EDA 作业一　二进制运算器及其数码管扫描显示电路

实验报告

姓名：赵文亮

学号：2016011452

班级：自 64

日期：2017 年 11 月 30 日

目录

1 实验目的

1. 学习面向 FPGA 的简单数字系统的设计流程。
2. 掌握 EDA 软件 Quartus II 的原理图输入方式。
3. 熟悉试验装置——实验板，掌握板上外设的工作原理。

2 预习任务

2.1 所需电路模块及功能

通过分析实验要求，可知本实验中所需电路模块及功能如下：

1. 1 位二进制全加器。输入两个加数 A, B 和来自低位的进位 CI，输出求和 S 与进位 CO。
2. 4 位二进制全加器。输入两个四位二进制加数 $A_3A_2A_1A_0, B_3B_2B_1B_0$ 和来自低位的进位 CI，输出求和结果 $S_3S_2S_1S_0$ 和向高位的进位 CO。
3. 二进制运算器。实现运算 $S = M + N$。输入两个三位二进制数 $M_2M_1M_0, N_2N_1N_0$，二者均为原码表示，M_2, N_2 为符号位，其它位是有效数字。输出三位二进制 $S_2S_1S_0$（原码表示的求和结果）以及符号位 $Sign$。
4. BCD-七段显示译码器。输入四位二进制数，输出七个变量 a∼g 来控制数码管中线段的点亮状态。
5. 分频器。输入时钟信号（来自晶振），输出两位二进制数 A_1A_0 作为数码管的位选地址[1]。
6. 数据选择器。根据位选地址 A_1A_0（来自分频器）来选择应该输出的数据。输入为 M, N, S 的有效数字 $M_1M_0, N_1N_0, S_2S_1S_0$ 和位选地址 A_1A_0。输出选择结果 $D_3D_2D_1D_0$ 作为 BCD-七段显示译码器的输入。
7. 2 线-4 线译码器。根据位选地址 A_1A_0 译码得到位选控制信号，输出 Y_3, Y_2, Y_1, Y_0 作为数码管的位选控制信号。

2.2 实验板外设资源

实验板的主要外设资源如表 1 所示。本次实验中需要用到的有拨码开关、LED、4 位扫描显示数码管、晶振、下载转换开关、配置芯片。

2.2.1 拨码开关

其原理图如图 1 所示。SW0∼SW7 接 FPGA 引脚。拨码开关拨向下时，开关断开，对应引脚输入高电平，反之输入低电平。

[1] 与最终实现的分频器略有不同，详见第 4 页 3.3

表 1: 实验板主要外设资源

输入模块	拨码开关	输出模块	LED
	独立按键		4 位扫描显示数码管
	4×4 矩阵键盘		蜂鸣器
双向模块	RS232-USB 接口	其他	晶振
	红外收发器		配置芯片
	扩展 IO		下载转换开关

2.2.2 LED

其原理图如图 2 所示。每个 LED 对应引脚输出高电平时被点亮。

2.2.3 4 位扫描显示数码管

实验中用到的 4 位扫描数码管是共阴极数码管。由图 3 可知，其共有 12 根数据线，按功能划分，可将他们分为段选线和位选线。段选线有 8 根（SEG_BIT1∼SEG_BIT8），是将 4 个数码管的每个线段（a、b、c、d、e、f、g、dp）的阳极并联在一起，接高电平的时候相应的段被点亮。位选线 4 根（SEG_DIG0∼SEG_DIG3），是将单个数码管内部的八个发光二极管的阴极接在一起作为位选端。由图 3 知，因为位选端连接了驱动电路，所以位选信号转为高电平有效。

当给定段选信号时，数码管上的字符内容都相同。给定位选信号后，对应位置的数码管将显示出字符。如果分时轮流控制各个数码管的位选端，各个数码管将轮流显示出字符。轮流显示过程中，每位数码管的点亮时间通常为 1∼2 ms，由于人的视觉暂留现象及发光二极管的余辉效应，尽管各位数码管并非同时点亮，但只要扫描的速度足够快，人眼看到的就是一组显示稳定的数据。

2.2.4 晶振

FPGA 输入时钟使用 50 MHz 的晶振。本实验中需要利用对晶振产生的时钟信号进行分频来实现数码管的扫描显示。

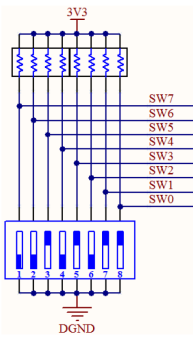

图 1: 拨码开关原理图

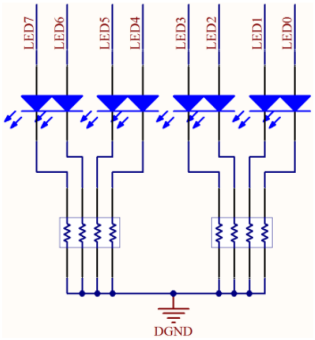

图 2: LED 原理图

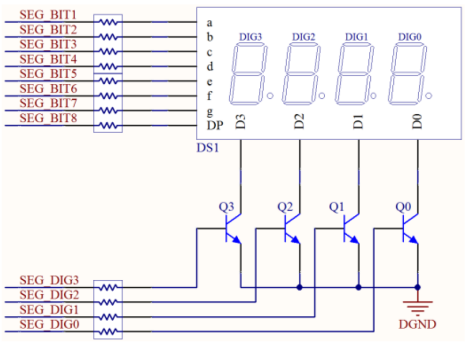

图 3: 数码管原理图

2.2.5 下载转换开关

有 JTAG 和 AS 两种模式。JTAG 模式下可以进行在线调试，但掉电之后数据消失；AS 模式下将程序写入 FPGA 配置芯片。

2.2.6 配置芯片

AS 模式下可将程序写入配置芯片中，每次上电时自动对 FPGA 进行配置，从而可以保证数据的持久。

3 设计思路

通过实验三（组合逻辑电路的设计），我对设计复杂组合电路的过程更为熟悉，也对电路模块的封装有了更深的理解。本次实验总的设计思路就是将整体功能划分为多个子功能，并确定每个功能模块所需要的输入输出接口，再予以实现。本次实验所需的电路模块见 2.1 。

