

**学生实验实习报告册**

|  |  |
| --- | --- |
| 学年学期： | 2017 -2018 学年🗹春🞎秋学期 |
| 课程名称： | 数字电路与逻辑设计实验A |
| 实验项目： | 基于FPGA的数字电子钟的设计与实现 |
| 姓 名： | 张永雄 |
| 学 号： | 2017210415 |
| 学院和专业： | 通信与信息工程学院通信信息类类 |
| 班 级： | 01141709 |
| 指导教师： | 应俊 |

**重庆邮电大学教务处制**

1. 系统顶层模块设计

该数字电子时钟设计具有如下功能：（1）秒钟、分钟以及小时计数的当前时间显示功能。设计思路：用两个模60计数器作为秒钟跟分钟的计数，用一个模24的计数器作为小时的计数，将实验平台上的50Mhz信号经由分频器fre\_div将1hz频率传给秒钟模60计数模块，将秒钟模60计数器进位信号传给分钟模60计数器作为秒计满60s的信号，将分钟模60计数器进位信号传给小时模24计数器作为分钟计满60min的信号，这就是秒/分/时计数原理；通过显示模块将秒/分/时的计数结果同步显示到数码管上，完成时间显示功能。（2）手动校时功能；设计思路：通过手动按键模拟上升沿（需按键消抖）作为进位信号给上级计数器，手动将上级计数器计数状态+1，通过一个按键控制是否校时，通过一个按钮选择分/时校时，通过一个按钮实现计数状态+1，完成手动校时功能（3）具有闹钟功能，闹钟具有开关、闹铃音乐、设置闹钟时间、暂停闹铃功能；设计思路：通过一个switch开关来控制闹钟模块的闹铃是否响铃；内置了一首音乐的闹铃音乐模块；通过一个按键来实现闹钟设置时间显示与当前时间显示的切换，类似于二选一显示；切换到闹钟显示之后即可通过前面设计的时钟校时模块的功能来设定你所需要的闹铃提醒时间，通过一个按键来确认你所设置的闹钟时间；当你的当时时间与你设置的闹铃提醒时间的分钟与时钟相同时，就会通过蜂鸣器发出内置音乐模块的声音；当你想提前让闹钟停止发出声音，你可以使用一个按键来停止播放闹铃，而且不需要你再次操作，下次同样能够在闹铃提醒时间准时响铃。（4）系统整体清零功能；设计思路:即利用计数模块中的74390的清零端来接自己的一个手动清零按键，将当前时间与闹铃提醒时间计数记录全部清零，闹钟功能也回到关闭状态，完成类似于恢复出厂设置功能。（5）整点响铃功能；实现整点时间响铃提醒。顶层设计图如下：

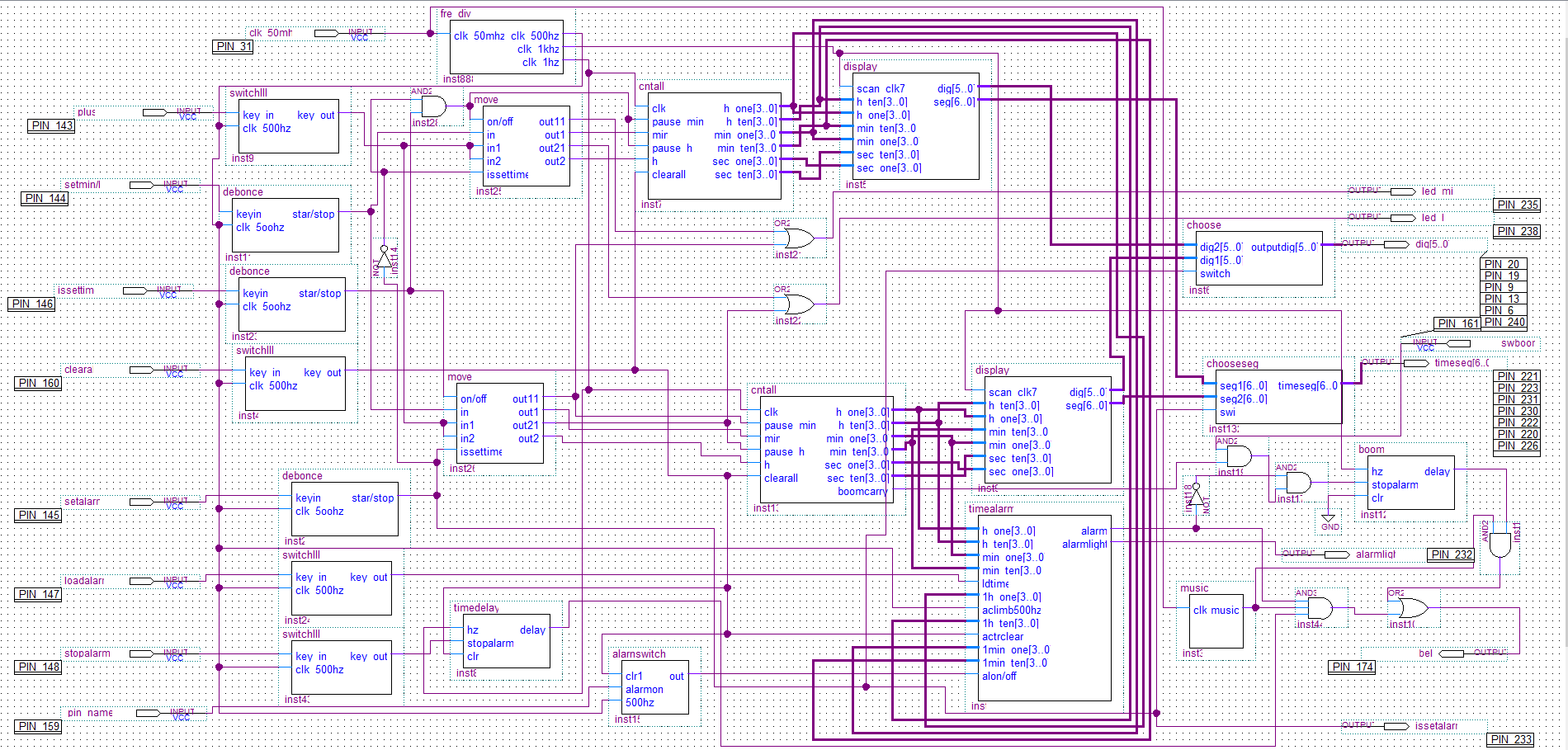


图1-1

1. 主要功能模块电路设计
   1. 分频模块

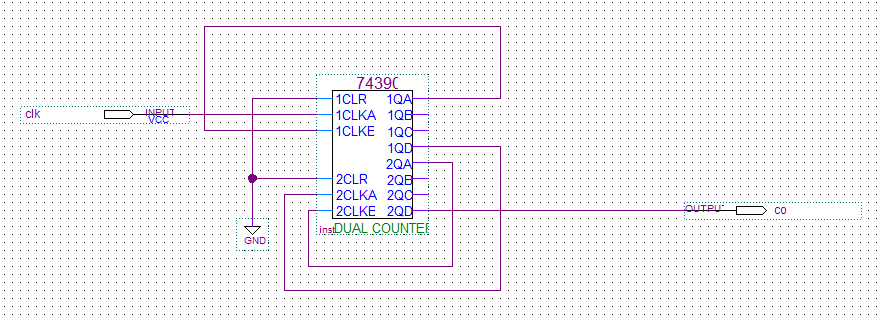
分频模块可以产生1Hz、500Hz以及1KHz的时钟脉冲、分别用做计时脉冲、按键消抖时钟脉冲以及动态显示模块动态驱动脉冲；由于平台提供50MHz的系统时钟，需要将分频产生上述几个时钟脉冲，通过简单的除法可知，通过50MHz除以50000000、100000以及50000即可获得相应频率，分频的基本原理就是计数，我们先使用74390制作模100的计数器方便使用，m100如下图：

