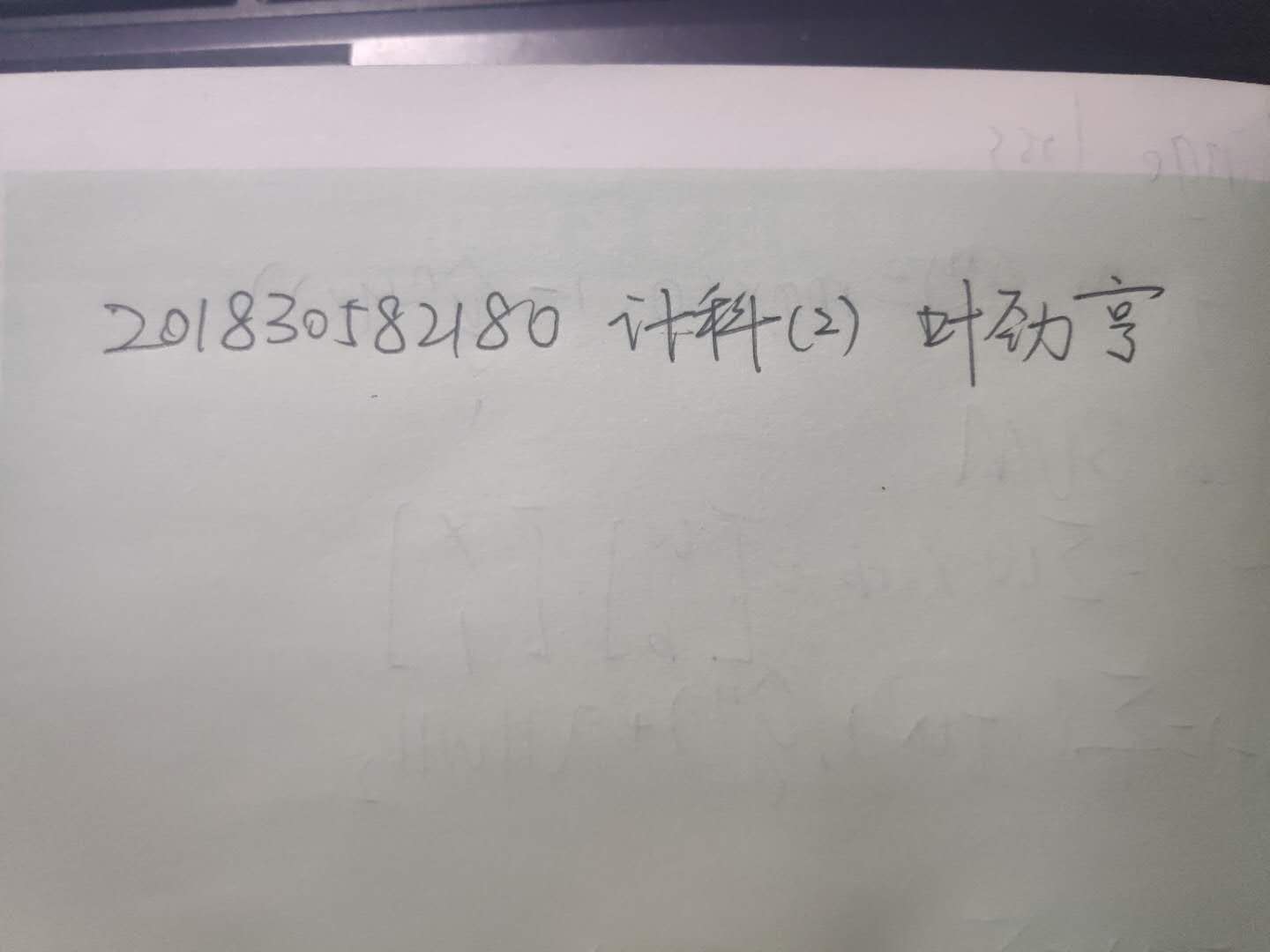
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 班级 | 2018级2班 | 姓名 | 叶劲亨 | 学号 | 201830582180 |
| 题目 | **2.2 进位加法器实验** | | | | |
|  |  | | | | |