3.1 二进制运算器

二进制运算器是本次实验的核心逻辑部分。其实现步骤如下：

1. 实现 1 位二进制全加器。可以通过分析真值表、化简逻辑式得到输出结果的逻辑式，进而画出电路图。我在本次实验采用了数电实验三的 1 位全加器电路。

2. 实现 4 位二进制全加器。将 4 个 1 位二进制全加器串接即可。

3. 二进制运算器。由于 M, N 是原码输入，S 是原码输出，而在运算过程中需要通过补码来计算，故在运算前后需要进行原码和补码之间的转换。这里利用了二者“取反加 1”的关系。当符号位是 0 时，该数非负，则无需转换；符号位是 1 时，需要对其余各位取反，并在末尾加 1。所以，取反操作可以通过有效数字位和符号位异或得到，而加 1 操作可以利用四位二进制全加器将该数与 0000 相加，并将低位进位端 CI 置为符号位得到。故总体实现过程为：

   （1）利用两片 4 位二进制全加器分别将 M, N 进行原码 $\Rightarrow$ 补码的转换，得到 $M_{\text{COMP}}, N_{\text{COMP}}$。

   （2）利用一片 4 位二进制全加器将 $M_{\text{COMP}}, N_{\text{COMP}}$ 相加得到补码运算结果 S_{COMP}。

   （3）利用一片 4 位二进制全加器将 S_{COMP} 转换位原码形式，并分别输出符号位 $Sign$ 和有效数字 $S_2S_1S_0$。

3.2 BCD-七段显示译码器

采用现成的中规模器件 7448 来实现。

3.3 分频器

为了将时钟频率降下来，我使用了三个模块进行分频。

1. 十分频模块。利用十进制计数器 74160 实现。

2. 五分频模块。利用置零法，将十六进制计数器 74161 改装实现。

3. 八分频模块。用于产生控制数码管的位选地址。直接利用 74161 的输出端实现。

利用 3 个十分频模块、2 个五分频模块、1 个八分频模块即可将时钟的 50 MHz 降到 250Hz。

动态扫描数码管显示的难点在于可能会有重影现象。实验中我也遇到了这个问题[2]。为了消除重影，我为分频器增加了灭灯输出端。在每次位选地址改变前后的一个区间内输出一个低电平作为灭灯信号，完美地解决了问题。

此外，由于实际运行过程中分频器输出信号 A_1A_0 是整个电路中频率最高的，为了避免于 A_1A_0 同时变化而带来的竞争——冒险现象，将 A_1A_0 设计成格雷码输出。

3.4 数据选择器

根据由分频器得到的 2 位地址来选择应该输出的数据。为了简化顶层电路，数据选择器输出结果设计成四位二进制数 $D_3D_2D_1D_0$，可以直接连接 BCD-七段显示译码器。内部通过两个双四选一数据选择器 74153 实现。

第三个数码管需要特别处理。当输入地址为 10（对应求和结果的符号）时，输入 74153 的数据全置为 1。这是因为 M, N 均为的十进制数值均不超过 3，则求和结果 S 的绝对值不超过 6，故输入地址为 00,01,11 时输出的高位 D_3 为 0。则 D_3 的值为 1 表示此次应该选通第三个数码管（求和结果符号），之后便可以利用 D_3 的值来控制第三个数码管的显示。

3.5 2 线-4 线译码器

根据位选地址 A_1A_0 译码得到位选控制信号。使用双 2 线-四线译码器 74139 来实现。

4 顶层电路及各模块电路功能

顶层电路如图 4 所示。顶层电路中的子功能模块主要有五个。左上角为封装的二进制运算器，可以直接得到运算结果；最左侧为分频电路，将晶振的时钟进行分频得到以 250Hz 变化的位选控制地址以及附加的灭灯输出控制端；中间为一个数据选择器，根据位选地址输出被选择的数据；最下侧为一个 74139 译码器，通过位选地址译码得到数码管的位选信号；最右侧是 BCD-七段显示译码器 7448，得到数码管的段选信号。中间附加了必要的门电路来实现逻辑功能。下面分模块进行详细分析。

4.1 二进制运算器

二进制运算器实现了将原码输入的 M, N 求和并输出原码结果 S。其底层模块主要有 1 位全加器和 4 位全加器。1 位全加器采用了实验三的电路，如图 7 。将四个 1 位全加器串接就构成了 4 位全加器，如图 6 。封装后的 4 位全加器功能很强大，利用它即可完成运算功能。图 5 是二进制运算器的电路。

二进制运算器实现过程最关键的步骤就是原码和补码之间的转换，其实现方法的原理见3.1。图中左上方和左下方的 4 位全加器分别将 M, N 转换为补码，中间的 4 位全加器实现了对 M, N 的补码的求和，得到 S 的补码，最右面的 4 位全加器将 S 的补码转换成了原码输出。

