# 实验一：一位全加器的设计与实现

【**实验要求**】：

1. 利用“与门”、“或门”、“非门”设计并实现1位全加器电路。

【**实验目的**】

1. 掌握组合逻辑电路的设计方法；
2. 熟悉Vivado2014集成开发环境和Verilog编程语言；
3. 掌握1位全加器电路的设计与实现。

【**实验环境**】

* FPGA虚拟仿真平台。
* Vivado2014 集成开发环境。
* Verilog编程语言。

【**实验原理**】

功能描述

[全加器](https://baike.baidu.com/item/%E5%85%A8%E5%8A%A0%E5%99%A8/9791810?fromModule=lemma_inlink)是能够计算低位进位的[二进制](https://baike.baidu.com/item/%E4%BA%8C%E8%BF%9B%E5%88%B6/361457?fromModule=lemma_inlink)加法电路。与[半加器](https://baike.baidu.com/item/%E5%8D%8A%E5%8A%A0%E5%99%A8/7653973?fromModule=lemma_inlink)相比,全加器不只考虑本位计算结果是否有进位,也考虑上一位对本位的进位,可以把多个一位全加器[级联](https://baike.baidu.com/item/%E7%BA%A7%E8%81%94/9704773?fromModule=lemma_inlink)后做成多位全加器.

真值表

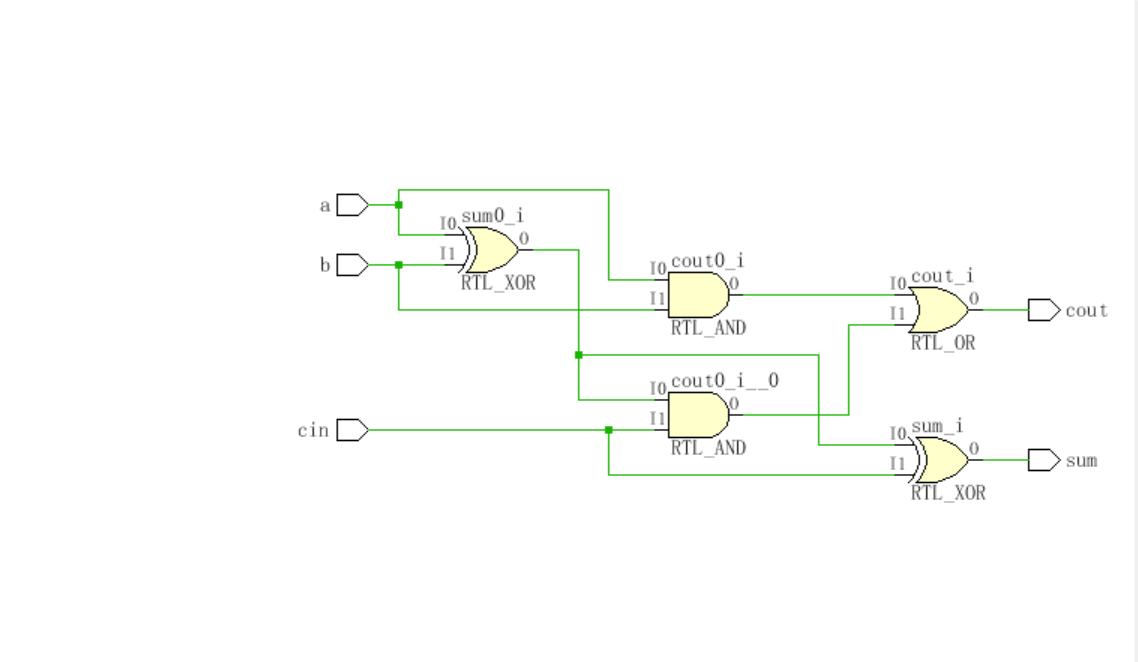
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | Cout | Sum |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