图2-1-1

通过添加多个m100分频后再通过74390制作对应分频器所需模值，得到所需要的三个频率输出，fre\_div原理图以及封装后的模块如下：

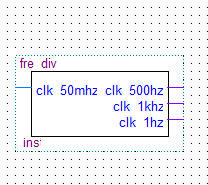
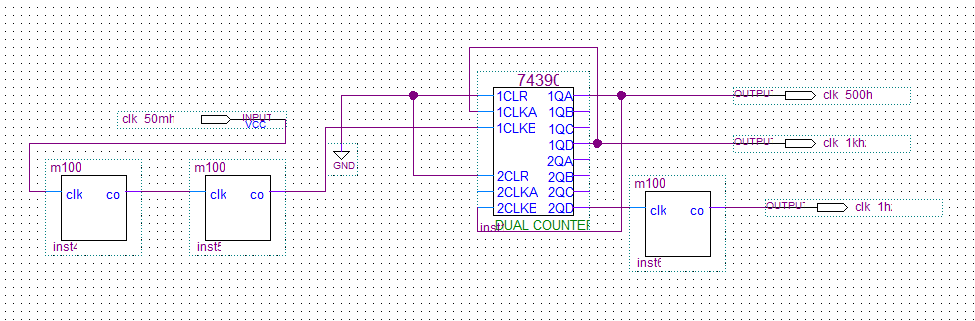


图2-1-2

* 1. 计时模块
     1. 分、秒计时模块（实现模60计数）

为实现秒钟跟分钟的计数，需要设计一个模60的计数器，我们可以利用74390来制作，采用10x10的方法，在计数达到60时实现异步清零，为了防止清零信号过短而造成清零不成功，利用D触发器来延长清零时间；将进位信号以及十位与个位的输出作为引脚，并且添加一个清零端口input作为手动清零端口，并设置一个input跟时钟输入端用与门相连，作为控制该模60计数模块是否工作的端口；具体设计图以及封装后模块如下：

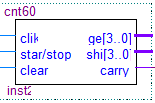
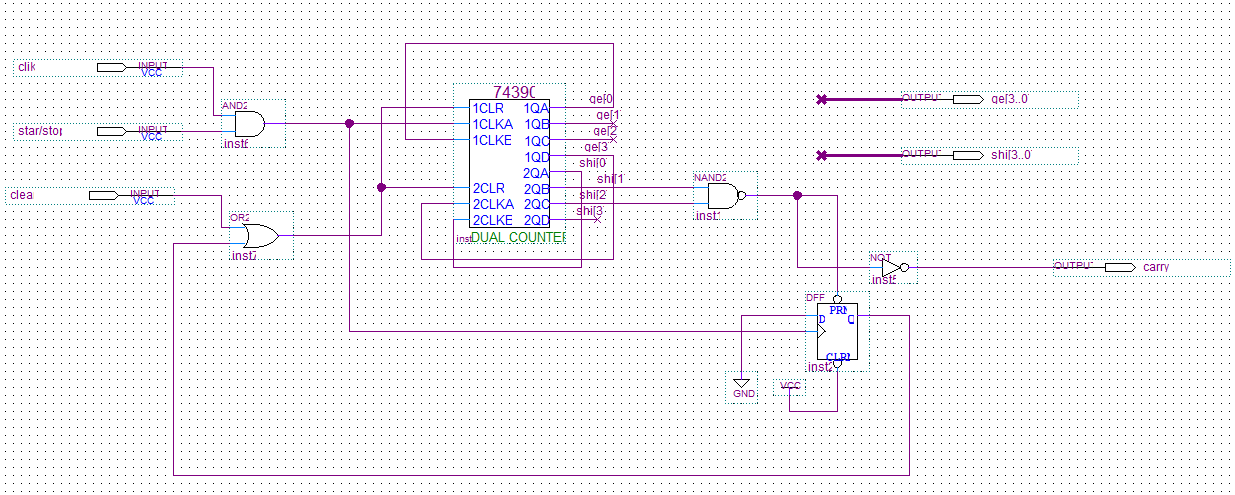


图2-2-1

* + 1. 小时计时模块（实现模24计数）

为实现小时的计数，需要设计一个模24的计数器，我们可以利用74390来制作，采用10x10的方法，在计数达到24时实现异步清零，为了防止清零信号过短而造成清零不成功，利用D触发器来延长清零时间；将进位信号以及十位与个位的输出作为引脚，并且添加一个清零端口input作为手动清零端口，并设置一个input跟时钟输入端用与门相连，作为控制该模24计数模块是否工作的端口；具体设计图如下：

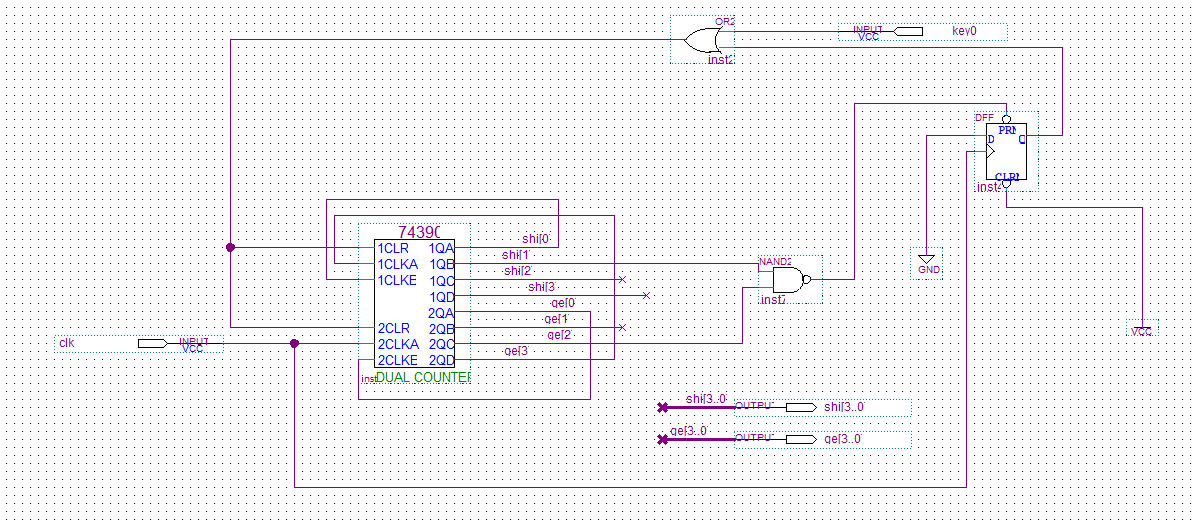


图2-2-2

* 1. 数码管动态显示模块
     1. 扫描模块couner6（实现6位数码管的扫描）

我们采用74390设计；模6计数器具体设计以及其封装元件还有波形图如下：

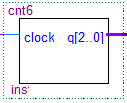
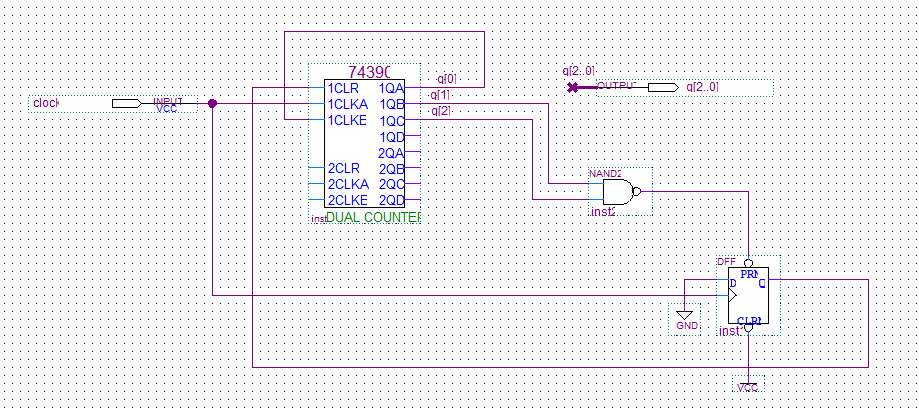


图2-3-1

* + 1. 位选模块dig\_select（实现6位数码管扫描时对应数码管的选通）

因为时钟显示需要6位数码管，我们的I/O端口有限，故需要实行动态扫描，我们通过一个模6的计数器来选择数码管的选通，也就是可以显示数据，当扫描速度足够快时，数码管与数码管显示间隔最大间隔极短再加上数码管的余辉效应以及人眼的视觉暂留，会让我们觉得6个数码管同时在亮，

我们首先设计位选模块，需要将模6计数器输出的3位二进制数转化为数码对应位数码管亮即该位为0的一段数（共阴数码管对应位为0时选通），因此我们选用74138译码器来做；具体设计电路、封装模块以及波形如下图：