[2] 见第 11 页 7

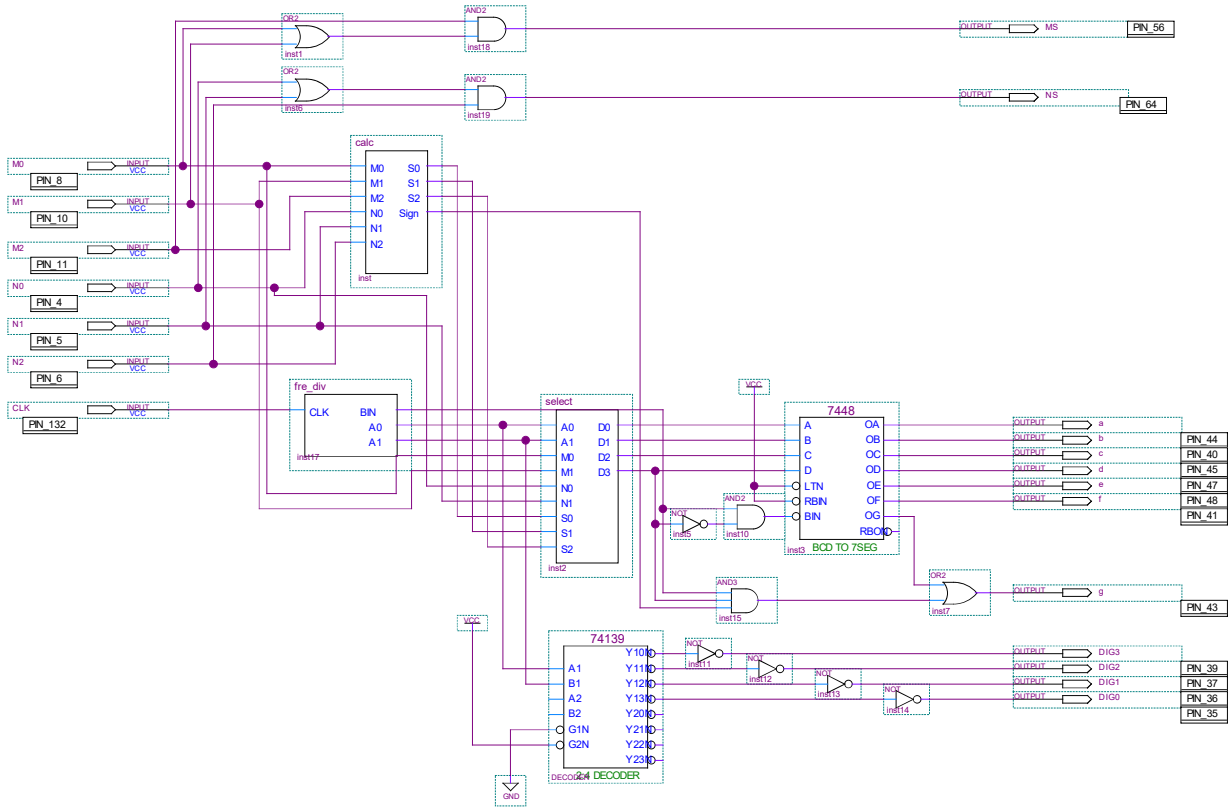

图 4: 顶层电路图

4.2 分频器

分频器利用晶振的时钟产生了数码管的位选地址信号 A_1A_0。这两个信号在一个周期之内需要遍历 00,01,10,11 四种取值，从而实现数码管的动态扫描显示。此外，为了从原理上有效消除数码管动态扫描过程可能出现的“重影”现象，我在分频器中设计了灭灯信号输出端。该信号在位选地址发生变化的前后，