**一、实验目的**

1、了解半加和全加运算器的电路结构。

2、掌握串行进位加法器和并行进位加法器的原理及设计方法。

**二、实验内容**

1、设计拥有共同输入端的 4 位带符号位的串行加法器和并行加法器，并比较两者的运算

输出结果。

**三、实验器件**

1、2/3/4 与门(74LS08/11/21)、非门(74LS04)、或门(74LS32)、异或门(XOR)等逻辑门。

2、三态门（74LS244）、LED 指示灯及数码显示管。

**四、实验原理**

实验中的加法运算器电路如图 2-7 所示。本电路包含了输入单元（拨码开关）、输出单元（数码显示管）和两个四位全加器。输入单元如图 2-3 所示，跟上述实验相同：拨码开关用来设置输入数据，低电平有效的开关 SW\_BUS 控制输入总线 DIN\_[0..7]与数据总线BUS\_[0..7]是否连通。数据总线 BUS 同时为两个运算器提供相同的输入端：BUS\_[0..3]和BUS\_[4..7]。输出单元是一对数码显示管，绿色显示管是并行进位加法器的输出端，红色显示管是串行进位加法器的输出端。

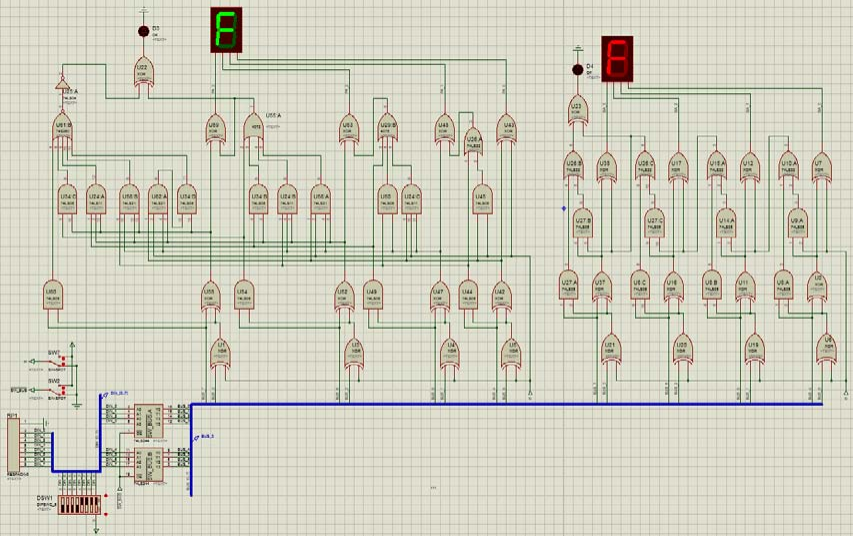


图 2-7 进位加法器电路图

加法器是执行二进制加减法运算的逻辑部件（减法可以通过补码相加实现），分为半加器和全加器（FA），不考虑低位的进位，只考虑两个二进制数相加，得到和以及向高位进位的加法器为半加器，如图 2-8 所示；而全加器(FA)是在半加器的基础上又考虑了低位过来的进位信号，如图 2-9 所示。

