**东南大学电工电子实验中心**

**实 验 报 告**

**课程名称： 数字电路实验**

**第 5 次实验**

实验名称： FPGA时序逻辑设计

院 （系）：电气工程学院专 业：电气工程及其自动化

姓 名： 王皓冬 学 号： 16022627

实 验 室: 401 实验组别：

同组人员： 实验时间：

评定成绩： 审阅教师：

**一、实验目的**

1、 进一步学习时序逻辑电路的分析和设计方法，掌握状态机设计方法

2、 了解可编程数字系统设计的流程，掌握 Quartus II 软件的使用方法

3、 掌握原理图输入方式设计时序逻辑电路的方法和流程

**二、实验原理**

1. **输入信号与输出信号编码**

输入信号

用A、B作为输入的两个一位十进制数。其中，减法功能时，A是被减数，B是减数，计算式为A - B。十进制数输入由键盘输入。

输出信号

用Ones-dig-作为输出。其中，(Z1Z0)组成计算所得数值，Z1为十位，Z0为个位。Ad、Sb、Mtp，即Addition、Subtraction、multiplication，输出为“1”时分别代表当前功能为加法、减法、乘法。

其他标识说明

(A3 A2 A1 A0)2、(B3 B2 B1 B0)2分别代表X、Y对应的四位二进制数。Ci-1代表来自低位的进位或借位。ASi用以选择加减模式，AS=0时代表加法器，AS=1时代表减法器。

用S0-S4、Aso、Co作为输出。(S3 S2 S1 S0)2代表相加或相减得到的一位BCD数，S4在加法功能时作为BCD码数值的十位，减法功能时代表符号位。Co代表向高位的进位或借位。ASo代表用于串接的加减法功能选择的输出，直接接到输入ASi上，“1”代表减法功能，“0”代表加法功能。

M0 ~M7是串行乘法器得到的各位结果，CDEFG代表每次串行乘法相加得到的二进制数各位。其中，(C0D0E0F0G0)2、(C1D1E1F1G1)2、(C2D2E2F2G2)2、(C3D3E3F3G3)2分别是四次串行乘法得到的5位二进制数结果。

**实验内容：**

**1. 实验基础部分(70%) 设计一个简易计算器，它具有下列运算功能：**

(1) 一位十进制数的相加；

(2) 一位十进制数的相减；

(3) 数值和运算符用4×4键盘输入（实验室提供接口程序），其中A为“+”，B为“-”，C为“×”，E为“=”

(4) 数值用数码管以十进制形式显示，以加法为例，初始时显示全“0”，先输入被加数，再输入运算符，按下运算符键后，数码管显示全“0”，再输入加数，方法和前面一样，最后按下“=”，数码管显示运算结果