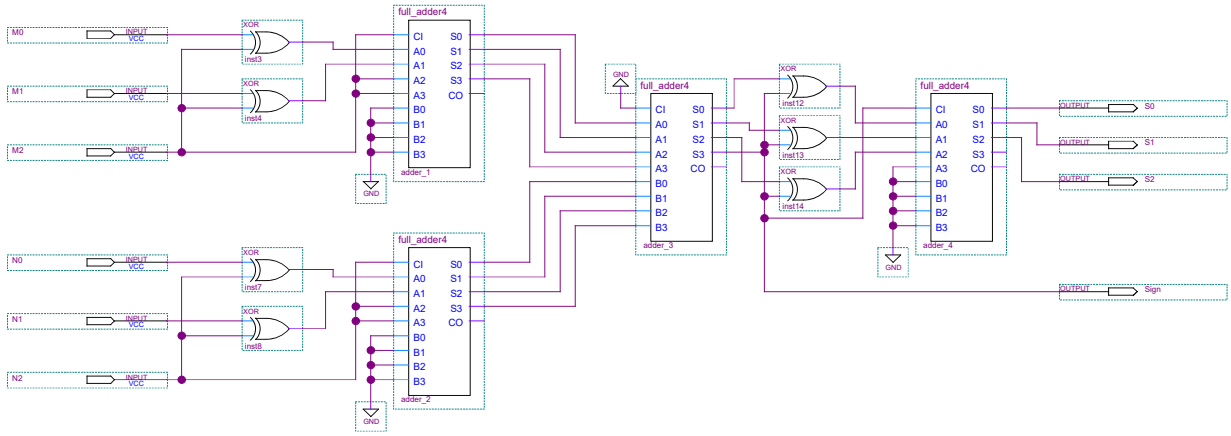

图 5: 二进制运算器

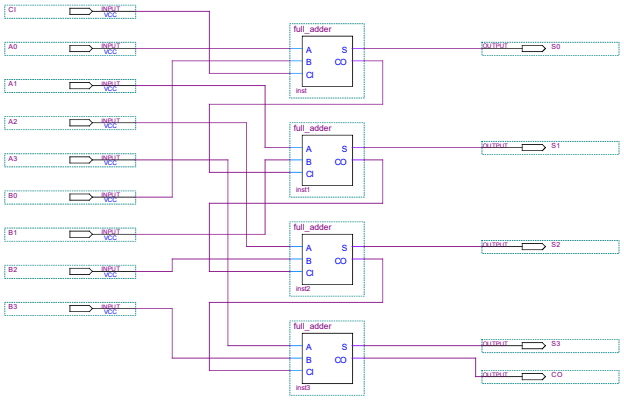

图 6: 4 位全加器

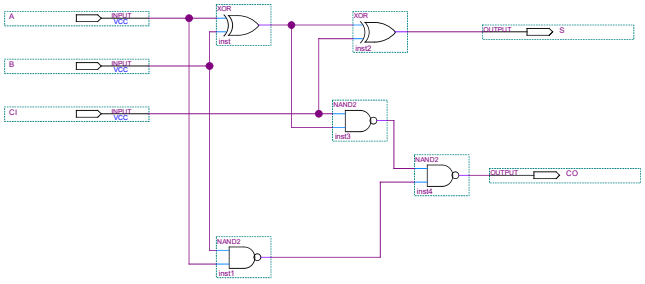

图 7: 1 位全加器

该输出端会给出低电平。将该输出端连接到 BCD-七段显示译码器的灭灯输入，即可实现数码管切换期间为灭灯状态，从而解决了重影的问题。

分频器输入的时钟信号 CLK 首先通过两级五分频模块（图 9 ）和三级十分频模块（图 10 ）。接着使用十六进制计数器 74161 的输出端得到输出信号。这里 A_1 输出信号频率为 74161 输入时钟的 1/16，将 OC 和 OD 输出结果异或作为 A_0 是为了使 A_1A_0 按照格雷码变化，从而避免后级电路由于 A_1, A_0 同时变化而产生的竞争——冒险现象。BIN 输出端由 OA 和 OB 异或而成，根据波形图可知，BIN 在 A_1A_0 变化时刻前后的一段时间会给出低电平。