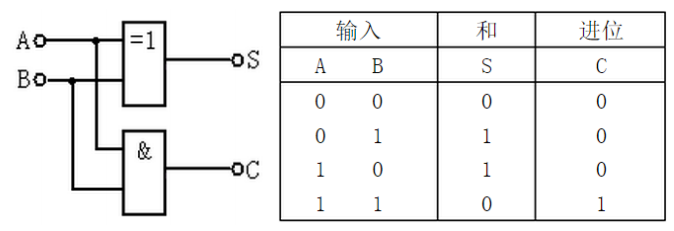


图 2-8 一位半加器逻辑电路图

半加器逻辑公式： Si = Ai⊕Bi 且 Ci+1 = Ai·Bi

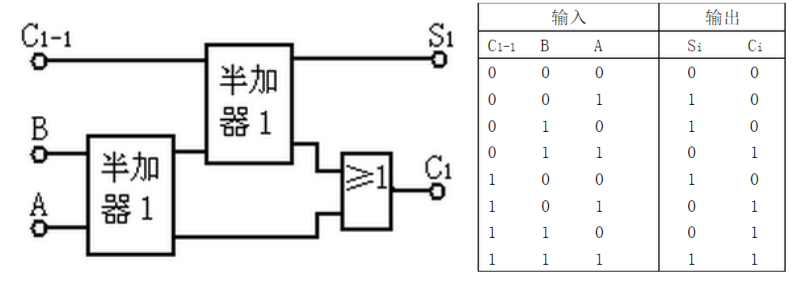


图 2-9 一位全加器（FA）逻辑电路图

全加器逻辑公式： Si = Ai⊕Bi⊕Ci 且 Ci+1=Ai·Bi + (Ai⊕Bi)·Ci

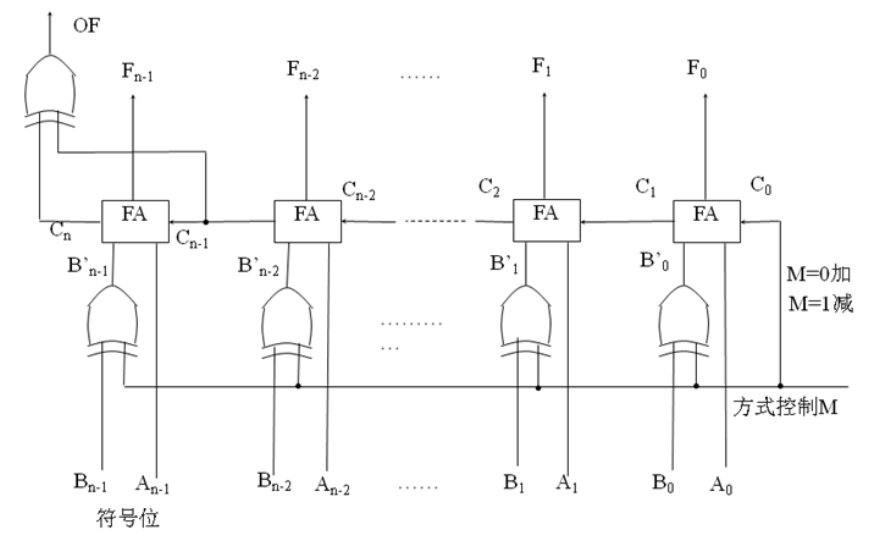
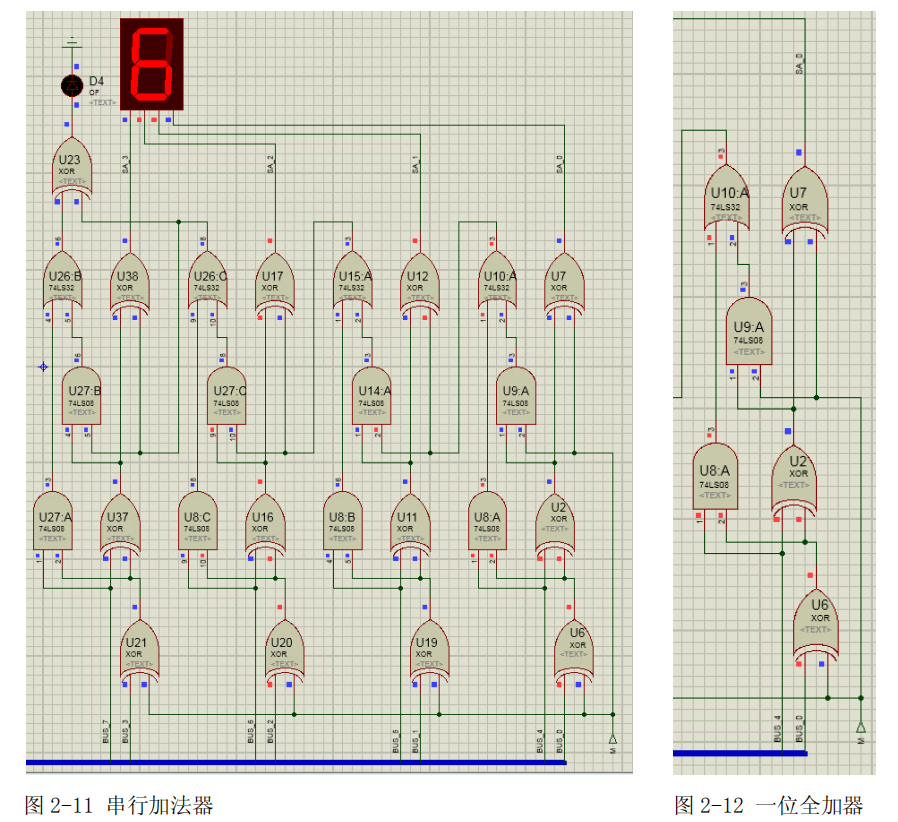