**2. 实验扩展部分(30%)**

(1) 一位十进制数的相乘，必须采用串行乘法实现；

(2) 其他自选功能

**3. 实验要求：**

(1) 根据设计要求划分设计层次、单元模块和接口信号，在预习报告上记录设计过程，绘制系统框图， 每个模块的状态转移图或 ASM 图，并设计验证方案。

(2) 用原理图输入法设计所有单元模块并编译，分析编译时产生的错误和警告信息

(3) 对所有的单元模块进行功能仿真，并记录和分析全部仿真结果

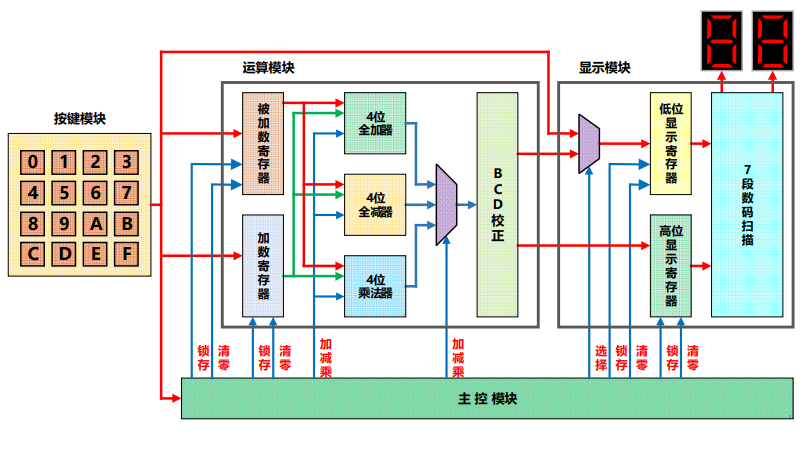
(4) 在顶层文件中连接全部单元模块并编译、综合、分配管脚和适配。

(5) 对整个系统进行时序仿真，并记录和分析仿真结果。

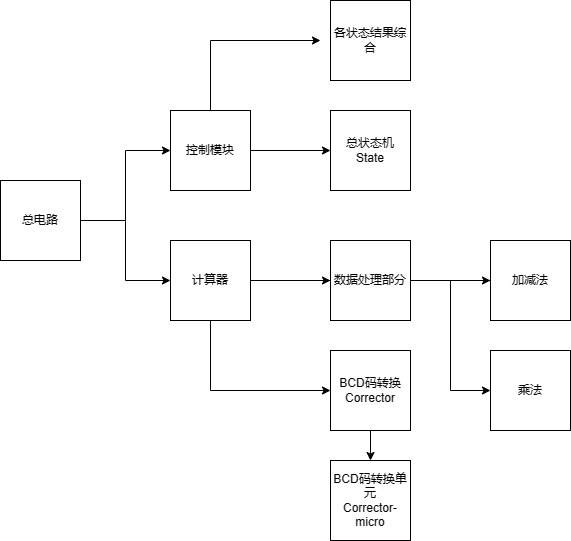
(6) 将仿真正确的设计下载到实验箱上，连接输入输出设备和示波器进行板级验证

**1. 逻辑模块**

该计算器的逻辑模块可按如下树状图表示：

****

粗略分为控制电路与数据处理电路两模块。控制电路包括了整个流程的状态机控制功能，数据处理模块包括了计算器的功能系统与校正系统。具体模块如下图。



**2. 模块设计与状态图**

**加减法功能：**

加减功能直接采用了实验二中设计的BCD加减器的设计，其真值表也引用了该实验实验报告的对应部分真值表。如下：

表 1 加法功能

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **输入** | | | | | | | | | | **输出** | | | | | |
| **A3** | **A2** | **A1** | **A0** | **B3** | **B2** | **B1** | **B0** | **CI** | **ASi** | **ASo** | **S4** | **S3** | **S2** | **S1** | **S0** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … |

**乘法功能（Mul4）：**

是通过串联乘法实现的。设计思路同教材。

在设计之初，采用的设计是4个7位二进制加法器串行实现，其中7位二进制加法器可以通过4位全加器得到。对比了该思路与教材思路，得到：

1. 该思路难以实现清零功能。虽然可以input一个clean信号，并接过一个反相器后与所有结果作与运算，使得clean输入为高电平时能将输出强行置零，但其清零时长难以控制。

2. 无寄存功能。这是最主要的一个区别，反思我自己的设计思路时，发现如果输入信号临时改变，会导致输出对应改变。

**主控电路（State）：**

设计时，对于“达到条件才进入下一个状态”有多种思路：

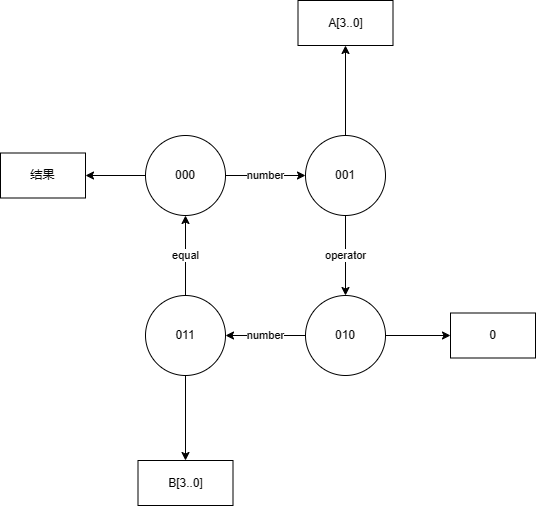
1. 可以采用74161，在使能端附加门电路组实现。由于过于复杂，未采用。由此引出一个话题：所有有转移条件的计数器，我们都可以采用在计数器使能端附加电路的方式实现功能。下文的思路也是基于此的。

