

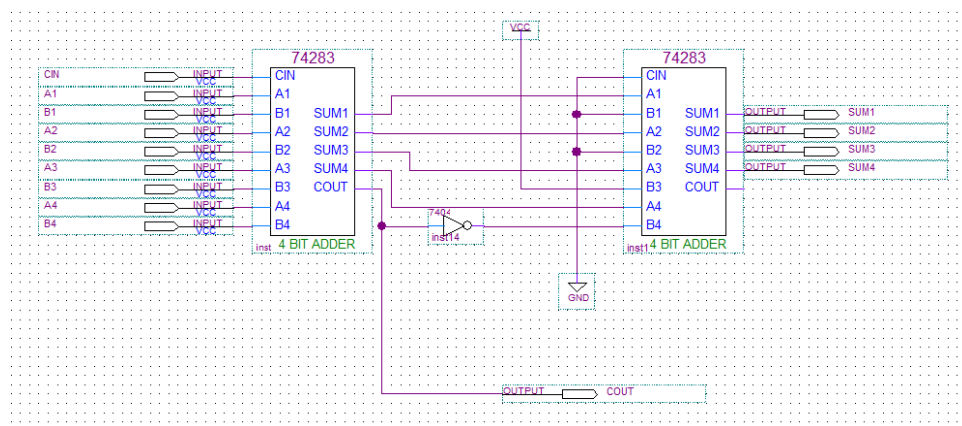
# 山东大学 计算机科学与技术 学院

## 数字逻辑 课程实验报告

学号：202000130198	姓名：隋春雨	班级：20.4
实验题目：实验 3 十进制数加法器		
实验学时：4	实验日期：2021 年 11 月 12 日	
<b>实验目的：</b> (1) 学习组合电路的设计方法； (2) 了解余三码的构成和十进制数加法器的构成和工作原理； (3) 熟悉 EDA 工具软件的使用方法。		
<b>硬件环境：</b> (1) 操作系统为 WINDOWS 7 的台式机一台； (2) 数字逻辑与计算机组成原理实验系统一台； (3) 与门电路和或门电路若干。		
<b>软件环境：</b> Quartus II 8.1 Web		
<b>实验步骤与内容：</b> (包括设计的逻辑电路，采用的逻辑门，或者是前期采用基本逻辑电路实现的符合逻辑，写出逻辑表达式，结果预期（采用什么方式展示，如采用那几个发光二极管等），以及最终实现的结果（是否与预期的结果一致，若不一致，是什么问题造成的，经过哪些改进，达到了最终的正确结果）) <b>【实验一】</b> (1) 实验原理：首先我们应该先了解什么叫做余 3 码。余 3 码是在建立在 8421 码的基础上的一种进制表示，8421 码表示从 0-9 的十进制数字，然后对应的一个十进制数字的 8421 表示，加上 3 就是他的余 3 码。所以我们的不难看出，余 3 码如果认为这只是一个简单的 01 序列的话，他的表示范围就是从 3-12		
Decimal	8421BCD	余3码
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100
也就是说：(0000~0010) 和 (1101~1111) 是非法码（即在余 3 码中不存在）		
(2) 实验要求：本实验要求在掌握四位并行加法器 74283 使用方法和理解余 3 码运算法则的基础上，利用 4 位二进制并行加法器 74283 和六非门 7404 设计一个用余三码编码的 1 位十进制数加法器，并通过发光二极管显示加法结果。余三码编码的 1 位十进制数加法器原理图如图 3.6 所示，其中 A4-A1 和 B4-B1		

为两个余三码编码表示的加数，CIN 为低位来的进位，SUM4-SUM1 为余三码编码表示的和数，COUT 为向高位的进位。

(3) 实验电路图：



(4) 余 3 码在现实生活中的应用：首先我们知道，余三码是由 8421 码加上 0011 后形成的无权码，所以，十进制 0 和 9, 1 和 8, 2 和 7, 3 和 6, 4 和 5 对应的余三码互为反码，这类似于我们之前学到的补码的运算，也是建立在互补上的数学技巧。

(5) 余 3 码加法原则：

两个十进制数的余 3 码相加，按逢二进一原则进行。

若其和没有进位，则减 3 (1101) 矫正。

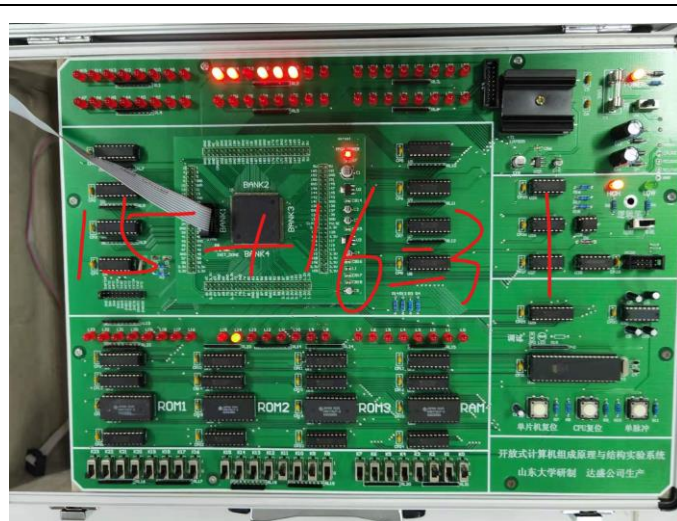
若其和有进位，则加 3 (0011) 矫正。

余三码编码的 1 位十进制数加法器原理图

Ci	A4	A3	A2	A1	B4	B3	B2	B1	C0	S4	S3	S2	S1
0	0	1	1	0	1	0	0	0	0	0	0	1	1
0	1	0	0	1	1	0	0	0	1	0	1	1	1
0	0	0	1	1	0	0	1	1	0	0	0	1	1

