**数字钟说明书**

本数字钟共有6个按键，分别通过键盘上的1、2、3、M、H、C进行控制。

|  |  |
| --- | --- |
| **开关1** | **功能：数码管显示切换**  当开关在上时为小时-分，当开关在下为小时-秒。  在主界面添加了一个显示灯，直观表示当前显示模式。  在秒表模式下，是小时-分和秒-百分秒切换按钮。 |
| **开关2** | **功能：设置模式和普通计时模式切换**  当开关在下时为普通计时模式，开关在上时为设置模式。  在设置模式中，可以进行小时、分和星期的调整。 |
| **开关****3** | **功能：秒表模式和普通计时模式切换**  当开关在上时为计时模式，当开关在下时为秒表模式。  切换秒表模式后，需要进行初始化。  在主界面添加了一个显示灯，直观表示当前模式。 |
| **按钮M** | **功能：设置模式下设置分钟；初始化置0**  在设置模式下，短按按键可以设置分钟（每次加1）。  在所有模式下，M和H同时按下可以初始化数字钟。  初始化表现为：时间设置为00:00:00，星期设置为1。 |
| **按钮H** | **功能：设置模式下设置小时；初始化置0**  在设置模式下，短按按键可以设置小时（每次加1）。  在所有模式下，M和H同时按下可以初始化数字钟。  初始化表现为：时间设置为00:00:00，星期设置为1。 |
| **按钮C** | **功能：计时模式、设置模式下设置闹钟**  在上述两个模式下的任何时间，短按按键可以将那一刻设置为闹钟，通过一个LED灯进行显示。 |

除了上述提到的状态指示灯，本数字钟还有如下LED/数码管进行显示：

|  |  |
| --- | --- |
| **时间显示数码管** | **功能：计时模式、设置模式、秒表模式显示时间**  计时模式：显示小时-分或小时-秒  设置模式：显示小时-分或小时-秒  秒表模式：显示小时-分或秒-百分秒 |
| **整点显示LED** | **功能：整点报时**  当时间为7:00:00-23:00:00的整时时，上LED亮；  当时间为上述时间之前4秒（如6:59:56-6:59:59），  下LED亮 |
| **星期显示数码管** | **功能：显示星期**  可以使用M、H进行初始化 |
| **闹钟显示LED** | **功能：到达闹钟时间后点亮** |