十分频模块直接利用中规模器件 74160 实现。五分频模块利用 74161 通过置零法进行改装实现。信

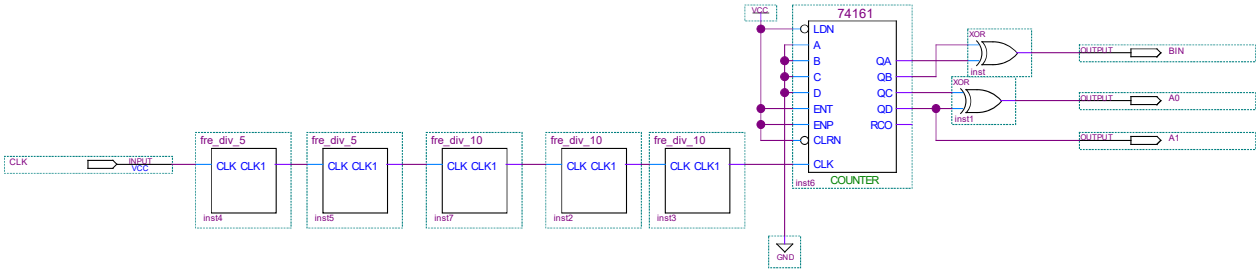

图 8: 分频器

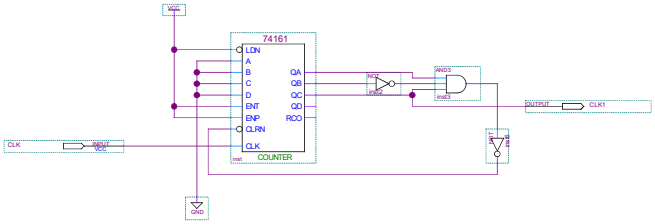

图 9: 五分频模块

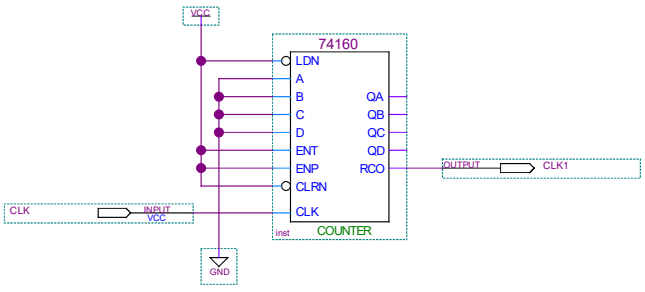

图 10: 十分频模块

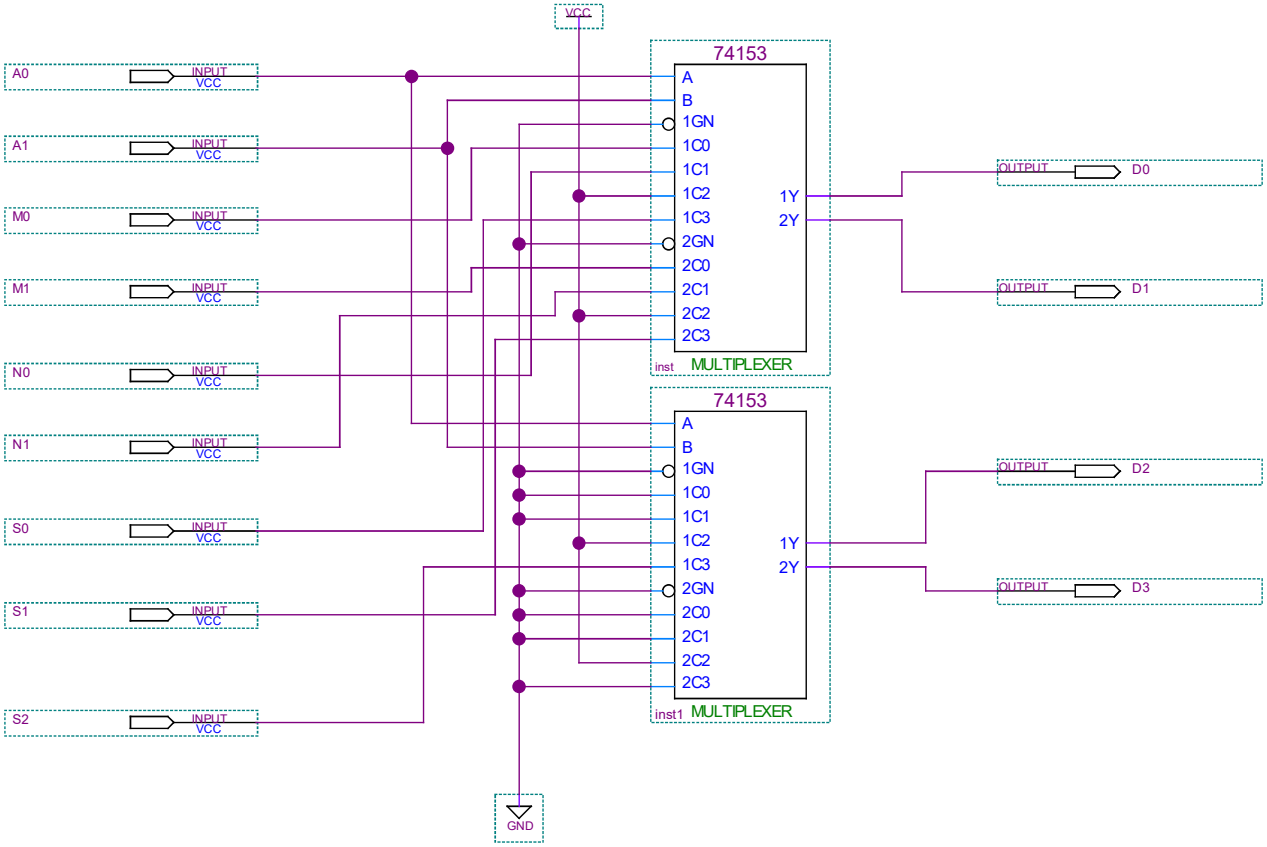

图 11: 数据选择器

号经过这两种模块后占空比会改变，但是由于最后一级的计数器 74161 各个输出端的信号的占空比均为 50%，分频器输出的位选地址的占空比也是 50%，满足后级电路的需求。

实验板上晶振的频率是 50 MHz，经过分频器之后频率降为

$$\frac{50 \times 10^6}{10^3 \times 5^2 \times 8}\text{Hz} = 250\text{Hz}$$

4.3 数据选择器

数据选择器由两片双四选一数据选择器 74153 构成，如图 11 所示。它将来自分频器的位选地址作为地址输入，实现了通过位选地址来选择需要输出的数据，输出的结果可以直接与 BCD-七段显示译码器相连。当位选地址为“00”时，输出“$00M_1M_0$”；为“01”时，输出“$00N_1N_0$”。

位选地址为“10”时比较特殊，此时应该选中第三个数码管，并在上面显示求和结果 S 的符号，故需要单独处理。注意到其余三种位选地址下最高位的输出均为“0”，故这里将最高位接入“1”便于区分。事实上其余三位可以接任意信号，但这里选择了全部接高电平。这是由于查阅了 7448 的真值表发现，当四位全是“1”时所有线段均不被点亮。这就保证了当位选信号为“10”时，7448 的输出端 OA~OG 均为低电平，从而方便了后续对符号的处理。

4.4 2 线-4 线译码器

使用现成的器件 74139，将位选地址译码，得到数码管的位选信号。

4.5 BCD-七段显示译码器

使用现成的器件 7448，从数据选择器获得输入，输出用来提供数码管的段选信号。然而，当位选地址为“10”时情况略有不同。这时第三个数码管被选中，由4.3可知，这种情况可以通过输入数据的最高位是“1”来判断。利用 7448 的灭灯输入端，可以由此实现灭灯；再利用运算器的输出符号位，通过简单的门电路，实现了负号的处理。

5 仿真波形及分析

5.1 全加器

5.1.1 1 位全加器

1 位全加器功能仿真波形如图 12 所示。实现了 $A+B+CI=\{CI,S\}$ 的计算[3]，图示位置有 $A=1,B=1,CI=0,CO=1,S=0$。即 1+1+0=10。