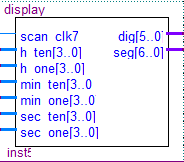
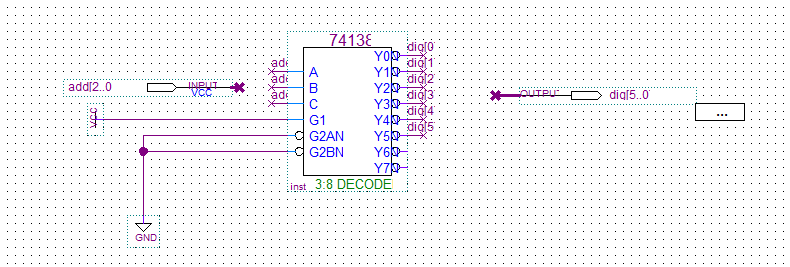


图2-3-2-1

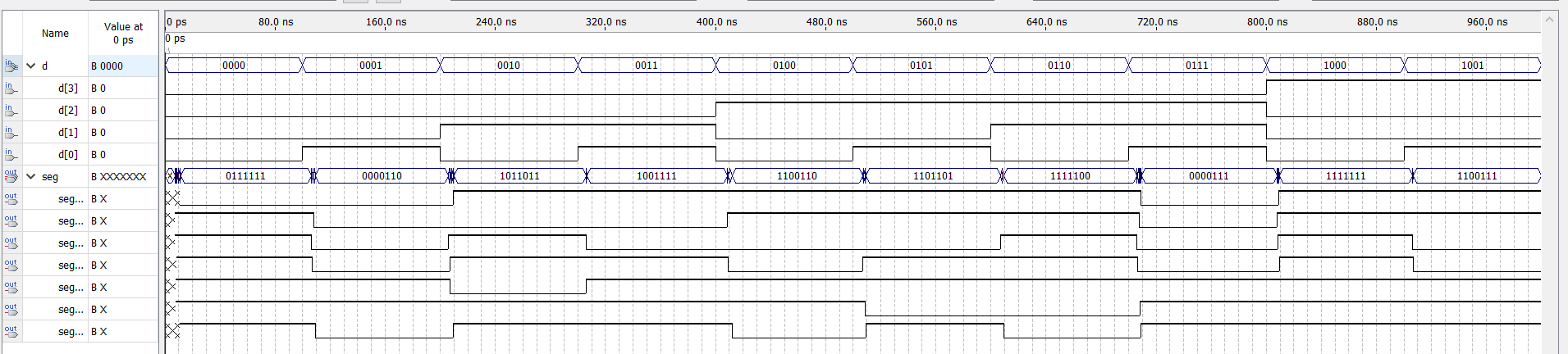


图2-3-2-2

* + 1. 段选模块seg\_select（实现6位数码管扫描时对应数码管数据的输入）

为了节省I/O口，我们采取动态扫描方案，当数码管某一位选通时，对应的那一位数据应该被传到译码器，然后传到I/O口，在该位上显示出对应的数字；因此我们需要一个数据选择模块，由于控制一位数字是由4位二进制编码控制，我们需要同时将他们传给下一级译码器，所以需要将每一位分别进行选择，用4个74151完成数选，然后封装，具体图示如下：

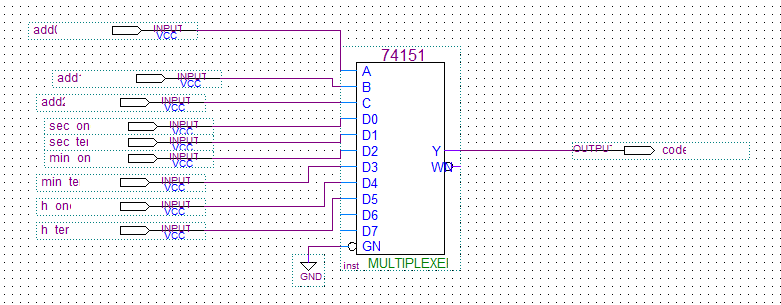


图2-3-3-1

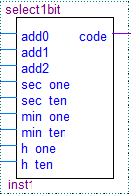
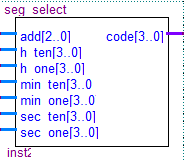
下图为seg\_select封装模块以及设计图：

图2-3-3-2 图2-3-3-3

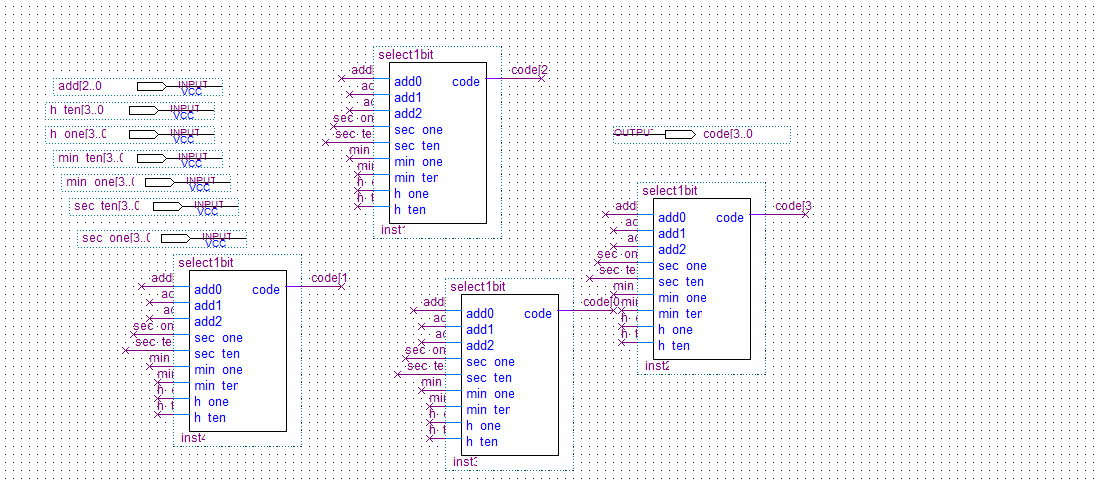


图2-3-3-4

* + 1. 译码模块decoder（实现对seg\_selet模块传来数据进行译码输出为数码管字形显示数据）

我们所知道的能够将四位二进制8421BCD码转换为数码管上显示对应数据应该显示的数字的模块有7448，下图为设计详图以及封装后的模块：

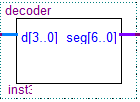
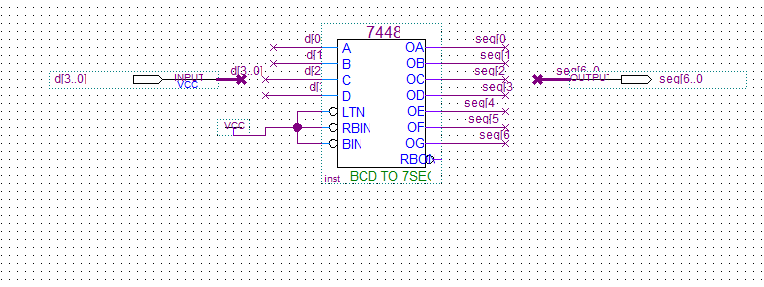


图2-3-4

* 1. 时钟分钟校时功能（实现手动调整时钟时间功能）

**2.4.1** 按键消抖模块（实现消除按键产生电平来回跳动的影响）

使用D触发器延时后多次检测的方法进行按键消抖，具体电路图见2-4-1-1。该电路是同步时序电路，一共使用了6个D触发器。由于时钟信号clk频率是500Hz，即信号经过1个D触发器延时为1个周期2ms，经过6个D触发器，延时为6×2mS=12ms，满足抖动时间最大10ms的要求。信号每经过1次延时进行1次采样，只有6次采样的结果均为高，输出才为高电平，说明12ms内信号均为高，判定为按键按下：若其中1次采样信号变为低电平，则说明信号为高的时间较短，输出为低，判定为抖动。