逻辑方程

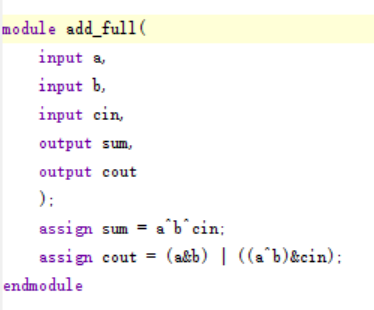
Sum=A⊕B⊕Cin

Cout=Cin(A⊕B)+AB

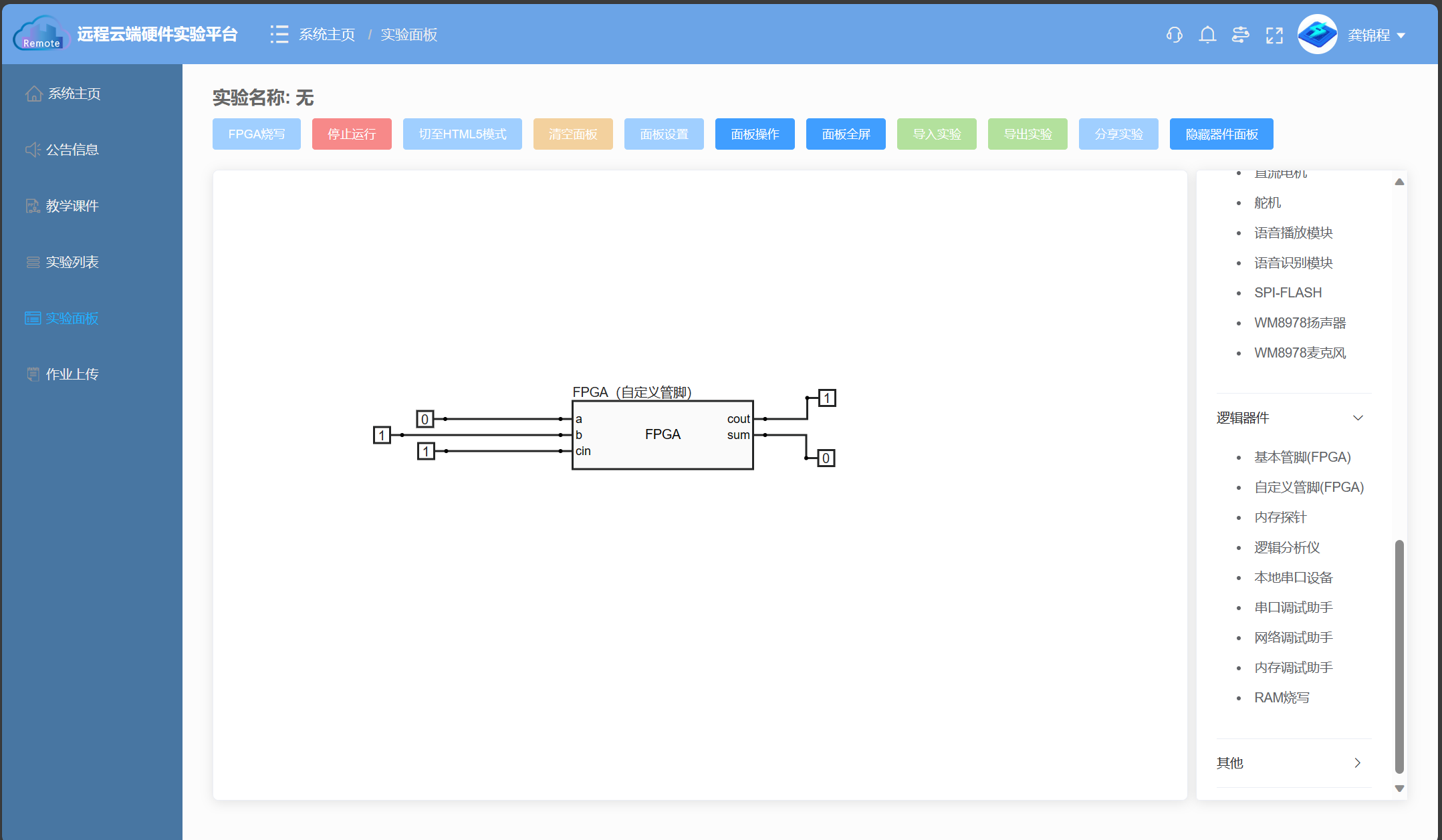
电路图

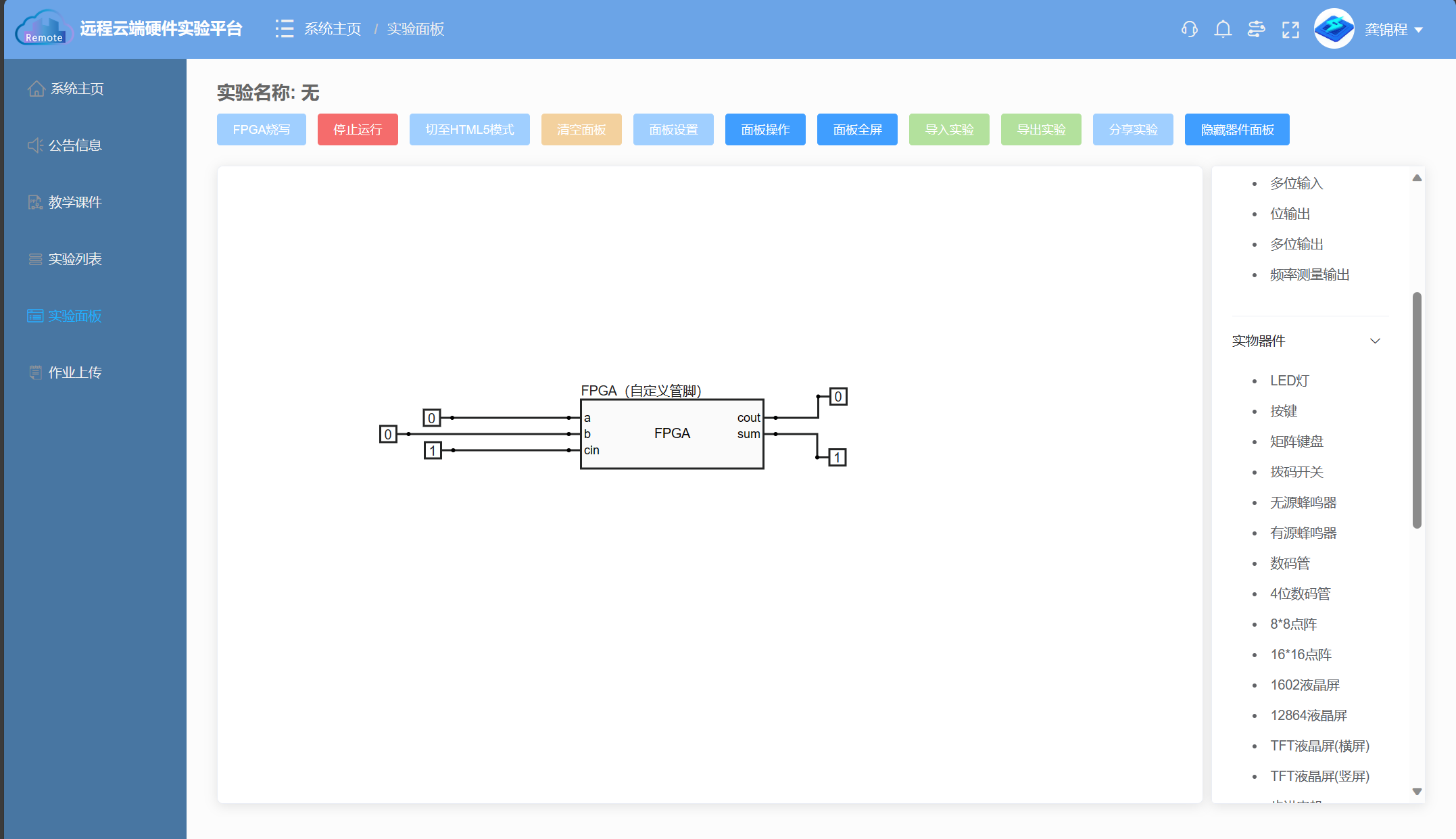