图 2-10 串行加法器（行波进位加法器）逻辑电路图本实验的串行加法器又称为行波进位加法器，如图 2-10 所示：由若干位全加器 FA 串行相连得到，其中低位 FA 的进位输出直接与相邻的高 FA 的进位输入相连。因为高位进位产生依赖于相邻低位 FA 的进位，所以各位运算依次完成，延迟较长，效率不高。图 2-11 所示是四位串行加法器仿真图，由四个图 2-12 所示的一位全加器串行形成，最高位是符号位，数字有效位是三位。M 端控制器件做加法(M=0)或减法(M=1)，因为“任意两数之差的补码等于被减数的补码与减数相反数的补码之和”，所以 M 端工作原理基于“[Y]补→[-Y]补”求补运算原则：取反再加 1。图 2-11 中执行的运算是：“101+001=110”。



本实验的并行加法器又称为超前进位加法器，如图 2-13 所示：也是由若干位全加器 FA组成，但是每一个 FA 所需的低位进位都不依赖其他 FA，而是根据最低进位 C0及各个位的加数、被加数即可同时计算所有进位 Cn。因此称之为“并行”加法器，其推导公式如下：

Ci+1=Ai·Bi + (Ai⊕Bi)·Ci令 Yn= An·Bn Xn= (An⊕Bn)，则有：

Cn+1=Yn+Xn·Cn 故 ：

C1 = Y0+X0·C0

C2 = Y1 + X1·C1 = Y1 + X1·Y0 + X1·X0·C0

C3 = Y2 + X2·C2 = Y2 + X2·Y1 + X2·X1·Y0 + X2·X1·X0·C0

C4 = Y3 + X3·C3 = Y3 + X3·Y2 + X3·X2·Y1 + X3·X2·X1·Y0 + X3·X2·X1·X0·C0

由图 2-13 中可以看出，每一个 FA 所需的低位进位都只跟 An、Bn 和 C0相关，与相邻 FA产生的进位无关。从右到左，随着位数越高，进位 Cn 生成电路越复杂。图 2-13 中执行的运算是：“101+011=1000”；因为进位影响到符号位改变，所以导致溢出，数码管旁边的 OF 指示灯点亮。