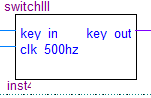
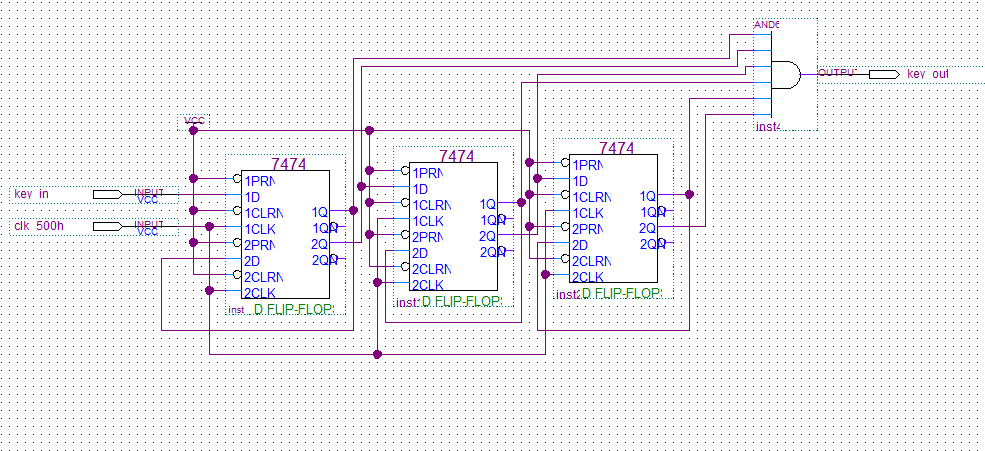


图2-4-1-1 图2-4-1-2

完成了按键消抖模块设计，我们需要设计一个按键来控制是否进行时间设置，当数码管显示为当前时间时，我们按下该键，可以开始校时，设置另一个按键，按一次可以再分钟与时钟调时间切换，然后再设置一个按键来手动加时间，具体设计电路图如下：

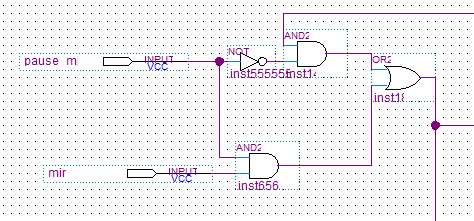


图2-4-1-3

当pause\_min为0时上半部分被选通，上级来的进位信号能够正常传给下一级，而手动加时间未被选通，因此无效，我们在此处还设立了一个LED灯输出来判断是否手动校时被选通；

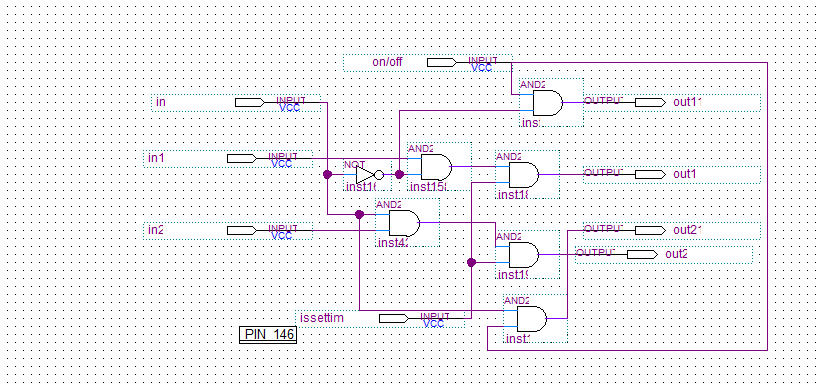


图2-4-1-4

上图为为按键移位电路图，on/off端口电平为1时，当按键未被按下也就in是电平是0时，上半部分被选通，下半部分未被选通，手动加时间只对上半部分的输入有效，反之亦然；

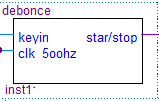
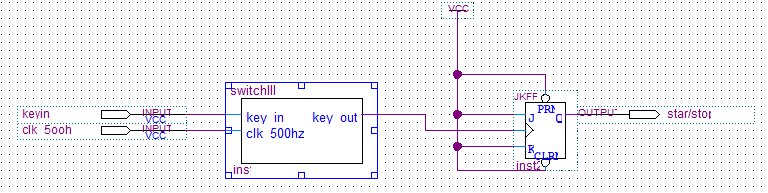
我们因此需要一个翻转模块，每得到一个按键的下降沿（消抖后），输出就会发生一次翻转，具体设计电路图如下：

图2-4-1-4 图2-4-1-5

* 1. 闹钟功能（实现设置闹铃时间，闹钟暂停响声，闹铃等功能）

2.5.1 闹钟与时钟显示间切换

闹钟模块显示跟时钟模块显示需要共用一套数码管，所以我们需要设计一个两组位选跟数选数据的选择的多数据二选一数据选择器；具体电路图以及封装模块如下（这组是位选二选一）：

