**实验二 组合电路设计实验报告**

**姓名**：xxx  **学号**：xxx

**班级**：xxx **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，且输出的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。

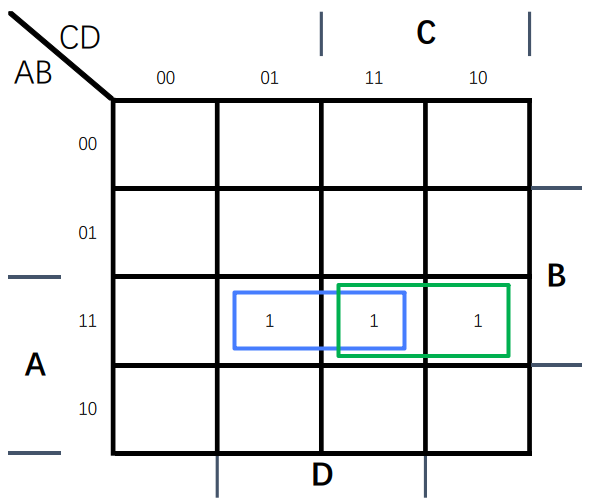
1. **实验约束**

* 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。

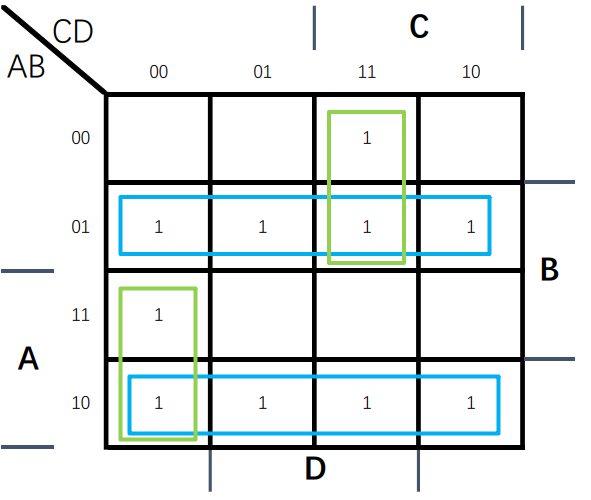
1. **电路设计**
   1. **规范化**
      1. 电路的行为：由4位二进制数得到其平方根四舍五入后的3位二进制数表示
      2. 输入：0到15的4位BCD码表示，分别为0000~1111
      3. 输出：0到15平方根的3位BCD码表示，范围为000~100
   2. **形式化**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **输入** | | | | **输出** | | |
| A | B | C | D | X | Y | Z |
| 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 |

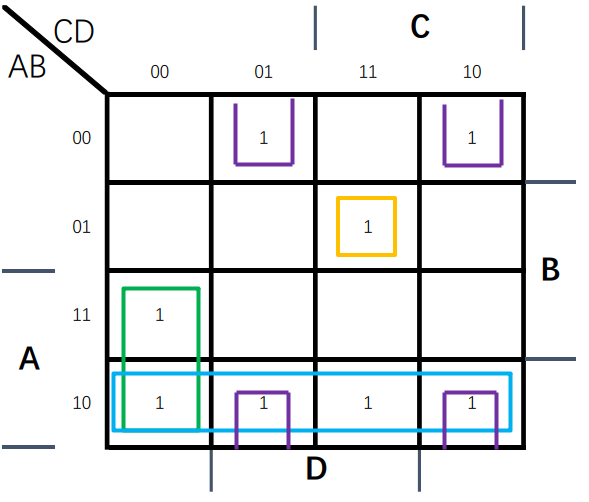
* 1. **优化**
     1. **卡诺图优化**
        1. **X = A·B·C + A·B·D**

****

* + - 1. **Y = A’·B + A·B’ + A’·C·D + A·C’·D’**

****

* + - 1. **Z = A’·B·C·D + A·C’·D’ + B’·C’·D + B’·C·D’ + A·B’**

****

* + 1. **提取公因子进一步优化**

T1 = C + D

T2 = A’·B

T3 = A·B’