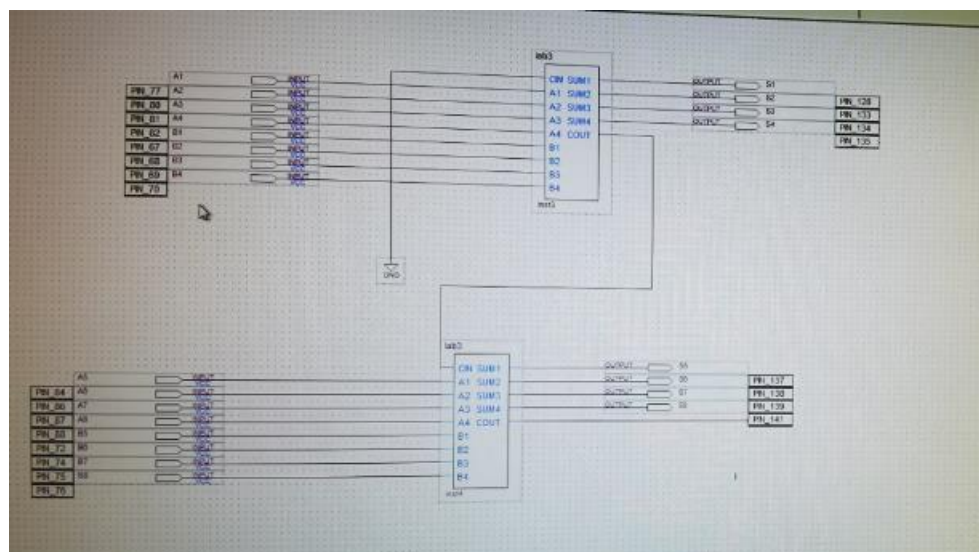
测试用例

(6) 验证：



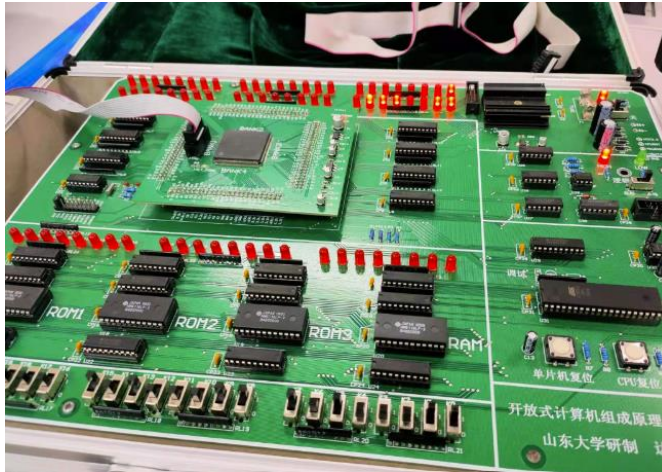
加法器的调试：根据余三码的运算规则，使用输入开关在余三码编码的 1 位十进制数加法器的输入端赋值，观察输出和数 SUM 及向高位进位 COUT 的值是否正确，并通过发光二极管显示相加结果。

(7) 实验二并行加法器原理图：



(8) 并行加法器的优点：首先我们知道：并行加法器通过多根引线的使用，使得原来的高位需要等待低位进位的时间大大缩短了，现在只需要等待两个并行加法器部件之间的进位就行了，但是同样的，这个有一个很大的问题：如果这是一个很复杂的原件，那它多出来的引线数目将会是一个很大的数量，达到了  $O(n^2)$  的级别，解决它的方法也比较容易想到，就是串行和并行结合。

(9) 测试结果：70+70=140，符合实验预期



### 结论分析与体会：

- (1) 余 3 码是在建立在 8421 码的基础上的一种进制表示,8421 码表示从 0-9 的十进制数字, 然后对应的一个十进制数字的 8421 表示, 加上 3 就是他的余 3 码。所以我们不难看出, 余 3 码如果认为这只是一个简单的 01 序列的话, 他的表示范围就是从 3-12

Decimal	8421BCD	余3码
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

也就是说: (0000~0010) 和 (1101~1111) 是非法码 (即在余 3 码中不存在)

- (2) 加深了在计算机的世界中, 取反的运算占据的重要地位, 无论是我们补码的使用, 还是余 3 码的使用, 都是我们取到了一个固定的值作为一个上限, 然后取上限与其相减, 这也是余 3 码中 2 进制与 10 进制转换的桥梁, 也是补码中负数的计算规则
- (3) 加深了对并行加法器的理解, 并行加法器其实是一种用空间换时间的设备, : 并行加法器通过多根引线的使用, 使得原来的高位需要等待低位进位的时间大大缩短了, 现在只需要等待两个并行加法器部件之间的进位就行了, 但是同样的, 这个有一个很大的问题: 如果这是一个很复杂的原件, 那它多出来的引线数目将会是一个很大的数量, 达到了  $O(n^2)$  的级别, 解决它的方法也比较容易想到, 就是串行和并行结合
- (4) 学会了利用进位数和非门的组合, 使得线的连接比较简洁。也理解了 8421

码如何进行加减运算，如果产生进位，那么+3；如果不产生，则-3

- (5) 我们在实验的过程中，应该结合面向对象的思想，比如说全加器那个实验，如果一个一个重新设置，那很麻烦，而且有一个非常大的弊端就是，如果你最后发现自己的思路错了，那这个实验就白做了，基本需要推倒了重来，在本次实验中，我们组也用到了面向对象，将元件封装起来。