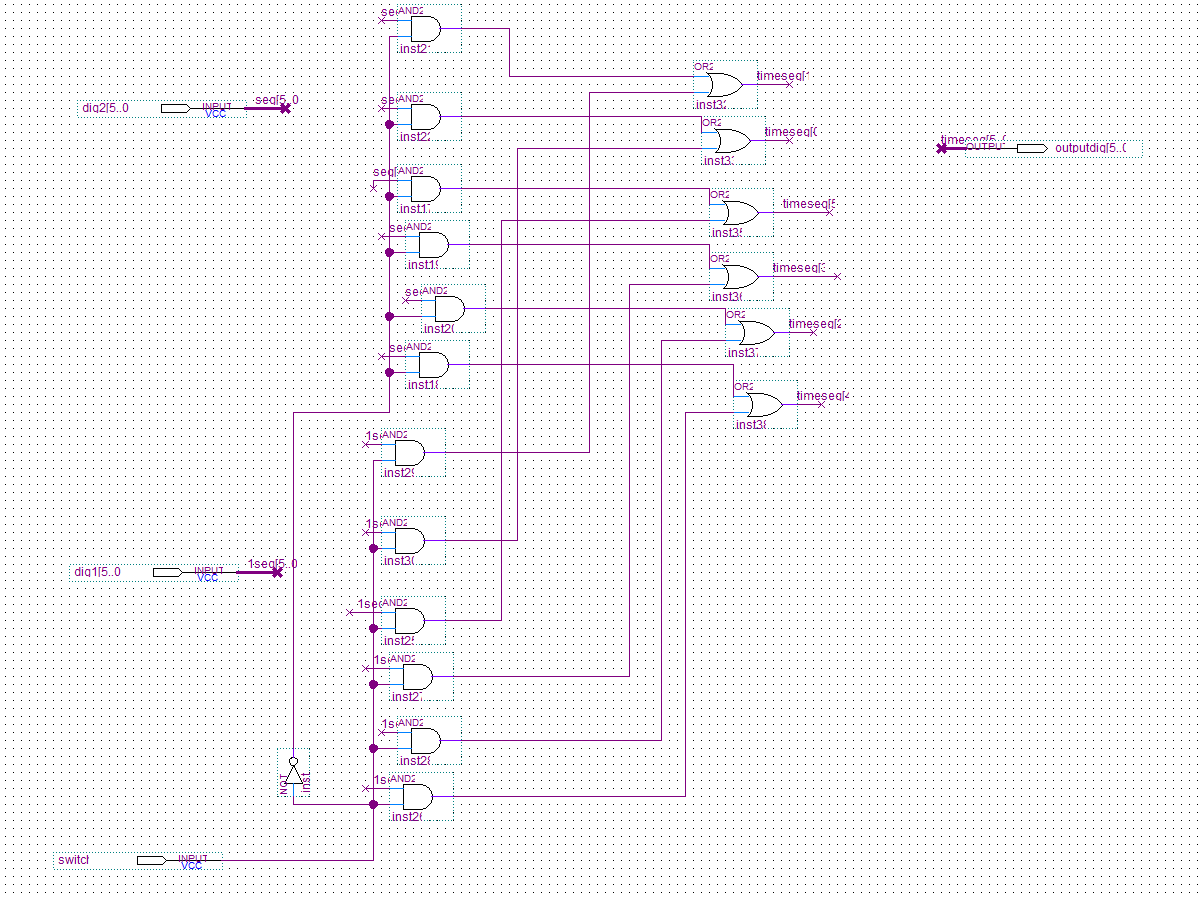


图2-5-1-1

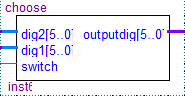
switch控制输出哪一路数据

图2-5-1-2

（这组是段选二选一）

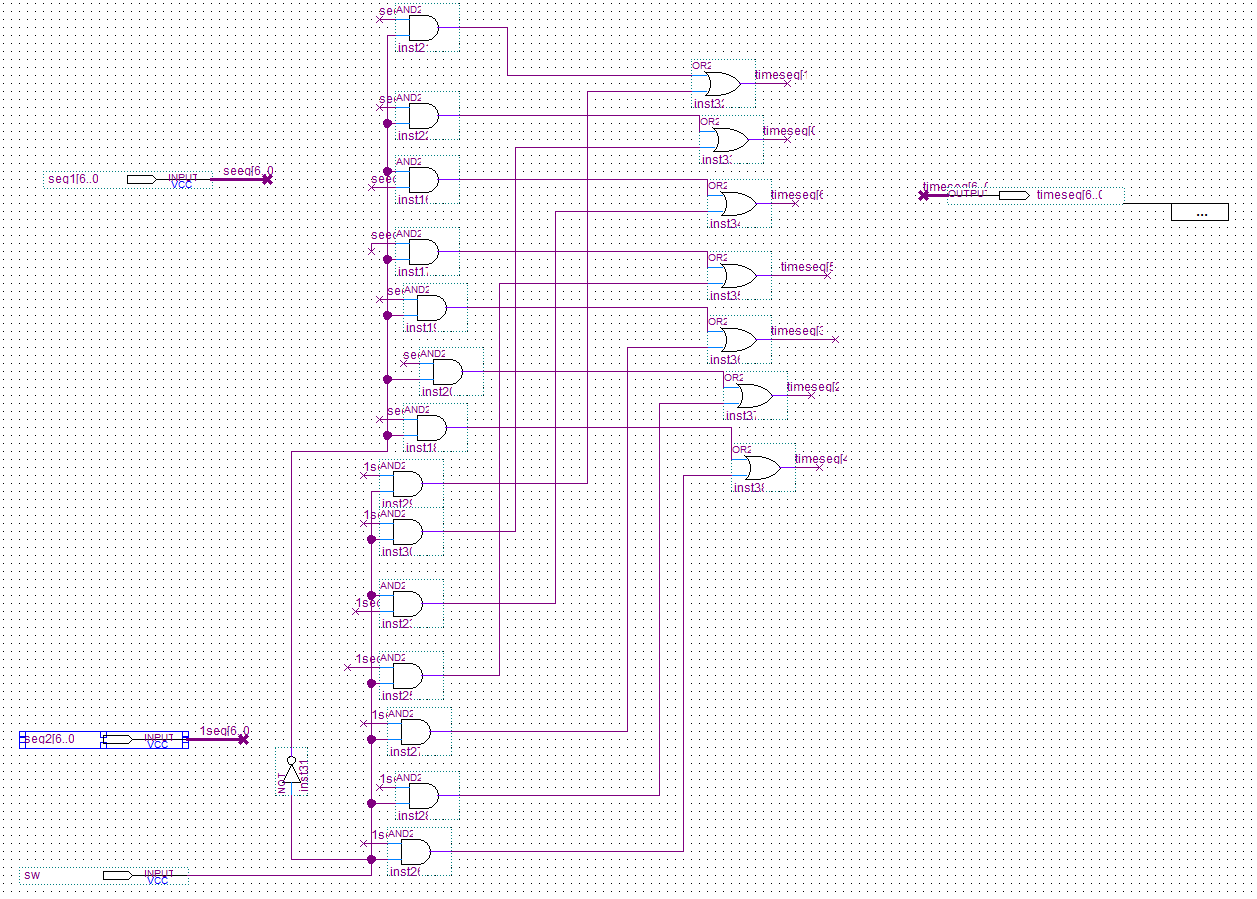


图2-5-1-3

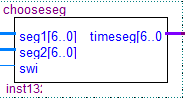
 swi控制该哪一路数据输出

图2-5-1-4

2.5.2 闹铃时间设置

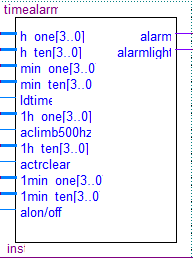
当我们上面所说的二选一切换到闹钟显示之后，我们就可以开始闹铃时间设置，类似于上面时间校时功能，也是手动开启时间设置，手动切换分/时，手动加指定位的计数值，也会有指定位的灯亮，但是我们会切断跟时钟计数模块的联系，以防我们设置时间时对该时钟模块造成影响。当我们的时间设置完了以后就可以将该时间设定为闹铃时间，我们利用数据寄存器，将设置的时间寄存在器件里面，然后用异或门或者同或门与时间作比较，当达到时间时会让音乐模块工作，通过蜂鸣器发声。具体设计电路图以及封装模块如下：alon/off为使能端，actrclear

图2-5-2-1

为清零端，ldtime为设置闹钟界面当前显示时间为闹铃时间端

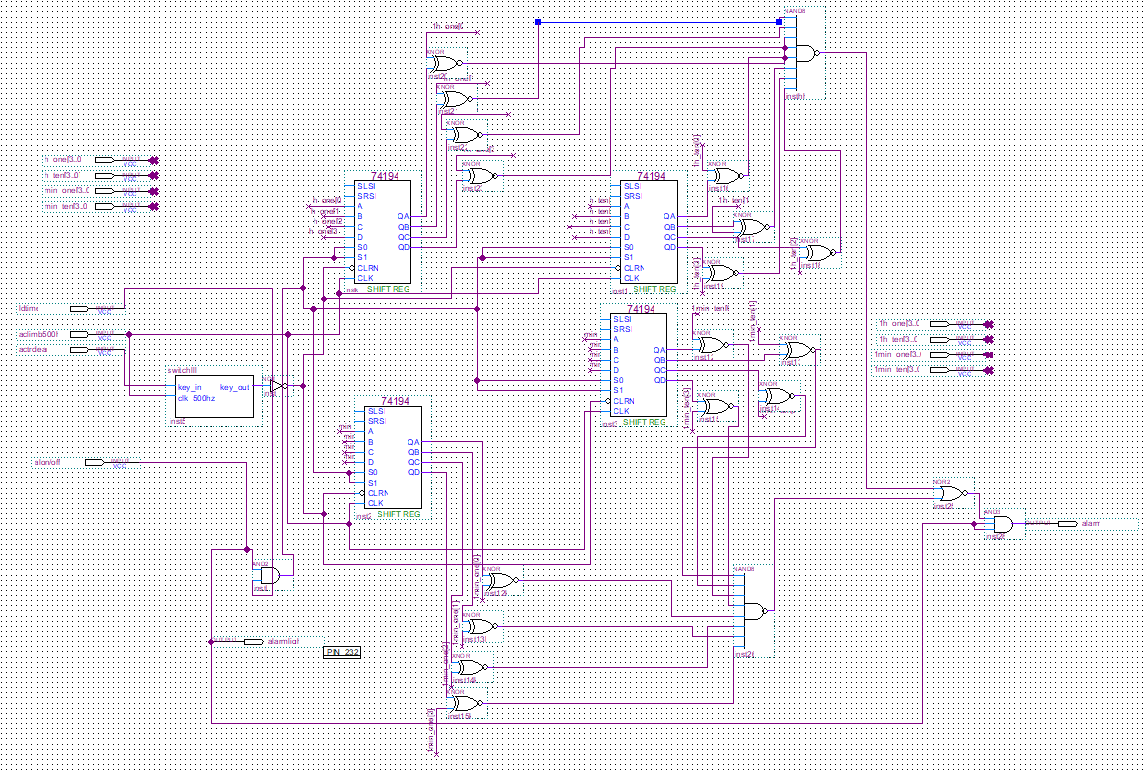


图2-5-2-2

2.5.3 闹铃暂停以及音乐模块

当闹铃响了之后，他会持续一分钟，此时我们想关掉它而不是关闭闹钟功能，我们就需要一个暂时关闭模块，这个模块利用了jk触发器的翻转，当我们不对他进行操作时，他稳定输出高电平，当我们给他一个下降沿时，jk触发器发生翻转，同时输入端输入失效，直到cnt60计数完成产生一次下降沿让他内部自己发生翻转回到原状态。具体设计电路及封装模块如下：