2. 可以采用上一个实验中的One-Hot码或二进制码序列检测器的设计。同样，该设计较为复杂，故也未采用。

3. 可以利用数选器74151实现不同状态判断不同条件。记74151的各输入数据下标为i，则接入对应条件，如接入number。将计数器的二进制计数反接到74151的地址端，实现选择条件的功能；再将151的正输出Y接到161的使能端，控制是否计数到下一个状态（地址），便可实现151与161的互锁控制。

这种方法较为简单，实际设计的时候采用的是这种方法。

对应状态转换图如图所示。



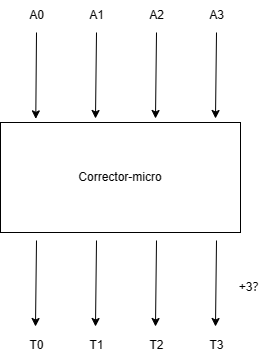
其中方框中内容为数码管应当显示的内容。

**8位二进制码转BCD功能（Corrector-micro，Corrector）：**

是利用转换单元（Corrector-micro）错位串联实现的，而转换单元采用的原理是“输入的4位二进制码大于等于5则加3”。这种方法的本质原理是，5（0101）左移一位，相当于乘2，得到的BCD码应是10（1000），但实际是1010（二进制码的10）。因此对0101加3得到1000，转换为BCD码。这其实是余三码的思路。

4个转换单元错位串联，即可实现移位校正。

逻辑图如下。



移位是在将Corrector-micro级联时完成的。该电路不需要状态转换机，可直接得出结果。这是因为乘法模块已有寄存功能，该模块只需要完成将乘法的二进制输出转换为BCD码输出的功能。

而判断“大于等于5”可以通过译码的思路实现。级联两个74138，得到4线-16线译码器。将输入的4位二进制数接入译码器端口，得到译码结果。将低位138的结果，以及高位138的结果作与非运算，即可判断“大于等于5”。这里没有考虑10~15的数据，因为属于无效输入，不会出现这种情况。

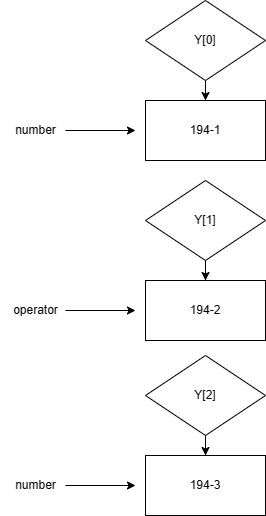
**数码管显示：**

由于计算器输出仅为计算结果，而数码管应实时显示输入数A、B和结果。因此，需要设计一个模块用于在不同State状态下正确输出对应的数值。

该模块的思路很简单，只需要将对应State状态的函数Y[i]与数值作与运算，再将所有状态的结果作或运算即可。

**寄存功能：**

利用74194寄存。采用3个74194，分别存放A、符号、B。其状态转换图如下。



其中，菱形框图代表判断。

**3. 逻辑函数式**

各功能部分逻辑函数式如下。

**2.1 加法：**

SD(二进制码)：

其中，代表（Q2Q1Q0）组成的最小项的现态。

SD(One-Hot码)：

**2.2 减法：**

用于产生待检测信号的16位ASPRG中，138组装4线-16线译码器，高位芯片与低位芯片的使能端分别如方程组（1）（2）所示：

(1)

(2)

**2.3 乘法**

乘法功能是利用封装好的adder4实现的。对于式 (B3 B2 B1 B0)2 Ai，记其来自低位的结果为(Ci-1D i-1E i-1F i-1)2，则应有：

(B3 B2 B1 B0)2 Ai + (Ci-1D i-1E i-1F i-1)2 = (Ci Di Ei Fi Gi)2

其中，(Ci Di Ei Fi)2作为下一个串行乘法的加数，Gi引出为Mi。最高位adder所得结果(C3D3E3F3G3)2直接引出为M7~M3。

8位寄存器的级联：将高位寄存器的QD接入低位寄存器的QA。

**4. 电路原理图**

**总电路状态机（State）：**

****

**串行乘法器（Mul4）：**

****

其中，由于CLK控制计数器间接控制74194使能端，与CLK直接接入74194之间存在延迟关系，因而实际计数变为1的瞬间74194仍然置数。所以计数器的结束位为5。