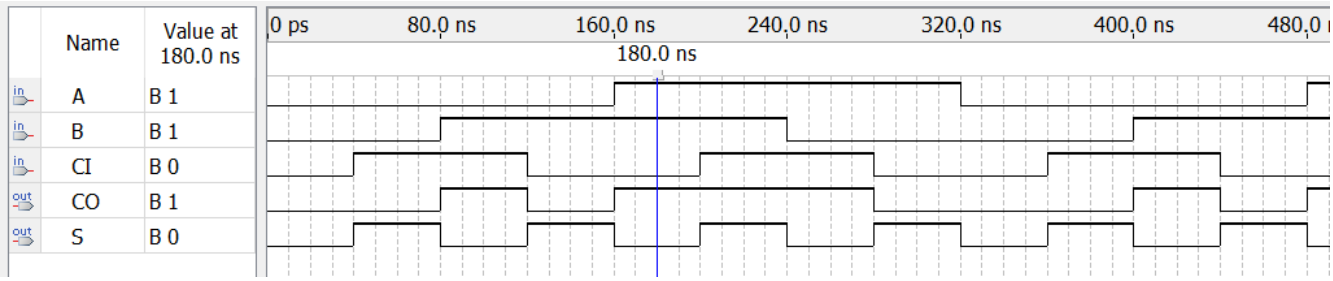

图 12: 1 位全加器功能仿真

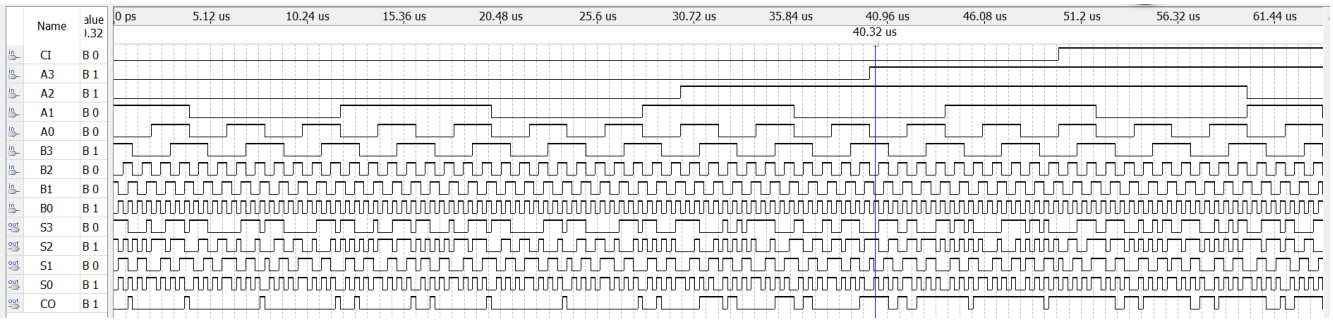

图 13: 4 位全加器功能仿真

5.1.2 4 位全加器

4 位全加器功能仿真波形如图 13 所示。实现了 $A_3A_2A_1A_0 + B_3B_2B_1B_0 + CI = \{CO, S_3S_2S_1S_0\}$。图示位置有 1100+1001+0=10101。

5.2 运算器

运算器的功能仿真波形如图 14 所示。

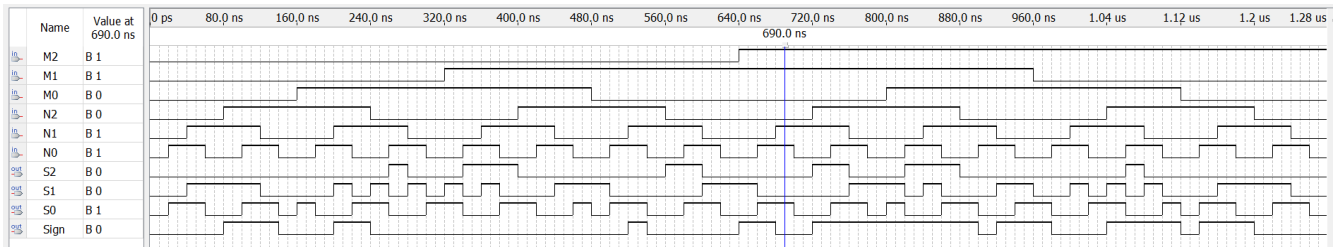

图 14: 运算器功能仿真

运算器实现了 $S = M + N$，且使用原码输入和原码输出。这样就可以方便地从波形图中读出结果。在图示位置有-2+3=1。

运算器的时序仿真波形如图 15 所示。

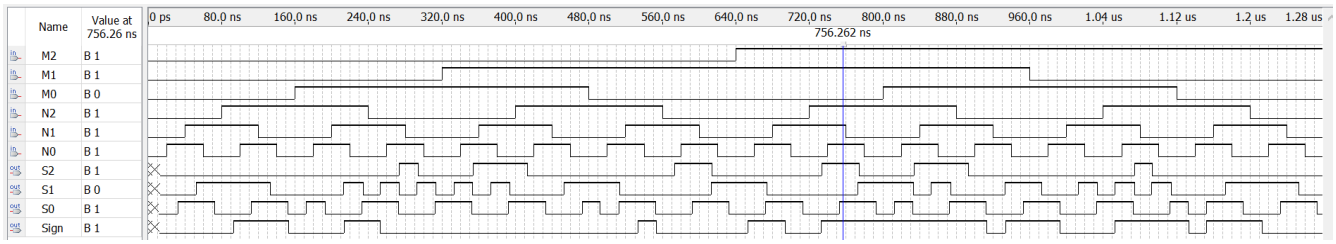

图 15: 运算器时序仿真

从波形图中可以看出传输延迟时间大约为十多 ns。图示位置有 (-2)+(-3)=-5。观察时序仿真波形时，要保证光标所处位置已经过了传输延迟时间。否则会发现输出结果不符合预期。