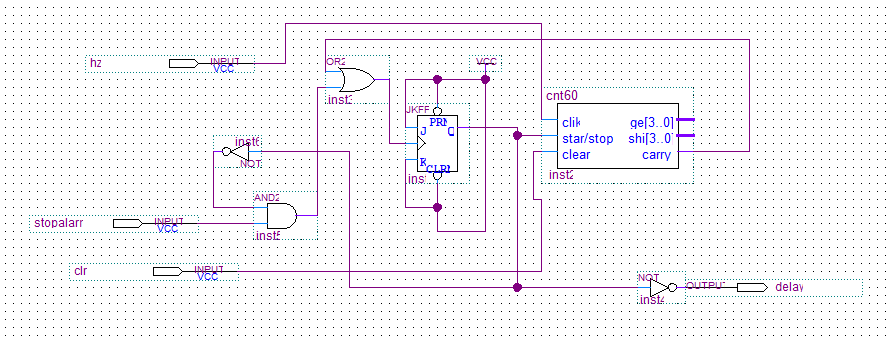


图2-5-3-1

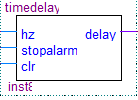
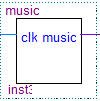
下图是一个内置有音乐的模块输入为50MHz；

图2-5-3-2 图2-5-3-3

* 1. 系统整体清零功能（实现将时钟/闹钟显示以及闹钟延时、闹铃提醒时间清零以及闹钟自动关闭功能）

各个计数器都留有清零端口，闹铃暂停模块也不例外，我们主要做出当清零按钮按下后要让闹钟自动关闭的功能，因为如果闹钟功能不能自动关闭将会导致一清零就是闹铃响的时刻00:00，所以我们需要设置一个手动开/关，而清零信号只可以将闹钟关闭而不可以再次开启的功能（以免短时间内多次清零），我们同样利用jk触发器的翻转功能，将输出的反与清零信号接与门然后再跟手动开启信号（需按键消抖）接或门，再接到jk触发器的时钟脉冲输入端口（翻转模块上面已经介绍）。具体设计电路图以及封装模块如下图所示：

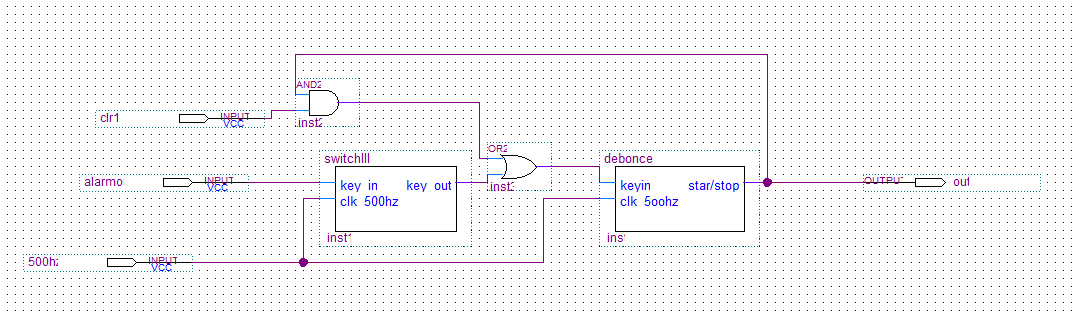


图2-6-1

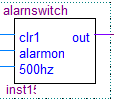
其中clr1为手动清零信号，alarmon为手动闹钟启动开关，接500Hz频率工作。

图2-6-2

2.7 整点响铃（实现整点提醒功能）

该功能类似于闹钟功能，但是想对于闹钟来说又有他自己的好处，能够整点响铃满足了许多不怎么计较分钟，但是在乎时钟的人，能够提醒人们过去了一小时，该模块利用了时钟计时模块中分钟的进位信号，设置了一个整点报时模块的开关，当开关关闭时，整点报时功能正常，当分钟模块向小时计数模块进位时，会有一个电平的变化，我们利用这个电平变化，激活整点响铃模块，由于设计中模块功能被激活好设置的持续时间时两秒（提醒即可），后恢复正常；同时，我们的整点报时模块为了不影响闹钟的响铃，所以设置的的优先级低于闹钟模块，具体设计电路图以及封装模块如下：

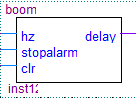
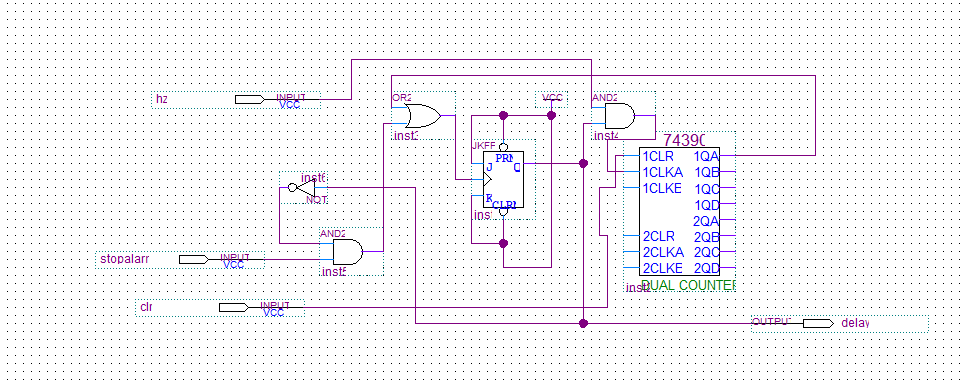


