1]X[0]，输出为输入平方根的二进制近似表示 ABC，对应图中 Name 栏的各项。对 Wave 进行分析解释如下：

输入 0：二进制输入 0000，二进制输出 000，对应十进制数字 0；
输入 1：二进制输入 0001，二进制输出 001，对应十进制数字 1；
输入 2：二进制输入 0010，二进制输出 001，对应十进制数字 1；
输入 3：二进制输入 0011，二进制输出 010，对应十进制数字 2；
输入 4：二进制输入 0100，二进制输出 010，对应十进制数字 2；
输入 5：二进制输入 0101，二进制输出 010，对应十进制数字 2；
输入 6：二进制输入 0110，二进制输出 010，对应十进制数字 2；
输入 7：二进制输入 0111，二进制输出 011，对应十进制数字 3；
输入 8：二进制输入 1000，二进制输出 011，对应十进制数字 3；
输入 9：二进制输入 1001，二进制输出 011，对应十进制数字 3；
输入 10：二进制输入 1010，二进制输出 011，对应十进制数字 3；
输入 11：二进制输入 1011，二进制输出 011，对应十进制数字 3；
输入 12：二进制输入 1100，二进制输出 011，对应十进制数字 3；
输入 13：二进制输入 1101，二进制输出 100，对应十进制数字 4；
输入 14：二进制输入 1110，二进制输出 100，对应十进制数字 4；
输入 15：二进制输入 1111，二进制输出 100，对应十进制数字 4；

可以发现，输出是输入平方根的近似表示，与期望值相同，电路设计无误。

6. 实验心得

在本次实验中，我借助 Vivado 软件第一次完整地完成了组合电路设计的全过程，从规范化、形式化、优化、工艺映射到电路的实现与验证，圆满地完成了本次实验。

通过本次实验，我掌握了运用 Verilog 语言对电路进行结构化描述的基本方式。在参照课本例题与乐学视频后，代码的实现本身较为容易。

然而，我在第一次进行仿真时得到了错误的结果。经过检查发现，在设计文件 `calculate.v` 的文件中，我采用了 `input wire[3:0] X` 定义输入，却在仿真文件 `testbench.v` 的文件中用 `reg[0:3] x` 模拟输入，两种方式的二进制表示分别为 `X[3]X[2]X[1]X[0]`和 `x[0]x[1]x[2]x[3]`。它们虽然具有相同的位宽，但所代表的意义不同，不应该混用。在调整之后，我得到了预期的答案。

同时，在阅读书上例题时，我不太能理解仿真文件中形如 `4'b0000` 表达式的含义。在查阅资料后，我发现该表达式各项分别对应<位宽><进制><数字>，即长度为 4 的二进制表示 0000。

在撰写实验报告时，我发现卡诺图的优化过程无法较好地表示。通过查阅资料，我发现 LaTeX 中的 `karnaugh-map` 包提供了对卡诺图绘制的支持，在阅读相关说明后，我成功实现了卡诺图的绘制。

在工艺映射的步骤中，我通过 `logcism` 软件对映射前和映射后的电路进行了绘制，并通过软件提供的真值表验证了电路的正确性。

本次实验是一个成功的开始，为后面的两次实验打下了良好的基础。