**校正器单元（Corrector-micro）：**



**8位BCD校正器（Corrector8）：**



该校正器是在Corrector-micro的基础上完成的，即校正器单元的封装芯片。

**BCD乘法器（BCDmul）：**



两模块简单顺联即可。

**BCD计算器（calculator）：**



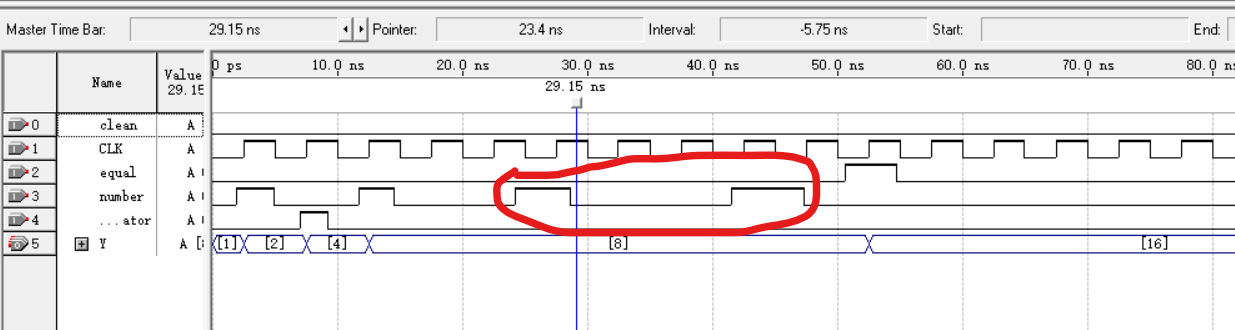
如上。其中，为了便于扩展功能，对8位输出的高四位与低四位及符号位分别输出。

**顶层（cal-UL）：**



**5. 仿真**

**总电路状态机(State)：**

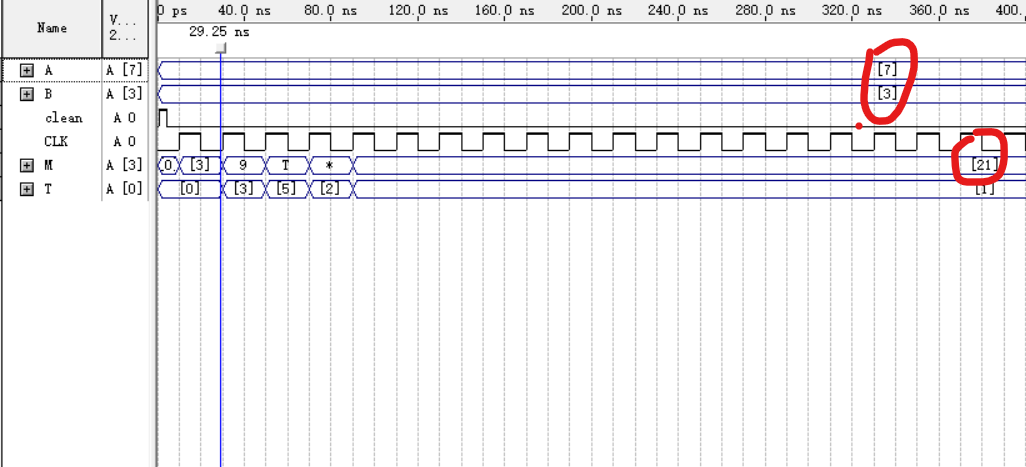
****

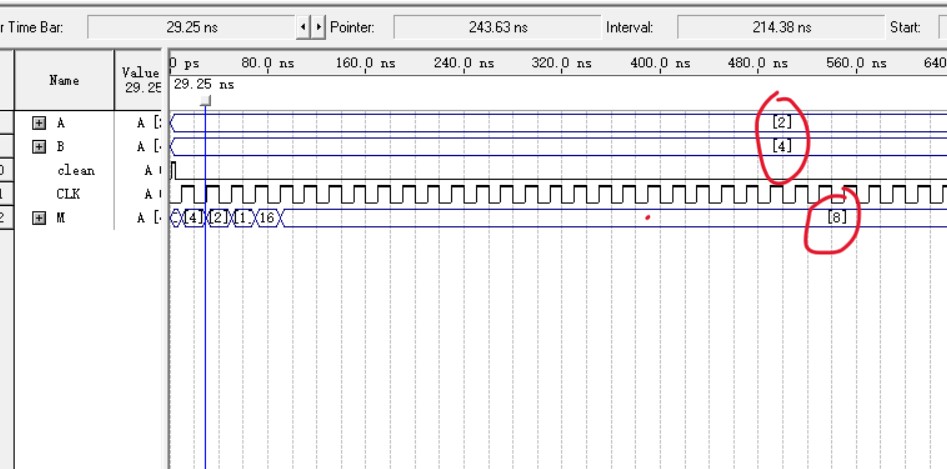
可以看到，状态机正确进行了状态转换。其中，Y初始状态为1的原因是Y是利用One-Hot码表征电路状态的，初始状态是Y[0]输出为1，Y[5..0]=00001，故初状态输出1。

红色部分为验证错误输入的部分。可以看到，应当输入等号（equal）时我们进行错误输入（这里为number），状态机不会进入下一个状态。

**串行乘法器(Mul4)：**

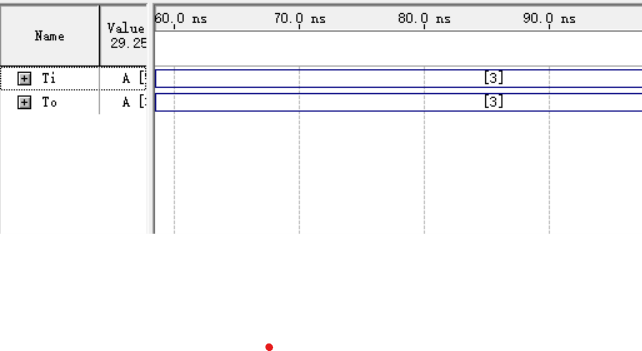
仿真所用计算式为7 x 3 = 21与2 x 4 = 8。M为输出，T为高位寄存器的输出端数值，是便于验证引入的。