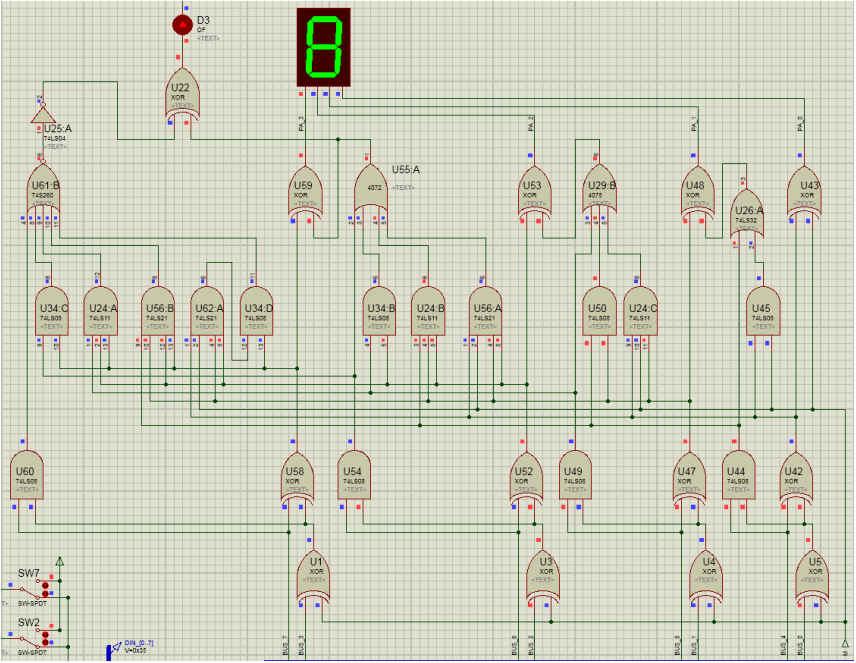


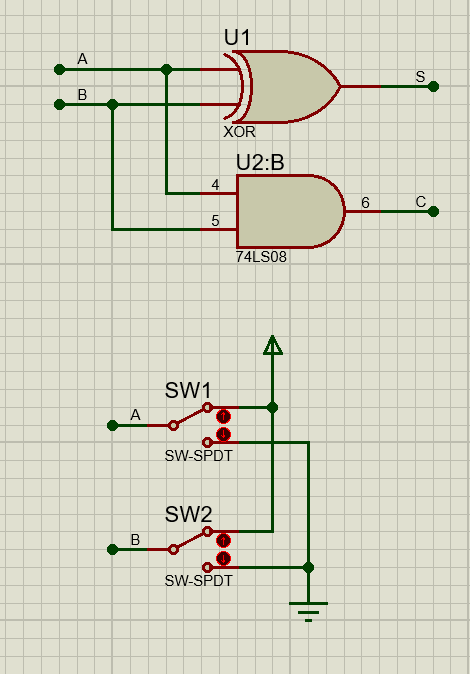
图 2-13 并行加法器

**五、实验步骤**

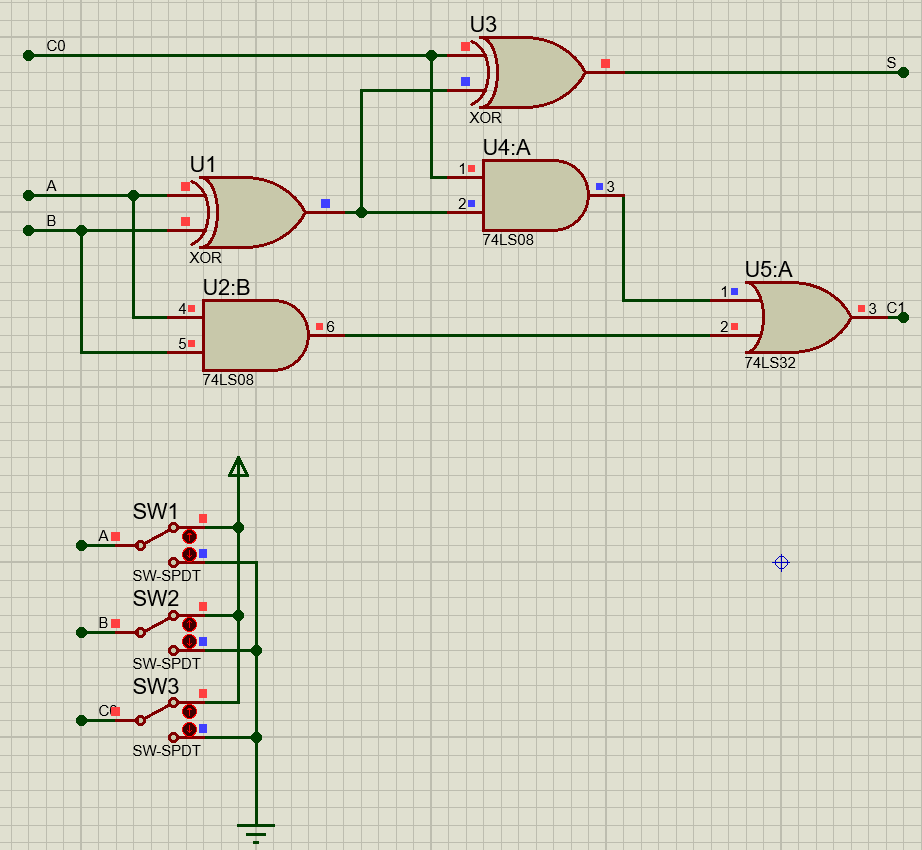
● 请制作半加器和全加器的电路，并验证其输入输出是否与图 2-8 和图 2-9 一致。