**一、设计要求**

在经过斟酌之后，本数字钟选择尽力达到以下要求：

**（一）基础要求**

（1）利用数字芯片，设计一个简易的数字钟。电源电压5V，用四个数码管正确显示时、分，切换显示秒，并能对分、时进行调整。

**（二）进阶要求**

（1）整点报时：从上午7:00~晚上11:00，要求从59分56秒开始到0分0秒，0分0秒的提示跟其他4秒要区分开。

（2）闹钟功能：设置闹钟，到达对应时间后给予反馈。

（3）显示星期。

（4）秒表功能：正常显示为秒和百分之一秒，通过切换显示小时、分钟。

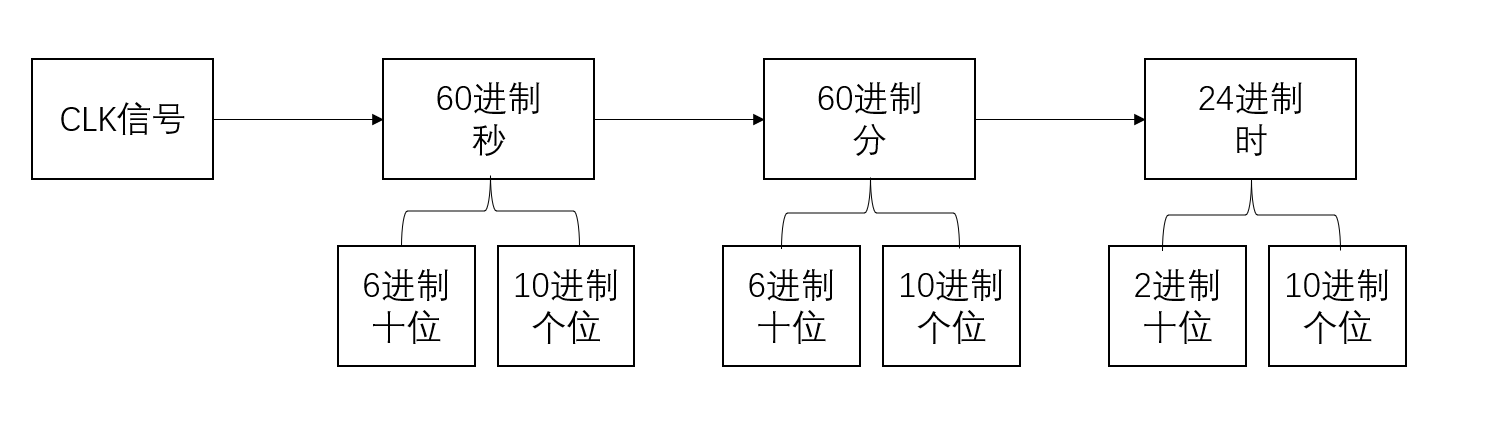
（5）显示切换。

**二、设计方案**

我将按照上述要求的顺序进行设计方案的讲解。

**（一）基础要求**

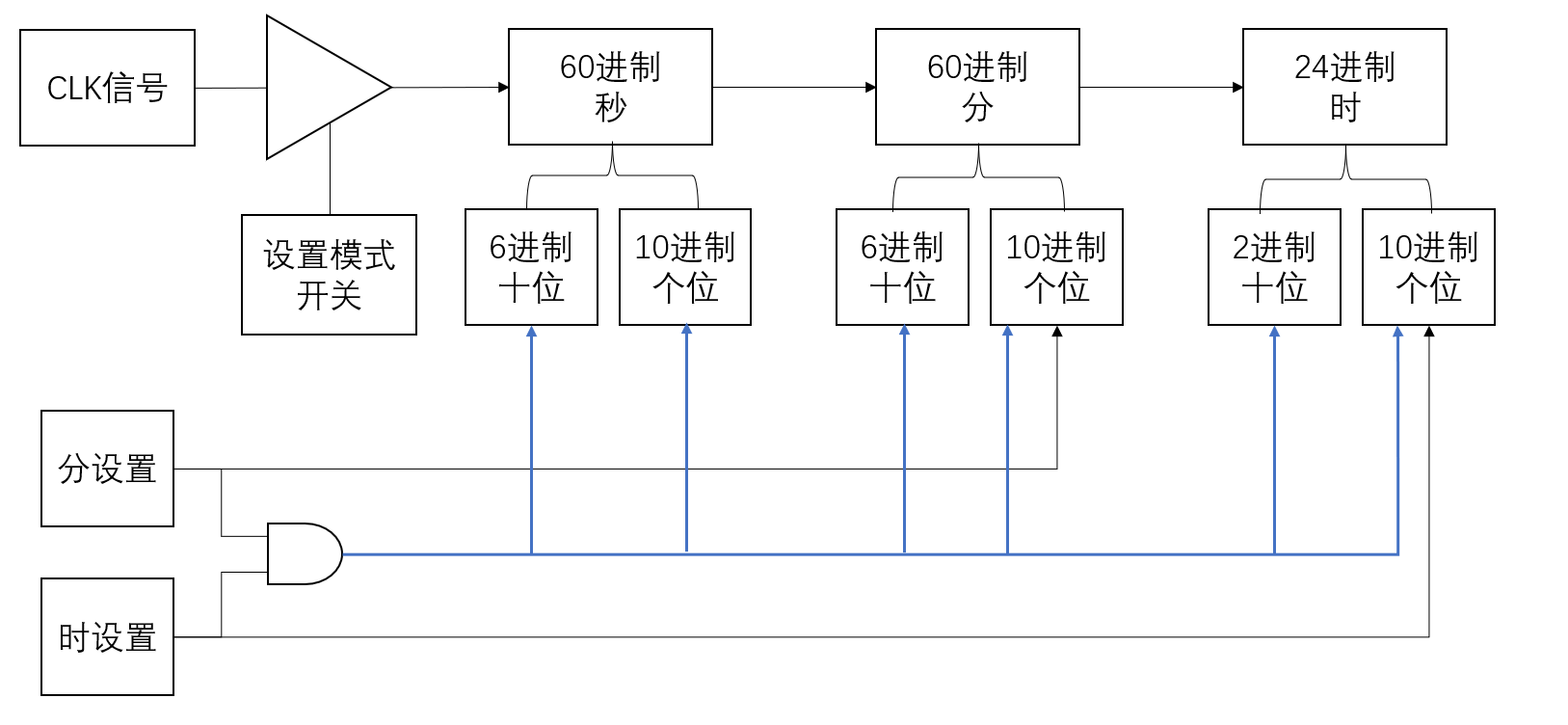
选择TTL系列的芯片与逻辑门。

**（1）计时功能**

设计框图如上，计数器选择74192N芯片，功能完备。

当时计数到24时，输出置0信号，将所有计数器设置为0.