5.3 分频器

图 16 是分频器的功能仿真波形图。为了便于观察，我去掉了两个五分频模块进行仿真。

和预期结果一样，输出的位选地址 A_1A_0 在一个周期内按照格雷码变换，而灭灯信号输出端 BIN 保证每次位选地址变化的前后均为低电平。

其中五分频模块的功能仿真波形如图 17 ，十分频模块的功能仿真波形如图 18 。

可以看出，两个分频模块均能成功实现分频功能。最后一级的 7448 确保了输出信号占空比为 50 %。

[3]此处借用 Verilog 中的位拼接运算符 {} 的表示方法，下同

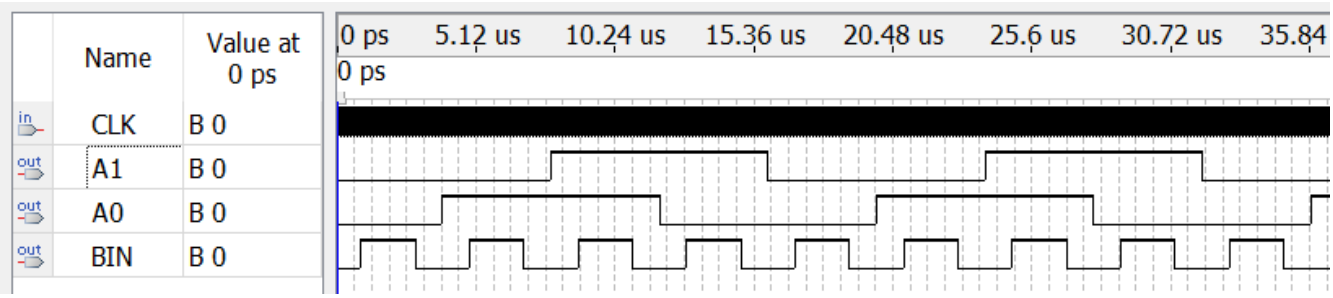

图 16: 分频器功能仿真

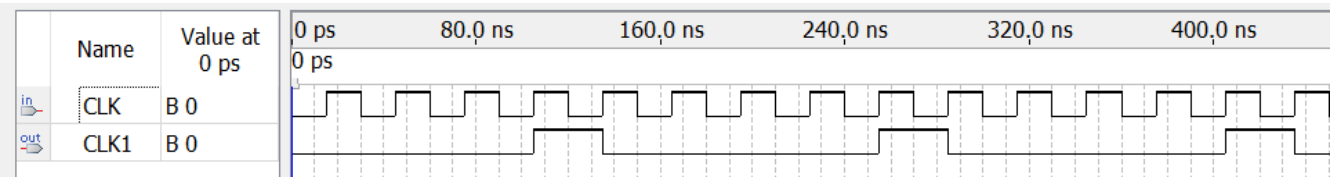

图 17: 五分频模块功能仿真

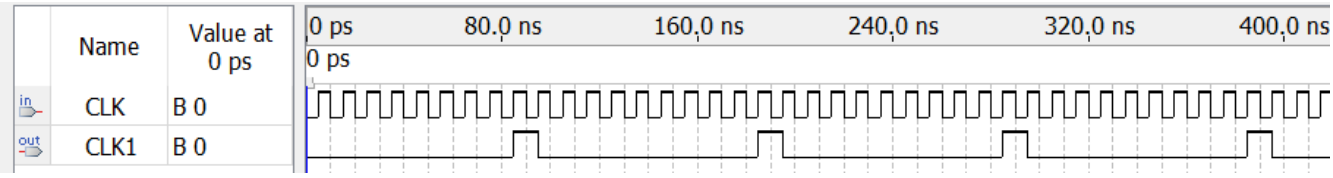

图 18: 十分频模块功能仿真

5.4 顶层电路

为了便于度数，这里将分频器中的五分频和十分频模块去掉，并取消了灭零输出端。得到的时序仿真如图 19 所示。从图中可以看到一些尖峰脉冲，例如图中红色圆圈中的部分。但放大之后发现与竞争——冒险现象无关，而是由位选地址的变化产生的。利用光标可以读取数值验证功能，图示位置 $M = 1, N = -2, \text{DIG3} = 1, b = c = 1,$。这说明此时正在显示最后一个数码管，且线段 b, c 被点亮。查阅数码管的原理图可知，此时正好对应显示 “1”。而 $S = M - N = -1$，其有效数字是 1，故结果正确。

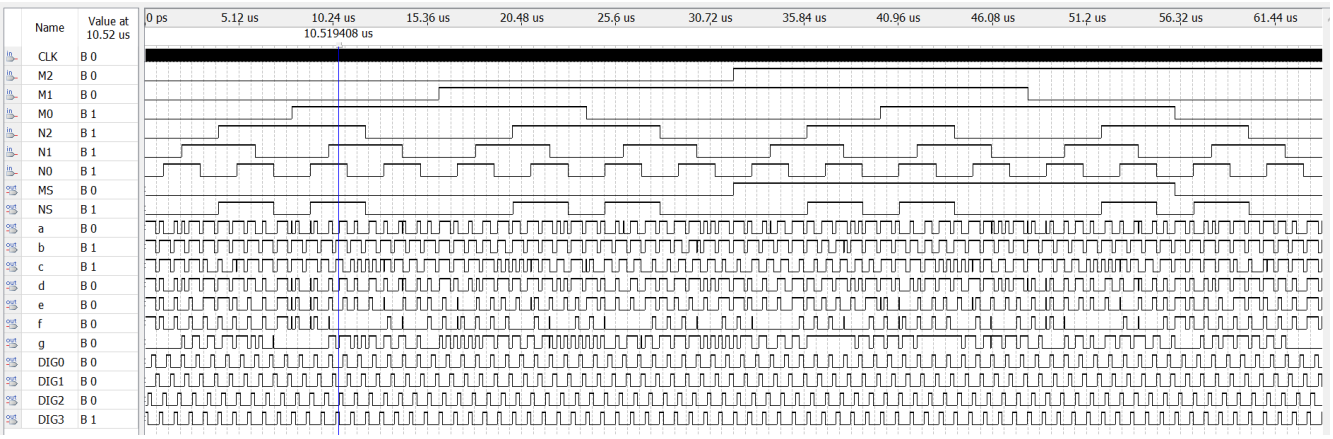

图 19: 顶层电路时序仿真

6 电路的下载

首先进行引脚分配。我使用了 6 个拨码开关 DIP2∼DIP7 来作为 M 和 N 的输入。四位数码管分别显示 M、N、S 的符号、S。使用两个发光二极管 D0 和 D7 来显示 M, N 的符号。最终效果如图 20 所示。此时 $M = -1, N = -3, S = -4$。从两个发光二极管可以清楚地读出 M, N 的符号，从数码管上可以看到运算数以及运算结果。