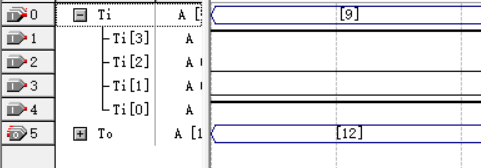


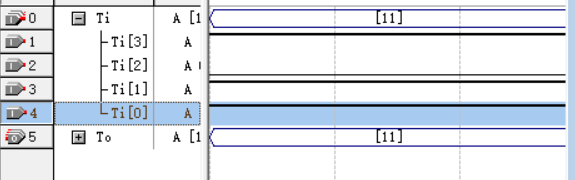


可以看到，在几次移位后，乘法器正确输出了结果的二进制表示，并能够保持。为方便验证功能，在t=0时为clean添加了一个高电平脉冲。

**校正器单元(Corrector-micro)：**



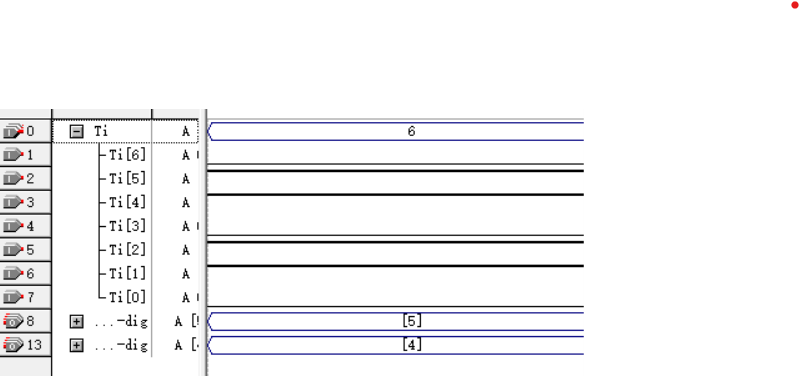




可以看到，数据为3时不校正；数据为9时校正为12；数据为11时超出校正范围，不校正。

**8位BCD校正器(Corrector8)：**

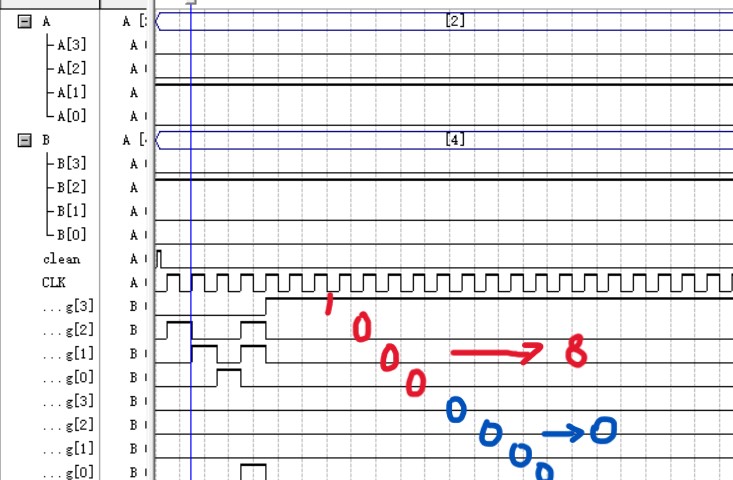
以54（0011 0110）为例。图中显示的Ti为“6”是错误的，因此截取了完整的波形表征是二进制数据54。



可以看到，十位和各位分别输出了5和4，结果正确。

**BCD乘法器(BCDmul)：**

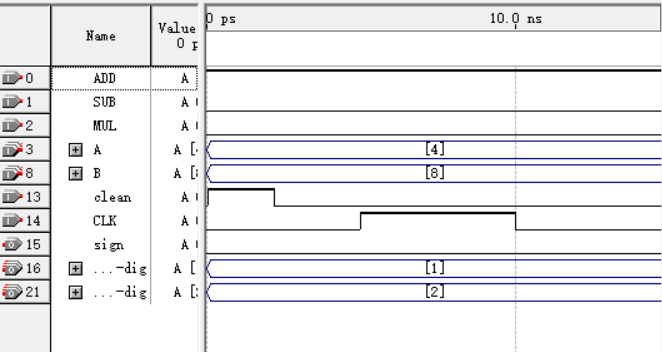
以2 x 4 = 8为例。