显示功能即引出Q­3Q2Q1Q0到数码管即可，数码管选择DCD\_HEX，避免经过译码器等器件，减少卡顿。

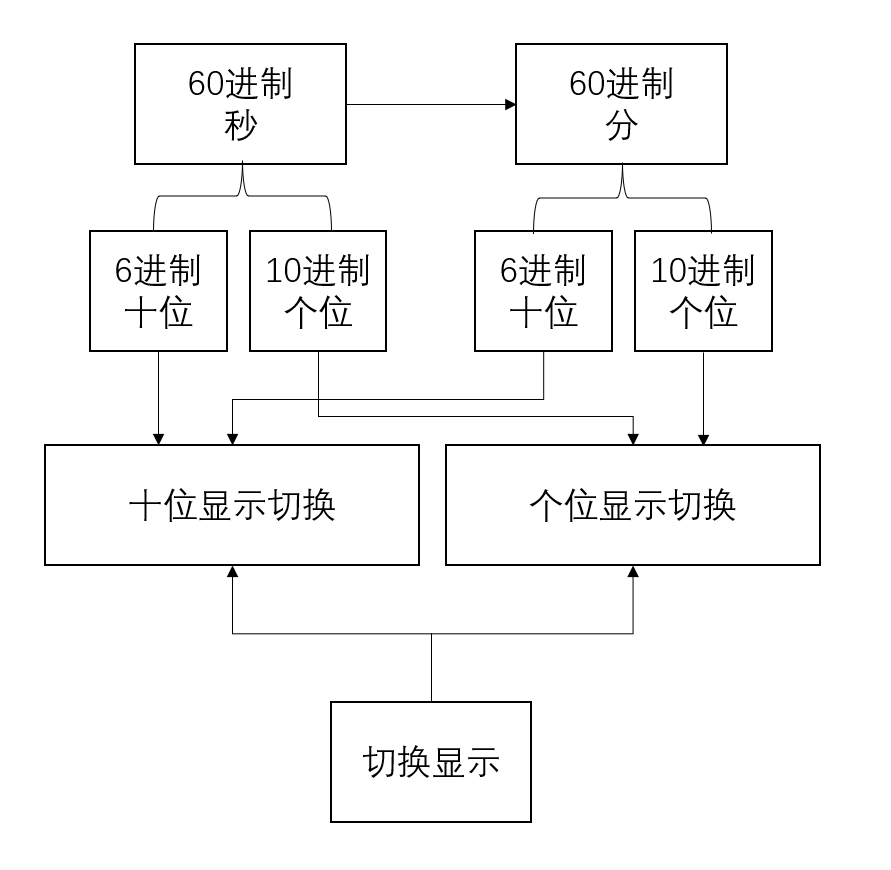
**（2）设置功能**

设置时间时时间需要静止，因此选择断开CLK信号。

通过两个按键分别设置分和时。

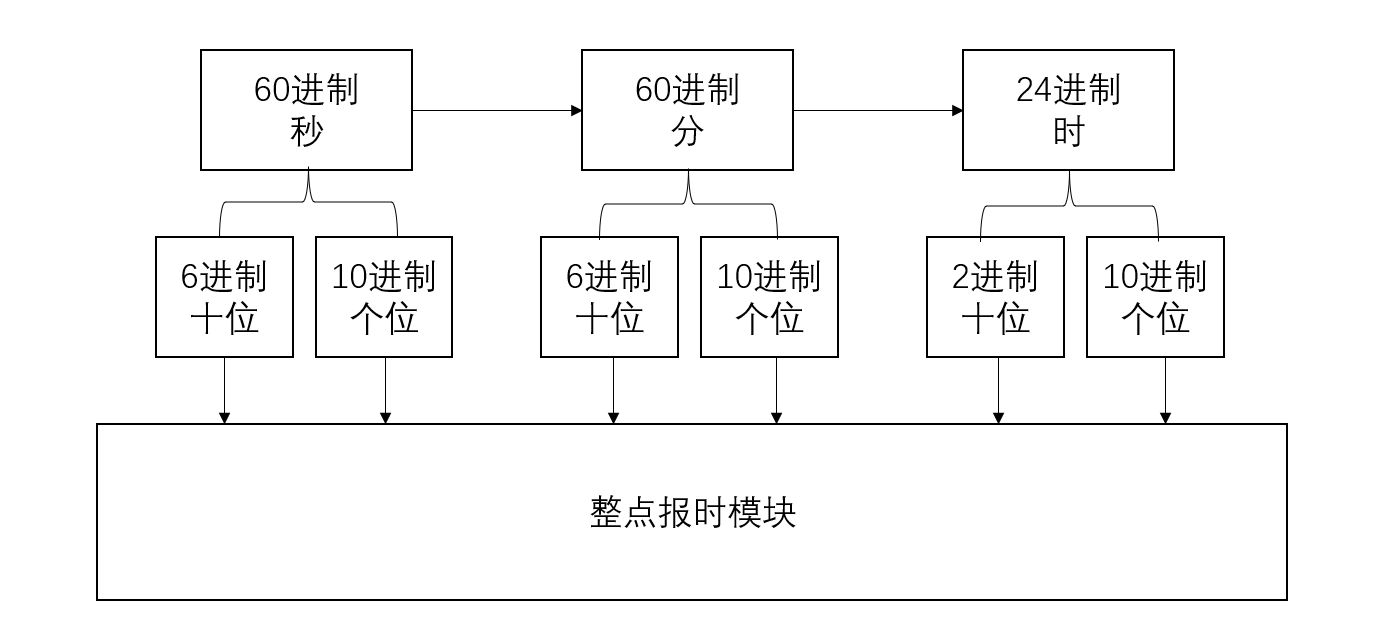
考虑到实际情况，当分、时设置按钮同时按下时，初始化信号（蓝色）被激活，将六个计数器全部设置为0。