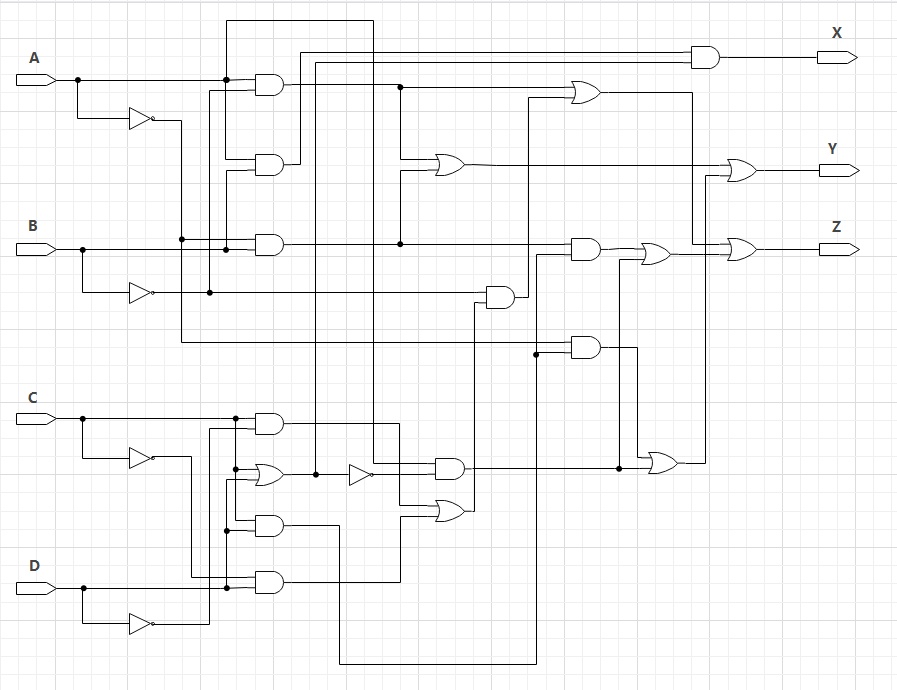
T4 = C·D

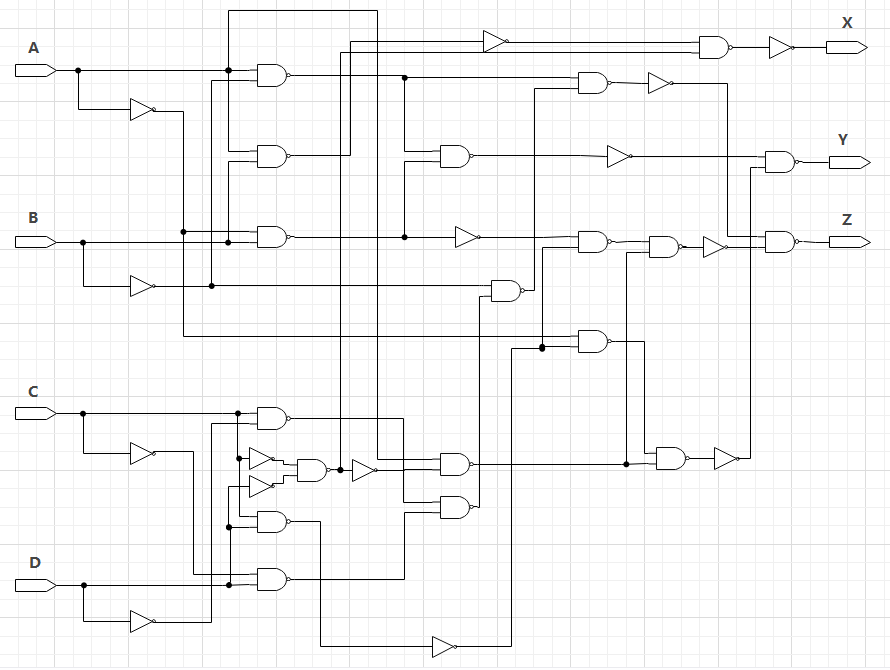
T5 = A·T1’

X = A·B·T1

Y = T2+ T3 + A’·T4 + T5

Z = T2·T4+ T5+ B’·(C’·D + C·D’) + T3

* 1. **工艺映射**
     1. **工艺映射前**
     2. **工艺映射后**



1. **电路实现**

`timescale 1ns / 1ps

module sqrt(i, o);

input [3:0] i; // 4位BCD码作为输入

output [2:0] o; // 3位BCD码作为输出

// 将输入的4位赋给A、B、C、D；将X、Y、Z赋给输出的3位

wire A, B, C, D, X, Y, Z, T1;

assign A = i[3], B = i[2], C = i[1], D = i[0];

assign o[2] = X, o[1] = Y, o[0] = Z;

// A\_, B\_, C\_, D\_, T1\_分别是A, B, C, D, T1取反

wire A\_, B\_, C\_, D\_, T1\_;

not notA(A\_, A), notB(B\_, B), notC(C\_, C), notD(D\_, D), notT1(T1\_, T1);

// T1 = C + D

or orT1(T1, C, D);

// X = A·B·T1

and andX(X, A, B, T1);

// T2 = A'·B

// T3 = A·B'

// T4 = C·D

// T5 = A·T1'

wire T2, T3, T4, A\_C, C\_D, CD\_;

and andT2(T2, A\_, B);

and andT3(T3, A, B\_);

and andT4(T4, C, D);

and andT5(T5, A, T1\_);

and CandD(CD\_, C, D\_);

and C\_andD(C\_D, C\_, D);

wire A\_T4;

and A\_andT4(A\_T4, A\_, T4);

// Y = T2+ T3 + A'·T4 + T5

or orY(Y, T2, T3, A\_T4, T5);