Verilog代码实现

****

实验结果与仿真结果



****

# 实验二：一位BCD码转余三码

【**实验要求**】：

1. 利用“与门”、“或门”、“非门”设计并实现BCD码转余三码的电路。

【**实验目的**】

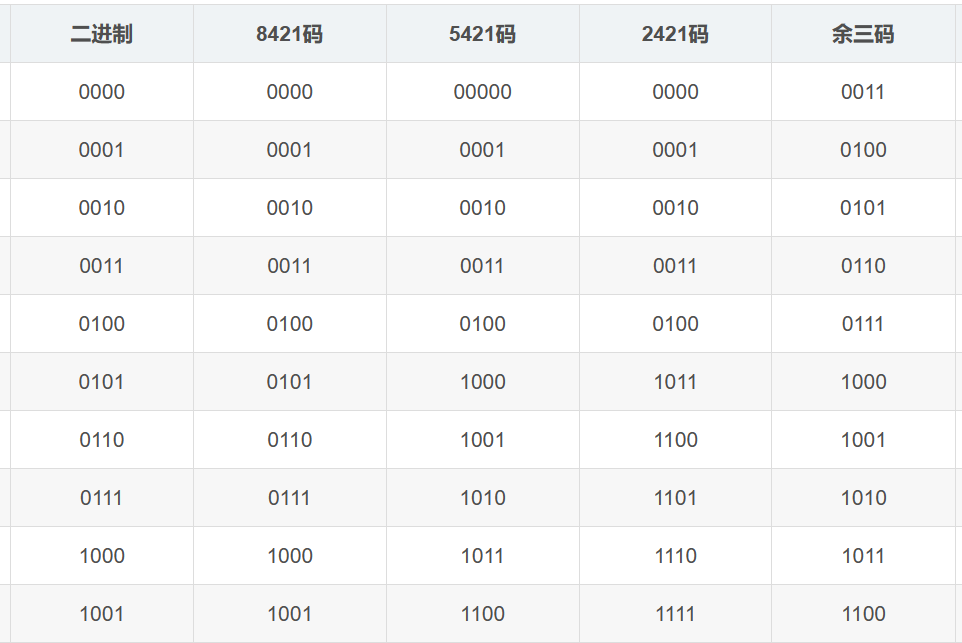
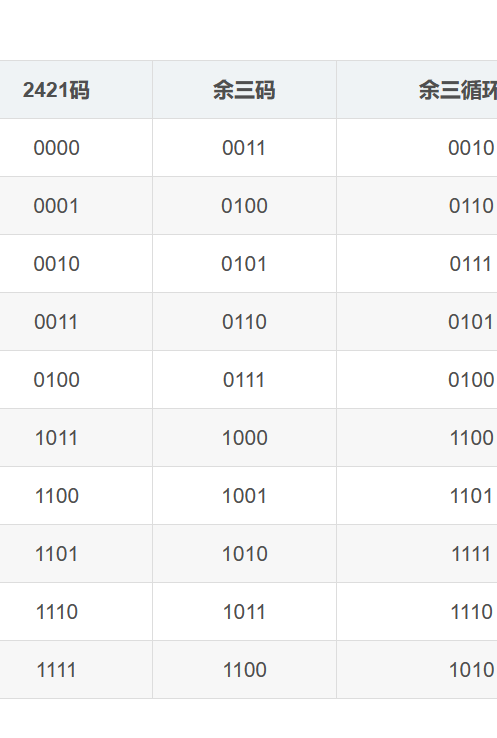
1. 掌握组合逻辑电路的设计方法；
2. 熟悉Vivado2014集成开发环境和Verilog编程语言；
3. 掌握BCD码转余三码电路的设计与实现。