**（3）显示切换**



通过将分、秒的十位和各位分别接入显示切换模块中，用一个切换显示信号进行控制。

**（二）进阶要求**

**（1）整点报时**

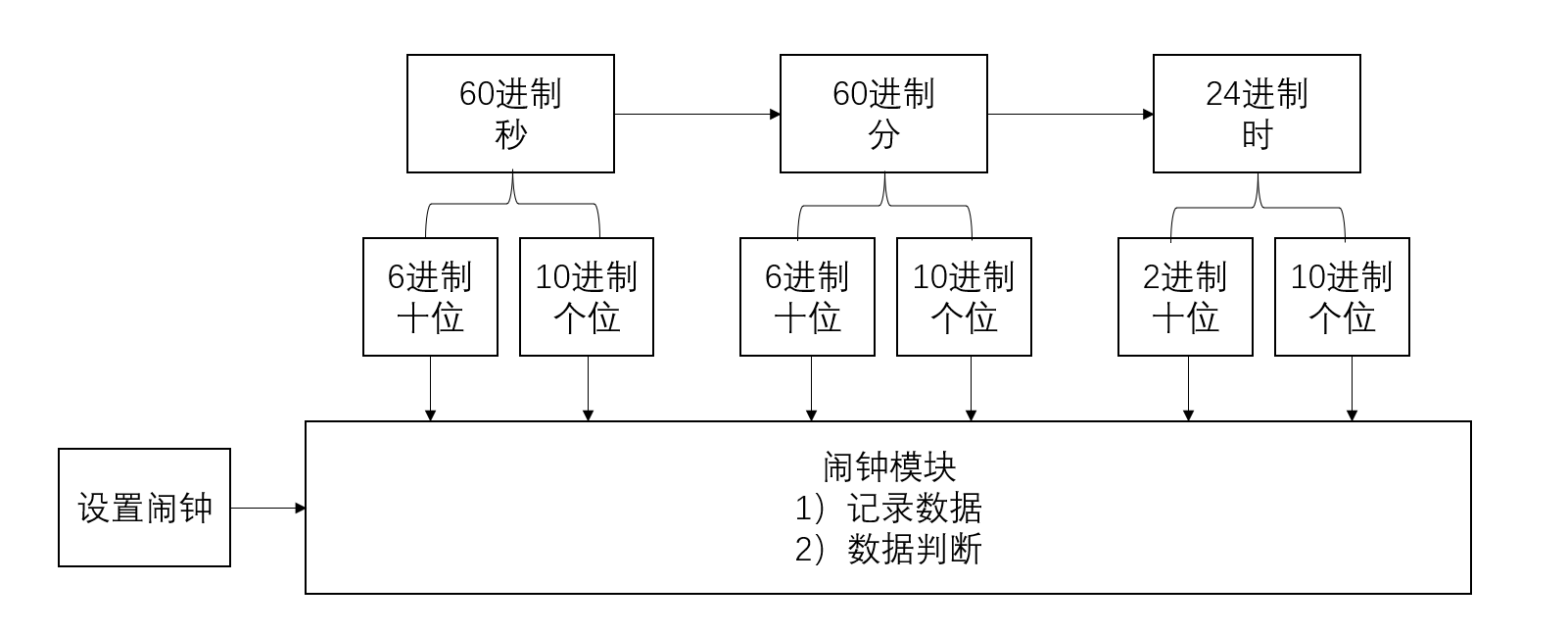
模块内进行两种类型的判断：类型一是59分56秒至59秒的非整时判断，类型二是0分0秒的整时判断。