解答：

半加器电路如下图，经验证，输入输出与图 2-8 一致



全加器电路如下图，经验证，输入输出与图 2-9 一致



● 令 BUS\_[7..4]=0101，BUS\_[3..0]=0010，M=0，写出该运算的真值表达式，记录并比较串行与并行加法器的运算结果，是否有溢出？如果改为 BUS\_[3..0]=0011，结果如何呢？

解答：令 BUS\_[7..4]=0101，BUS\_[3..0]=0010，M=0：

，

与相加得到

串行与并行加法器的运算结果为7，结果没有溢出。

如果改为 BUS\_[3..0]=0011：

，

与相加得到

串行与并行加法器的运算结果为8，结果发生溢出。

● 令 BUS\_[7..4]=0101，BUS\_[3..0]=0011，M=1，写出该运算的真值表达式，记录并比较串行与并行加法器的运算结果，是否有溢出？如果运算器的输入改为 BUS\_[7..4] =0011，BUS\_[3..0]=0101，M=1 不变，结果如何呢？

解答：令 BUS\_[7..4]=0101，BUS\_[3..0]=0011，M=1：

，

对取补码得

与相加得到

串行与并行加法器的运算结果为2，结果没有溢出。

如果运算器的输入改为 BUS\_[7..4] =0011，BUS\_[3..0]=0101，M=1 不变：

，

对取补码得

与相加得到

串行与并行加法器的运算结果为E，结果没有溢出。

● 令 BUS\_[7..4]=1101，BUS\_[3..0]=0011，M=0，写出该运算的真值表达式，记录并比较串行与并行加法器的运算结果。是否有溢出？如果改为 M=1，输入不变，结果如何呢？