图2-7-1 图2-7-2

1. 各模块的测试方案及测试结果
   1. 计时模块
      1. 分、秒计时模块（实现模60计数）

测试方案：用quartus II软件仿真模60计数结果。

测试结果：如图3-1-1为模60仿真结果。

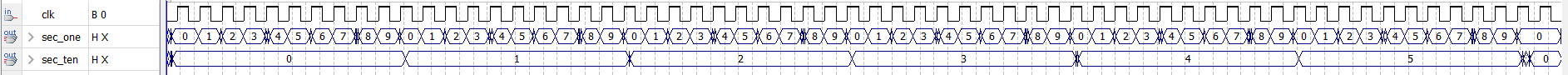


图3-1-1

计数值个位在时钟脉冲下由0增加至9后清零，十位接收到个位进位信号后从0增加至5，后清零重新计数，电路仿真后功能显示正常。

* + 1. 小时计时模块（实现模24计数）

测试方案：用quartus II软件仿真模24计数结果。

测试结果：如图3-1-2为模24仿真结果。

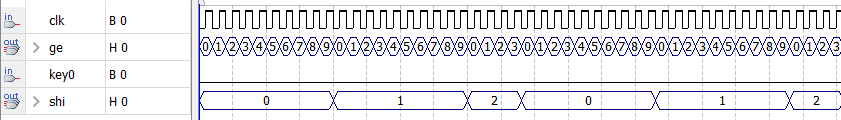


图3-1-2

观察波形可知，该计数器在时钟脉冲下个位由0增加至9后清零，十位接收到个位进位信号后从0增加至1，在短暂延时清零信号2后清零重新计数，完成模24计数功能。

* 1. 数码管动态显示模块
     1. 扫描模块counter6

测试方案：用quartus II软件仿真模6计数结果。

测试结果：如图3-2-1为模6仿真结果。

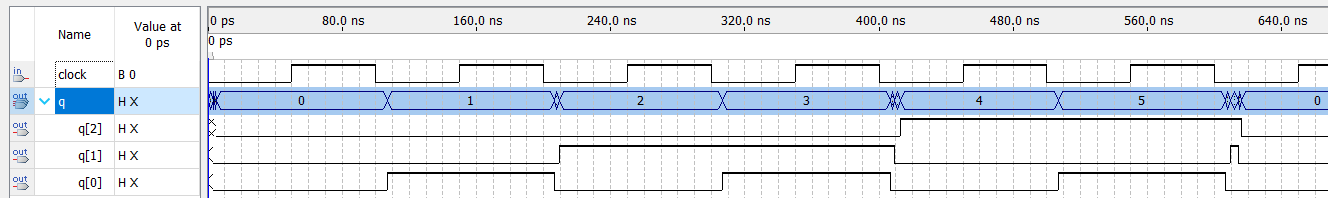


图3-2-1

计数值在时钟脉冲下由0增加到5，后自动清零，完成6模计数。

* + 1. 位选模块dig\_select（实现6位数码管扫描时对应数码管的选通）

测试方案：用quartus II软件仿真dig\_select输出结果。

测试结果：如图3-2-2为dig\_select仿真结果。

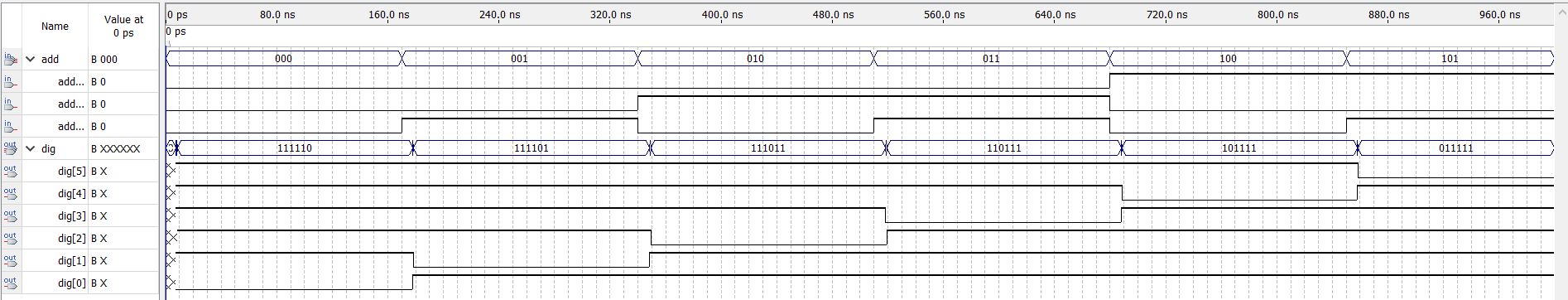


图3-2-2

观察分析该图可知，数据输入从000增加到101，成功译码并输出位选地址111110到011111。

* 1. 扩展功能模块

3.3.1 闹钟模块

3.3.1.1 按键消抖模块（实现消除按键产生电平来回跳动的影响）

测试方案：用quartus II软件仿真按键消抖模块输出结果。

测试结果：如图3-3-1-1为按键消抖模块仿真结果。

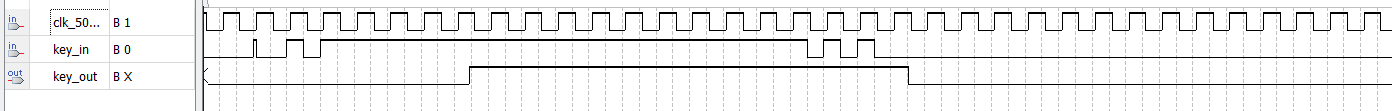


图3-3-1-1

只有在系统按键在一定时间内全是高电平，输出端才会输出高电平。

3.3.1.2 闹钟与时钟显示间切换（实现按一次按键在时钟与闹钟显示间切换的功能）

测试方案：用quartus II软件仿真闹钟与时钟显示间切换输出结果。

测试结果：如图3-3-1-2为按键翻转模块仿真结果。（位选与段选）

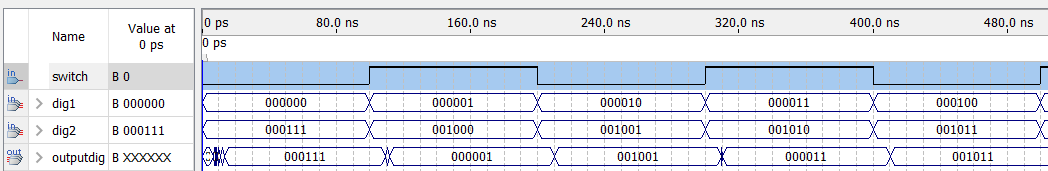


图3-3-1-1

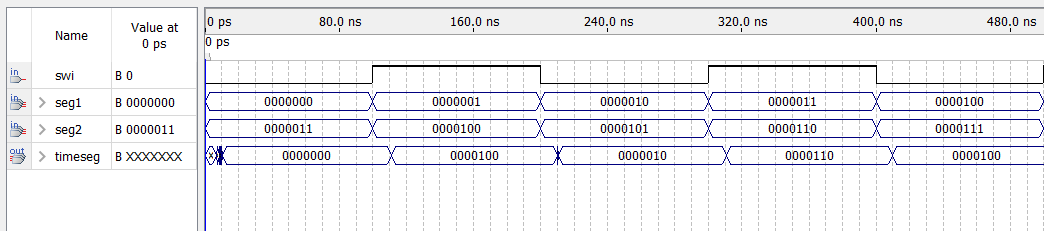


图3-3-1-2

由波形仿真可知，位选与段选在switch开关不同电平时实现了输出的切换，且切换延迟短，功能正常。

3.3.1.3闹铃时间设置

测试方案：用quartus II软件仿真闹铃时间储存与响铃模块仿真结果。

测试结果：如图3-3-1-3为闹铃时间储存与响铃模块仿真结果。

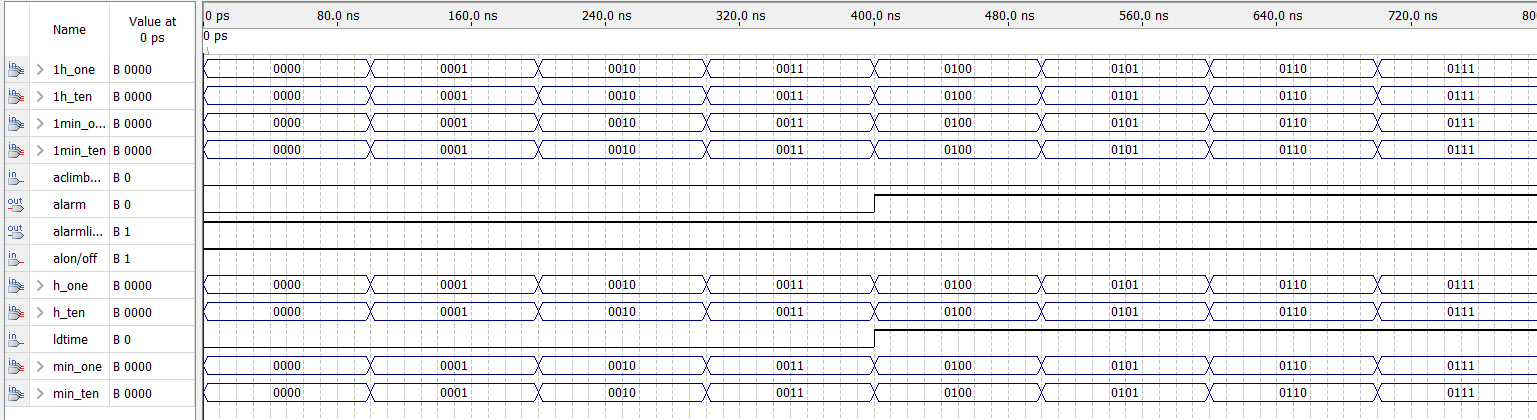


图3-3-1-3

当闹钟功能打开时，alarmlight端口输出高电平，表示闹钟功能打开，当我们ldatime输出高电平时，我们成功设置了闹铃时间，当当前时间跟闹铃时间一致时，alarm端口输出高电平表示闹钟正在响铃。完成闹铃储存与定时响铃功能。

3.3.1.4暂停闹铃与音乐模块

测试方案：用quartus II软件仿真暂停闹铃与音乐模块仿真结果。

测试结果：如图3-3-1-4-1~2系列为暂停闹铃与音乐模块仿真结果。

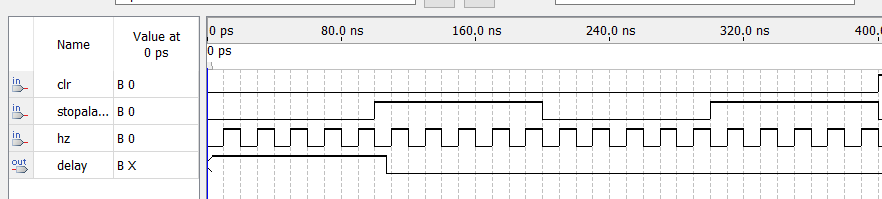


图3-3-1-4-1

由图我们可以看到，当stopalarm端口来了一个高电平要求停止播放闹铃时，输出delay翻转为0，闹铃不再响；如果短时间内（内置计数器计数周期）多次暂停信号来临时，不会对输出结果造成影响，只有计数器进位信号才能使该模块回到初状态；