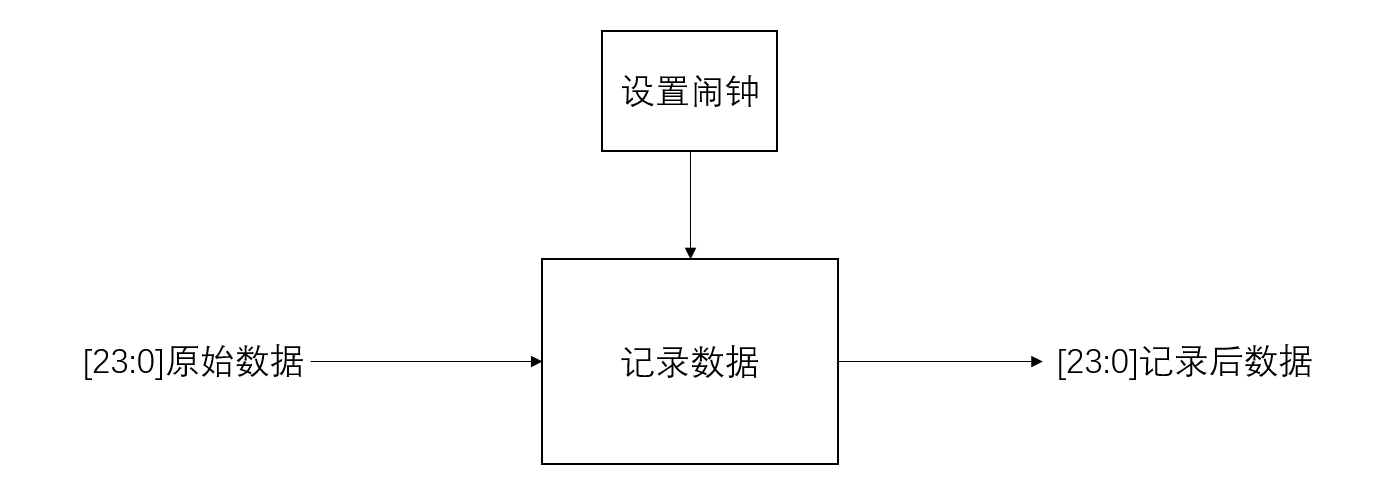
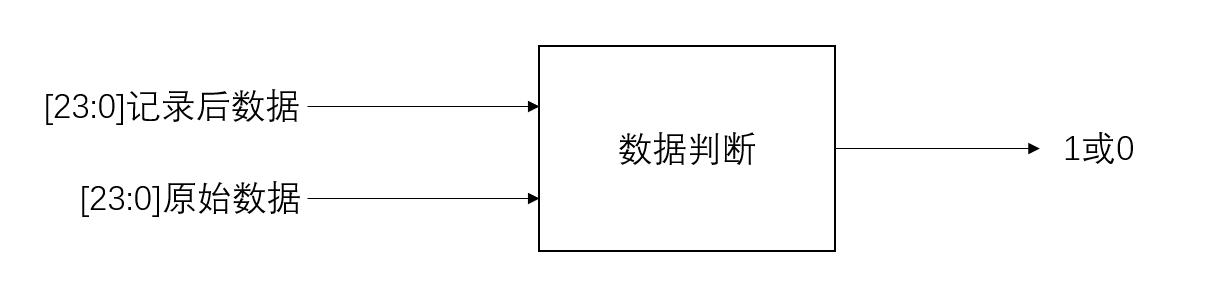
【时】选择6-23的数据，其中6用来做类型一的判断，7-22用来做类型一和类型二的判断，23用来做类型二的判断。