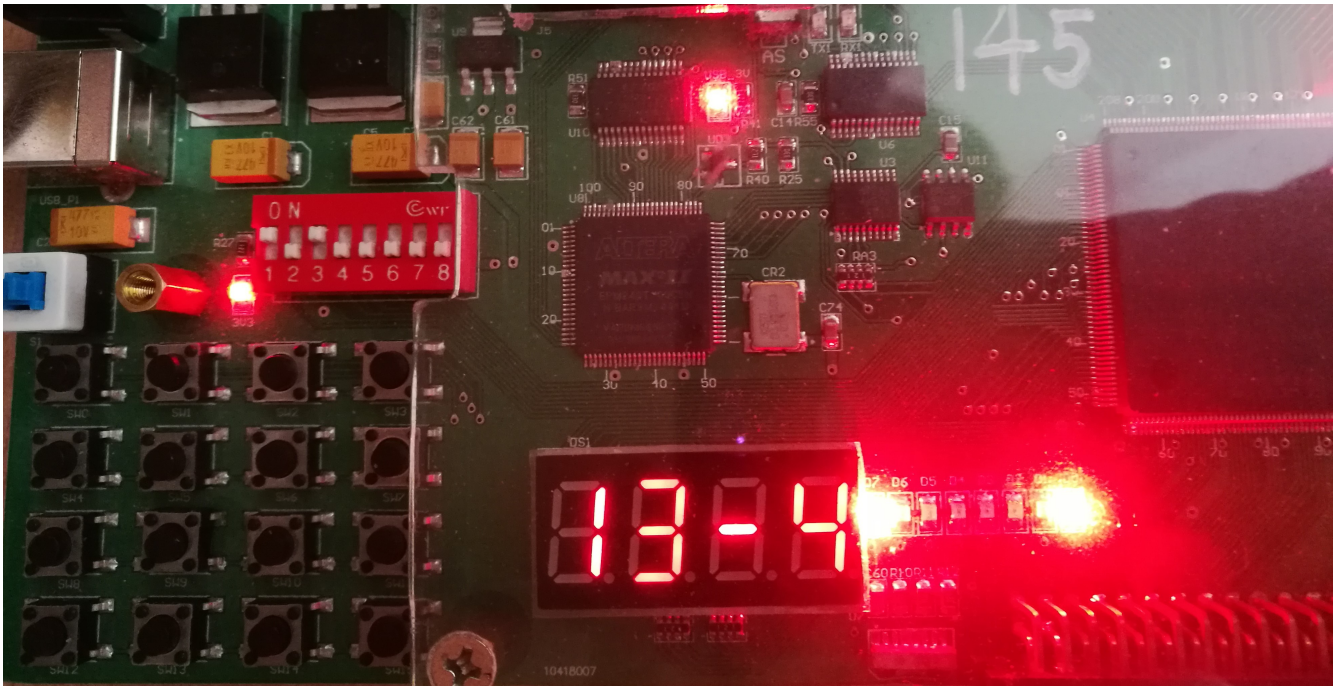

图 20: 最终效果图

由于我使用了 AS 模式进行下载，程序的持久性得到保证，掉电之后再次上电仍能运行而不用重新下载。

7 设计和调试中遇到的问题及解决方法

本次实验整体来讲较为顺利，但是过程还是出现了一些问题：

1. 数码管显示重影。这个问题困扰了我很久。我首先按照要求设计了分频器，并将频率降到了 250 Hz，但是仍旧有重影现象。一开始我把原因归结于扫描频率过高，事实上也确实有这个因素的影响。于是我试图使用增加分频器级数的办法来解决。由于我对实验结果要求比较高，我采用了非常苛刻的判别方法：在黑暗的环境中测试，并拨动拨码开关改变输入，观察四个数码管上是否存在重影现象，尤其是第三位。加了很多级的分频器实验效果仍不理想（其实在灯光下已经看不出来重影）。最后我终于消灭了重影，也增加了很多级分频器。

   但是当我准备拍照时，我发现用手机能明显看到数码管的闪烁，也就是说我增加的分频器级数过多，导致扫描频率跟不上。于是我又陷入了沉思。

我开始思考产生重影的本质原因。重影产生无非是位选信号和段选信号无法做到同时到达。例如，如果段选信号先改变，位选信号后改变，则会在原来的位产生一个重影，反之亦然。所以，解决问题的一个思路就是保证在位选信号改变的时刻所有的段选信号均为“0”，即灭灯。按照这个思路，我修改了分频器的设计，增加了附加的灭灯控制输出。并将频率调整回预期的 250 Hz，测试后发现结果非常完美。

2. 对 Quartus II 的时序仿真功能不熟练。一开始我试图使用 Quartus II 来对分频器进行时序仿真，但是我总也得不到预期的结果。后来我发现，这是由于输入信号的频率过高。于是我查看了全编译中传输延迟时间的结果，通过适当减少分频器中的分频模块来验证分频器的输出。

3. 拨码开关损坏。刚开始的时候通过一个简单的数据选择器来测试实验板，我发现改变第一个拨码开关的输入不能得到预期的结果。后来我使用一个简单的电路测试，将这个开关和一个 LED 相连，我发现这个开关只能输出高电平。于是我在后续实验中使用了其他的拨码开关。

4. 程序掉电即消失。一开始我以为这是实验板的问题，后来我知道，这是由于下载程序时使用了 JTAG 模式。于是我查阅了实验板使用手册，学会了使用 AS 模式下载程序，这样就实现了程序的持久化。