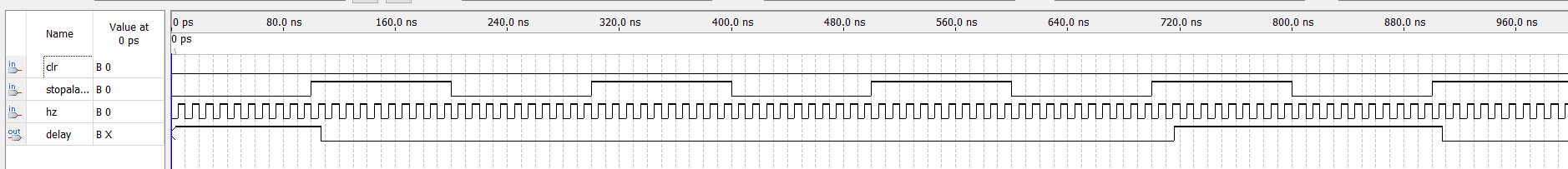


图3-3-1-4-2

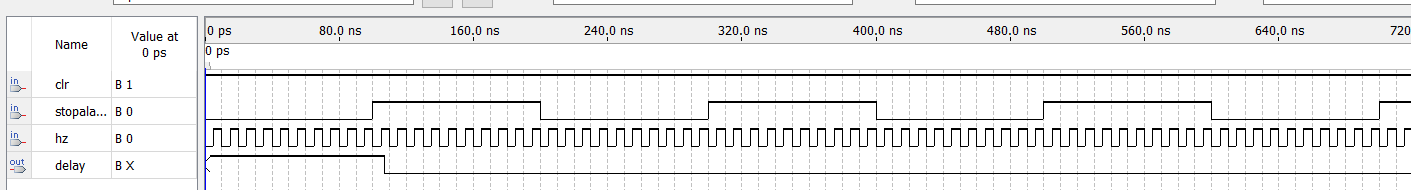
当模块的CLR端口一直处于高电平也就是一直在系统清零时计数模块会清零从头开始，这样的话该模块就能够保持清零后闹钟闹钟暂停效果仍旧会持续一个内置计数器周期。

图3-3-1-4-3

闹铃功能为固话的系统内置音乐，改变了闹钟单调的闹钟提示音，经由与门与闹钟是否响铃端口接与门后输出到蜂鸣器上，完成闹钟暂停与音乐模块功能。

3.3.2 系统整体清零模块

系统清零模块是将手动按键输入的高电平输入到各个计数器的清零端口上，让他们重新计数（位选端跟段选的内置计数器不需要），各个计数器清零效果已在上边叙述，此处不再展示；只对系统闹钟的一个问题做出解决，也就是如果闹钟功能打开后，如果清零将会导致闹钟设置的时间与系统当前时间相同都为00：00，闹钟会立即发出闹铃声，因此我们对闹铃开关做出了改动，人可以通过按键手动控制闹钟开关；闹钟功能开启后，系统整体清零时，我们这个模块将会对该信号做出处理，使该模块输出翻转，是的闹钟功能自动关闭，并且在短时间内多次清零信号到来时只对第一次信号做出处理，而且接下来的几次清零信号会重置内部计数器，使得该模块仍旧处于工作状态，只有等内部计数器完成一个计数周期后才能使该模块恢复初始状态。该模块类似于闹钟延时模块，除去了自动翻转恢复初始状态的功能，正好达到了清零后闹钟关闭，再次开启需要手动开启的功能，此处不再给出波形图。

3.3.3 整点响铃

测试方案：用quartus II软件仿真整点响铃模块仿真结果。

测试结果：如图3-3-3为整点响铃模块仿真结果。

整点响铃的功能借鉴了前面的闹钟延时模块，本来我的延时模块在一般时候输出为0，一旦有电平的变化输入，就会将其激活，并且维持激活状态两个时钟脉冲的周期，波形图如下：

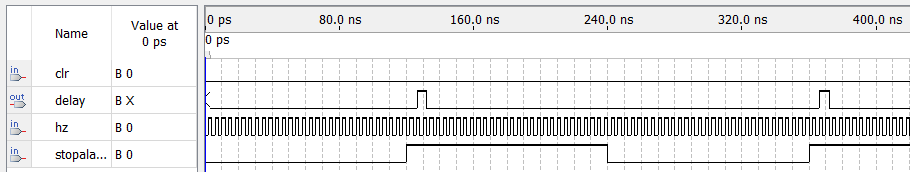


图3-3-3

从图中我们可以看到，当stopala端口输入一个变化点电平后会使delay输出翻转，且持续了两个时钟脉冲hz周期，达到设计要求。

* 1. 系统总体测试

表1 系统总体测试结果记录表

|  |  |  |
| --- | --- | --- |
| 测试内容 | 测试方案 | 测试结果 |
| 秒计数 | 时钟连接1Hz，观察秒计数结果是否能够从00-59正常计数，且能够正常向分进位。 | 正常 |
| 分计数 | 时钟连接100Hz，观察分计数结果是否能够从00-59正常计数，且能够正常向小时进位。 | 正常 |
| 小时计数 | 时钟连接1KHz，观察分计数结果是否能够从00-23正常计数，且能够正常归零。 | 正常 |
| 时钟校时 | 按下时钟校时按键，选择校时位，按键加时间，观察能否正常校时。 | 正常 |
| 闹钟 | 按键切换到闹钟显示模块，按下时钟校时按键，选择校时位，按键加时间，观察能否正常设置闹钟时间；设置恰当的闹钟时间，观察闹铃会不会准时响；按下闹钟暂停按钮，观察闹铃会不会暂停，然后隔一段时间再次调整时间，闹铃会不会继续准时响。 | 正常 |
| 系统整体清零 | 观察当系统清零按键按下后系统的时钟跟闹钟显示是否都已清零，在闹钟功能开启的状态下清零看是否闹钟功能自动关闭，观察多次清零信号会不会对闹钟功能的开启与关闭造成影响 | 正常 |
| 整点响铃 | 打开整点响铃功能，观察当前时间（小时）每达到一个整数时间时，会不会响铃（初步设计为响铃两秒），再观察闹钟时间与整点时间响铃冲突时，谁会更优先（设计为闹钟优先级更高）。 | 正常 |

1. 系统设计实现过程中遇到的主要问题、解决思路和解决方案
2. 在闹钟与时钟显示模块切换的设计过程中出现了小时位不显示或者显示跳动的问题，通过更换绑定引脚发现不是问题所在，再次检查电路，并且请教老师，更改增加了一个位选的二选一模块，也就是在时钟显示与闹钟显示之间将两者的数据（包括位选与段选）做出彻底的二选一，两组数据不做出任何的交集，避免端口冲突；重新设计后解决了该问题。
3. 刚开始设计时没有考虑到按键不消抖会对系统造成何等影响，导致我的校时按键增加位数远不止加1，后面请教老师以及查阅书籍后发现了按键消抖模块，成功解决该问题。
4. 系统整体清零后会导致闹钟立马响铃。发现时清零后闹钟功能开启时，系统时间与闹钟时间相同，同为00：00，导致响铃，因此重新设计了一下闹钟功能开启按键，改为key按键，内部设置为翻转模块，并且在清零后能够使闹钟功能自动关闭，并且需要手动开启，该功能已经完善且工作正常。
5. 系统内部有些接口报warning，说内部端口时钟与GND或者VCC连在一起，请教老师，发现有可能是逻辑设计错误，老师指导后发现确实是内部电路有的端口接错线路，整改线路后工作正常。
6. 心得体会

通过数字电子时钟的设计，让我对电子信息专业产生了浓厚的兴趣，在学习与设计过程中不断查阅资料、询问老师，让我的数字电路与逻辑设计这一门课程的知识得到了进一步的扩展与巩固，尤其是自己的动手能力得到了很大的提升，逻辑思维能力、分析推理能力也得到了很大的进步，虽然说我的设计有可能比不上其他人的设计，但是我的设计完全是自己一步一步摸索，分析设计而做出的，最重要的是我得到了老师的认可与指点，老师的意见是非常宝贵的，老师的鼓励也是我们进步的动力！