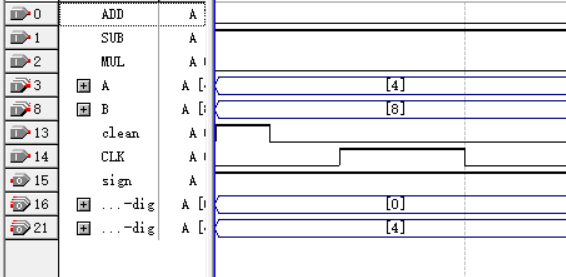


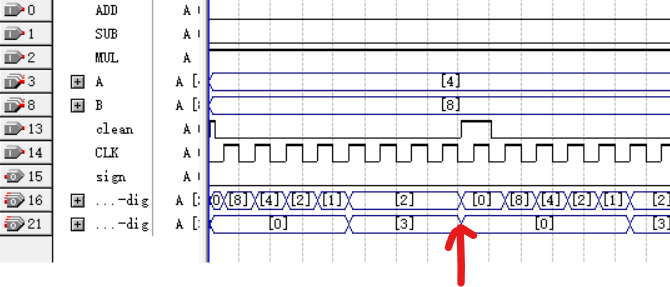
可以看到，**十位（蓝色）**与**个位（红色）**输出为0、8，正确输出了BCD码结果。

**BCD计算器(calculator)：**

以A=4、B=8为例。即计算式分别为4 + 8、4 – 8、4 x 8。从上至下分别为加法、减法、乘法。

（加法：4+8）

（减法：4-8）

（乘法：4x8）

可以看到，对于输入A=4、B=8，各功能均正确输出了BCD码表示的计算结果。其中，减法功能的sign（符号）也正确输出了“1”，即负号。

同时，对于乘法功能，若在时钟的低电平时期加入一个clean的高电平信号，观察到结果立刻被清零。这表明clean的异步清零功能有效。

**顶端（cal-UL）：**

**三、实验记录**（记录实验具体步骤、原始数据、实验过程、实验中遇到的故障现象、排除故障的过

程和方法等）

实验步骤：

遇到的问题及解决方法：

**四、实验仪器**

**五、实验小结**（总结实验完成情况，对设计方案和实验结果做必要的讨论，简述实验收获和体会）

**六、参考资料**