【**实验环境**】

* FPGA虚拟仿真平台。
* Vivado2014 集成开发环境。
* Verilog编程语言。

【**实验原理**】

功能描述

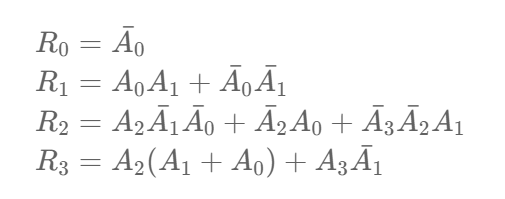
 

以上为BCD码与余三码，设置的逻辑电路就是实现从BCD转余三码，使用与非或门。

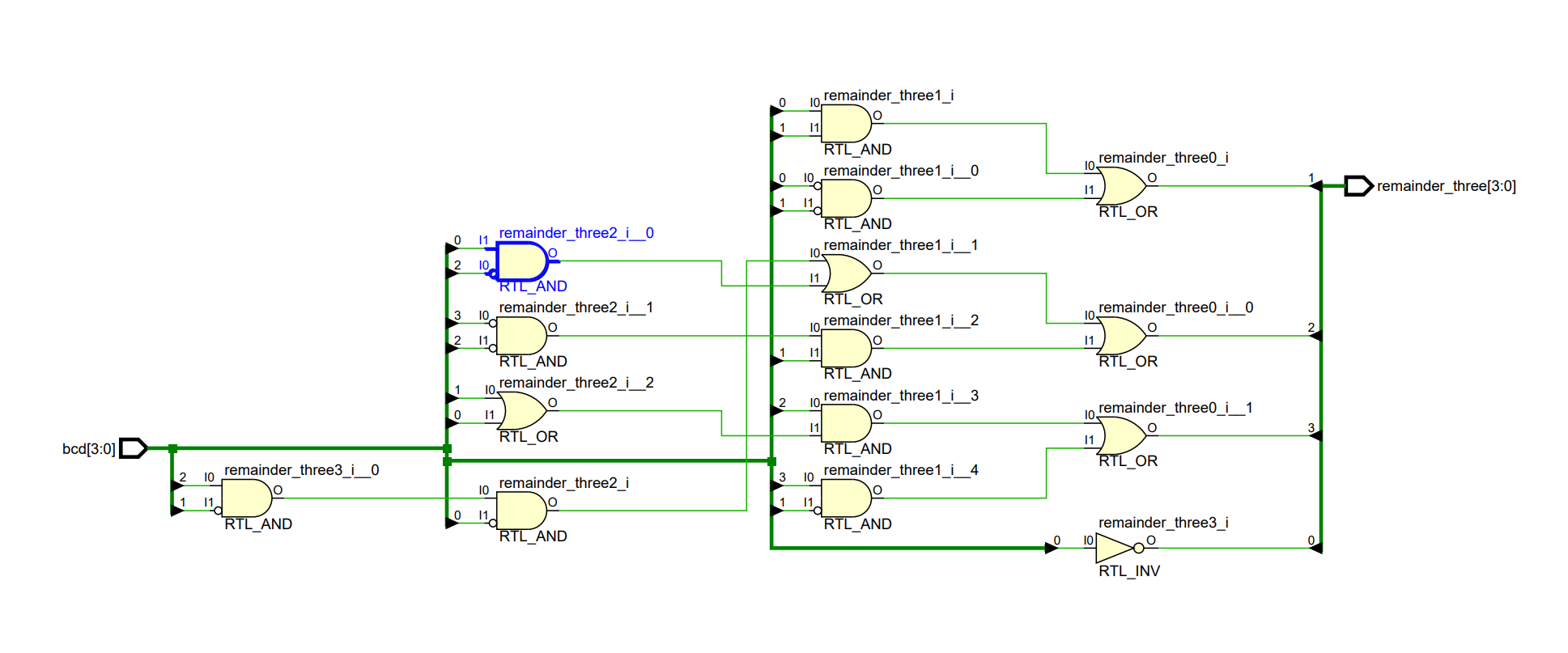
真值表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BCD | | | | 余三码 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

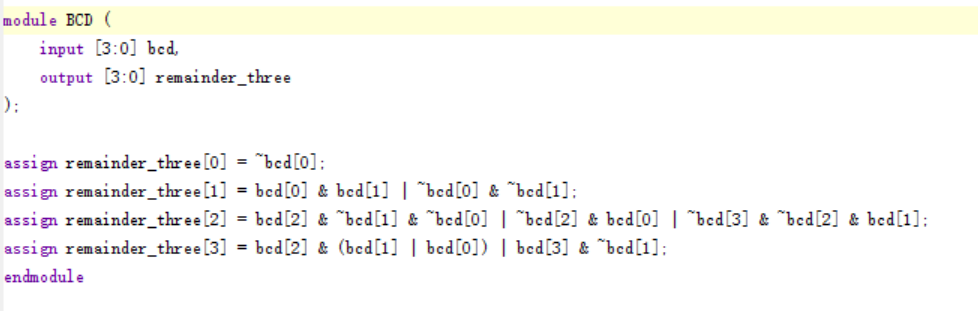
逻辑方程



电路图



Verilog代码实现



实验结果与仿真结果