解答：令 BUS\_[7..4]=1101，BUS\_[3..0]=0011，M=0：

，

与相加得到

串行与并行加法器的运算结果为0，结果没有溢出。

如果改为 M=1，输入不变：

，

对取补码得

与相加得到

串行与并行加法器的运算结果为A，结果没有溢出。

**思考题**

1、请问本实验运算器电路是补码运算器还是原码运算器？假设需要把运算器电路改为四位无符号数的原码加法器，请问电路怎么修改？

解答：本实验运算器电路是补码运算器。

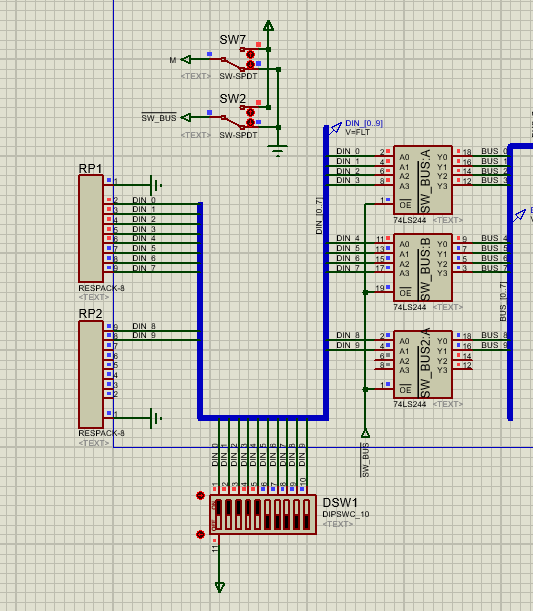
如果需要把运算器电路改为四位无符号数的原码加法器，只需将M端接入0即可。

2、本实验中，四位补码的表数范围是多少？若要修改为五位补码运算器（符号位一位，数值位四位），其表数范围又是多少？请问电路怎么修改？

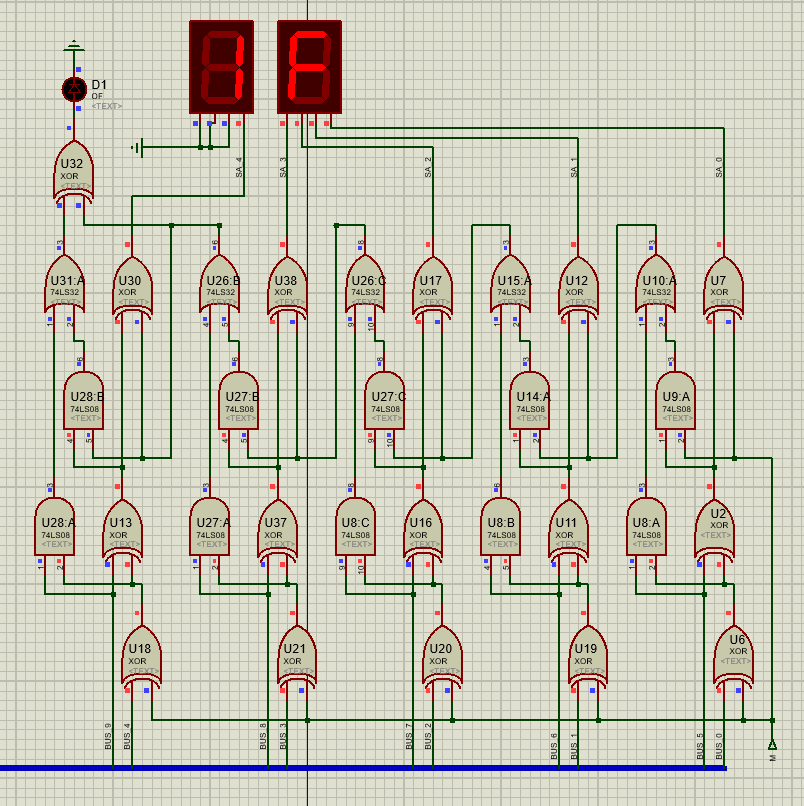
解答：本实验中，四位补码的表数范围是[-8,7]，若要修改为五位补码运算器（符号位一位，数值位四位），其表数范围是[-16,15]。

若要修改为五位补码运算器（符号位一位，数值位四位），以采用串行进位为例：

首先修改输入设备，使用两个RESPACK-8和一个DIPSWC\_10支持10位二进制输入。



然后修改补码运算器，使其支持五位二进制输入，如下图



**六、实验结果分析**

1、在进行进位加法器实验中，要注意数据是采用补码还是原码，是有符号数还是无符号数，同时还要注意数据的表数范围，例如4位补码运算器的表数范围就是[-7,8]。

2、补码运算器可以实现加法和减法运算，这取决于M端传入的是0还是1，传入0采用加法运算，1则采用减法运算。

3、该电路没有溢出检测，所以得到的结果不一定正确。

**七、本次实验的体会和建议**

1、通过本次实验，我了解了半加和全加运算器的电路结构，掌握串行进位加法器和并行进位加法器的原理及设计方法。

2、串行加法器有助于初步了解加法运算，但是它的效率远不如并行加法器，所以实现并行加法器还是很有必要的。

3、通过动手实现加法器，我对proteus软件有了更深的认识。