// t1 = C·D' + C'·D

wire T2T4, t1, B\_t1;

or ort1(t1, CD\_, C\_D);

and B\_andT(B\_t1, t1, B\_);

and T2andT4(T2T4, T2, T4);

// Z = T2·T4+ T5+ B'·t1 + T3

or orZ(Z, T2T4, T5, B\_t1, T3);

endmodule

1. **电路验证**
   1. **TestBench**

`timescale 1ns / 1ps

module testbench();

reg [3:0] moduleInput = 4'b0; // 作为模块输入

wire [2:0] moduleOutput; // 作为模块输出

integer i;

initial begin

// 每隔10ns将输入加一

for(i = 0; i < 16; i = i + 1) begin

# 10;

moduleInput = moduleInput + 1'b1;

end

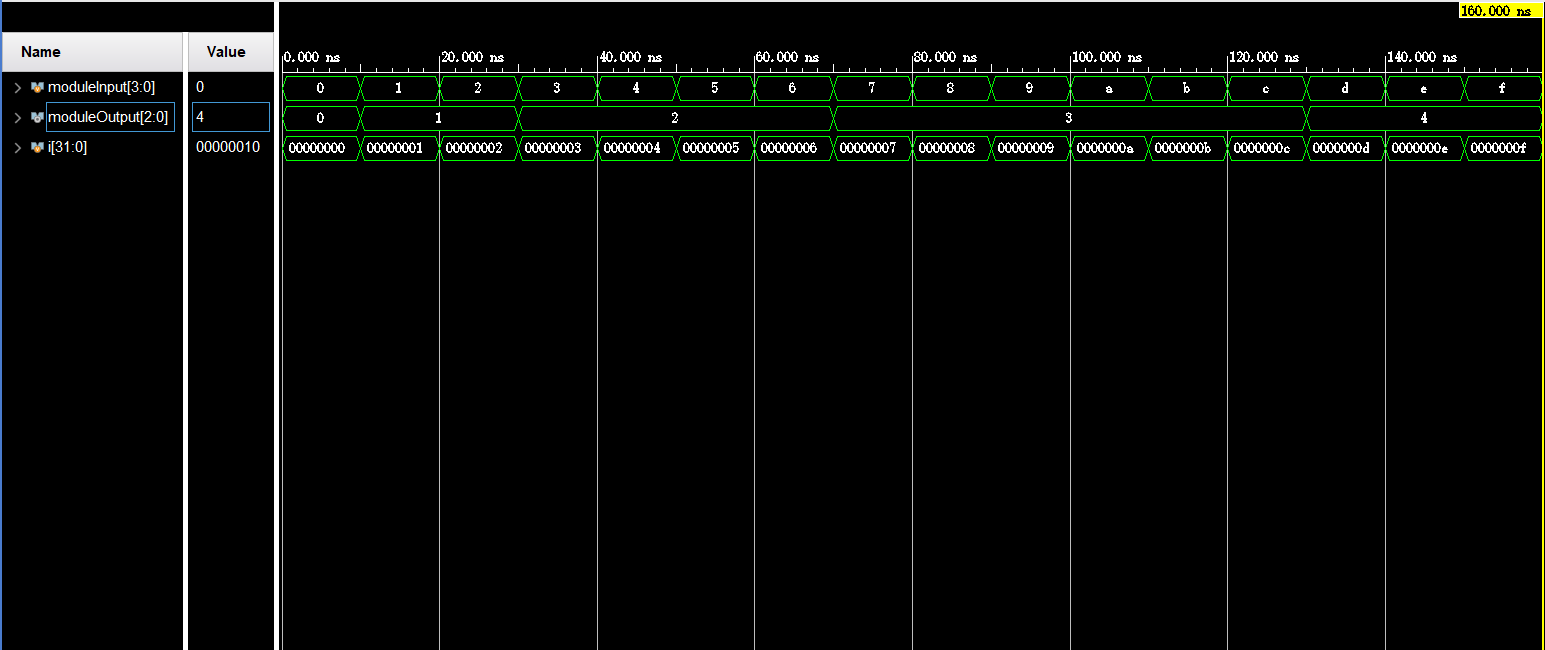
// 所有输入都验证完后停止仿真

$stop;

end

sqrt sqrt(moduleInput, moduleOutput);

endmodule

* 1. **仿真结果**

通过该仿真图可以轻易看出，本实验所设计的电路完美实现了题目的要求。在输入为0时输出0，在输入为1~2时输出1，在输入为3~6时输出2，在输入为7~12时输出3，在输入为13~15时输出4。

1. **实验心得**

本次实验让我初次掌握了Verilog在组合电路中使用方法，掌握了组合逻辑电路从设计到验证的全过程。在做实验的过程中，我又捡起了课上讲过的易错点，如卡诺图要贯穿2的幂次方个方格等。在进行提取公因式优化时，由于看错了要优化的式子，导致做了许多无用功，最后在仿真时才找到了错误，浪费了许多时间。总之，本次实验圆满完成了既定任务。