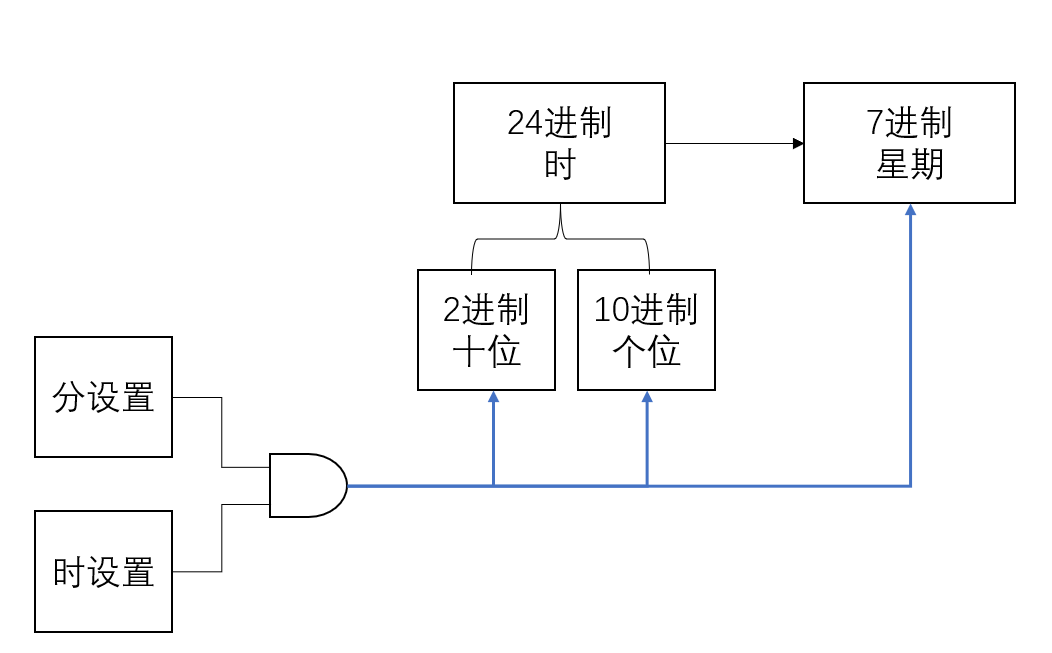
【分】选择59的数据。

【秒】选择56-59的数据。

上述三者选择后的信号（1或0）经过与逻辑之后，即可实现时间判断。

**（2）闹钟功能**

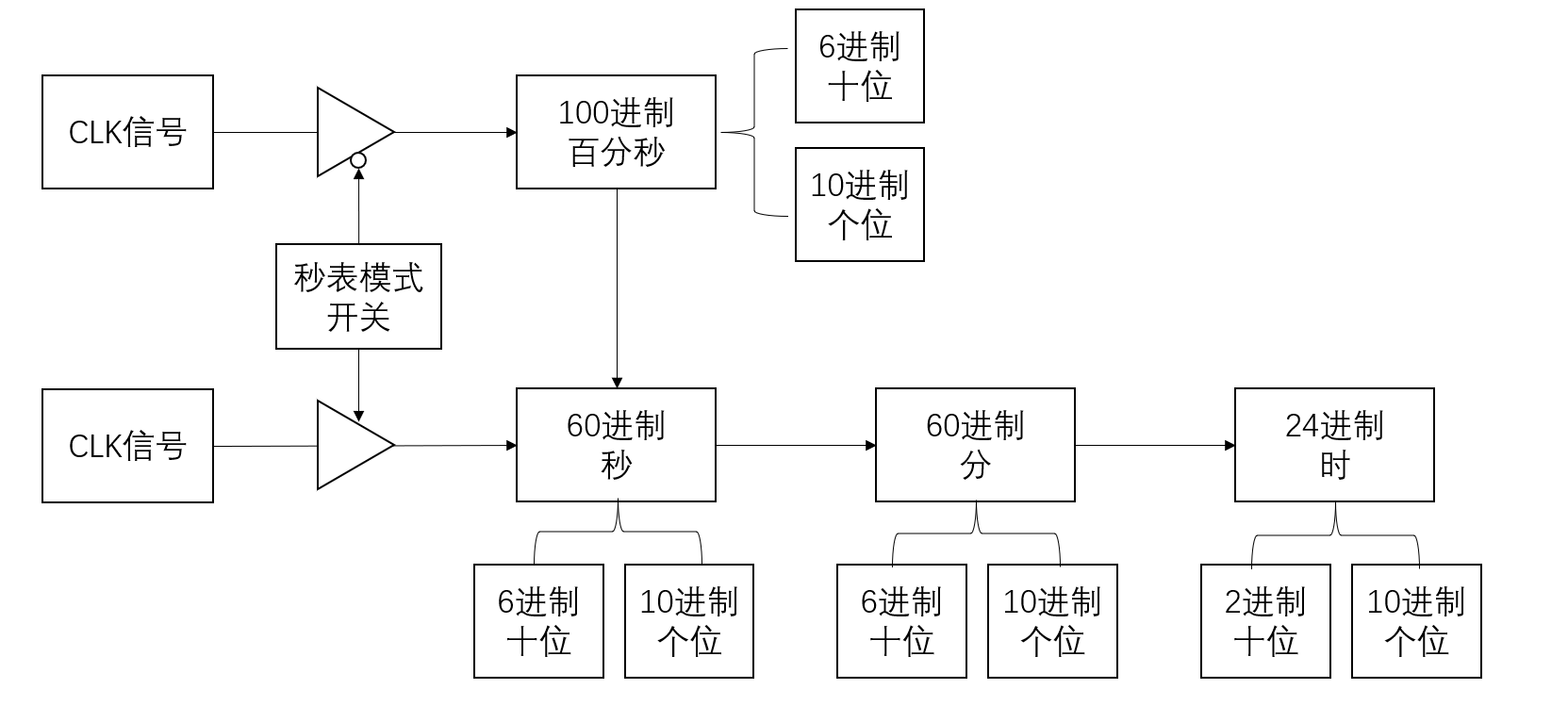
设置闹钟按键按下时，闹钟模块的“记录数据”被激活，通过D触发器将数据全部记录；随后，“数据判断”持续进行，记录后数据通过和原始数据进行同或操作，全部相同时输出1，闹钟模块生效。

**（3）显示星期**

较为容易，设计一个7进制计数器即可。

为了符合逻辑，将先前的初始化信号引入到星期中，同时按下则置1。

**（4）秒表功能**

**a.切换模式功能**

使用一个开关进行模式切换，同时设计一个100进制百分秒计数器。

**b.显示切换**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 开关1 | 开关3 | 数码管3 | 数码管2 | 数码管1 | 数码管0 |
| 0 | 0 | 秒  （十位） | 秒  （个位） | 百分秒  （十位） | 百分秒（个位） |
| 0 | 1 | 时  （十位） | 时  （个位） | 秒  （十位） | 秒  （个位） |
| 1 | 0 | 时  （十位） | 时  （个位） | 分  （十位） | 分  （个位） |
| 1 | 1 | 时  （十位） | 时  （个位） | 分  （十位） | 分  （个位） |

开关为00时，秒表模式-显示秒和百分秒；

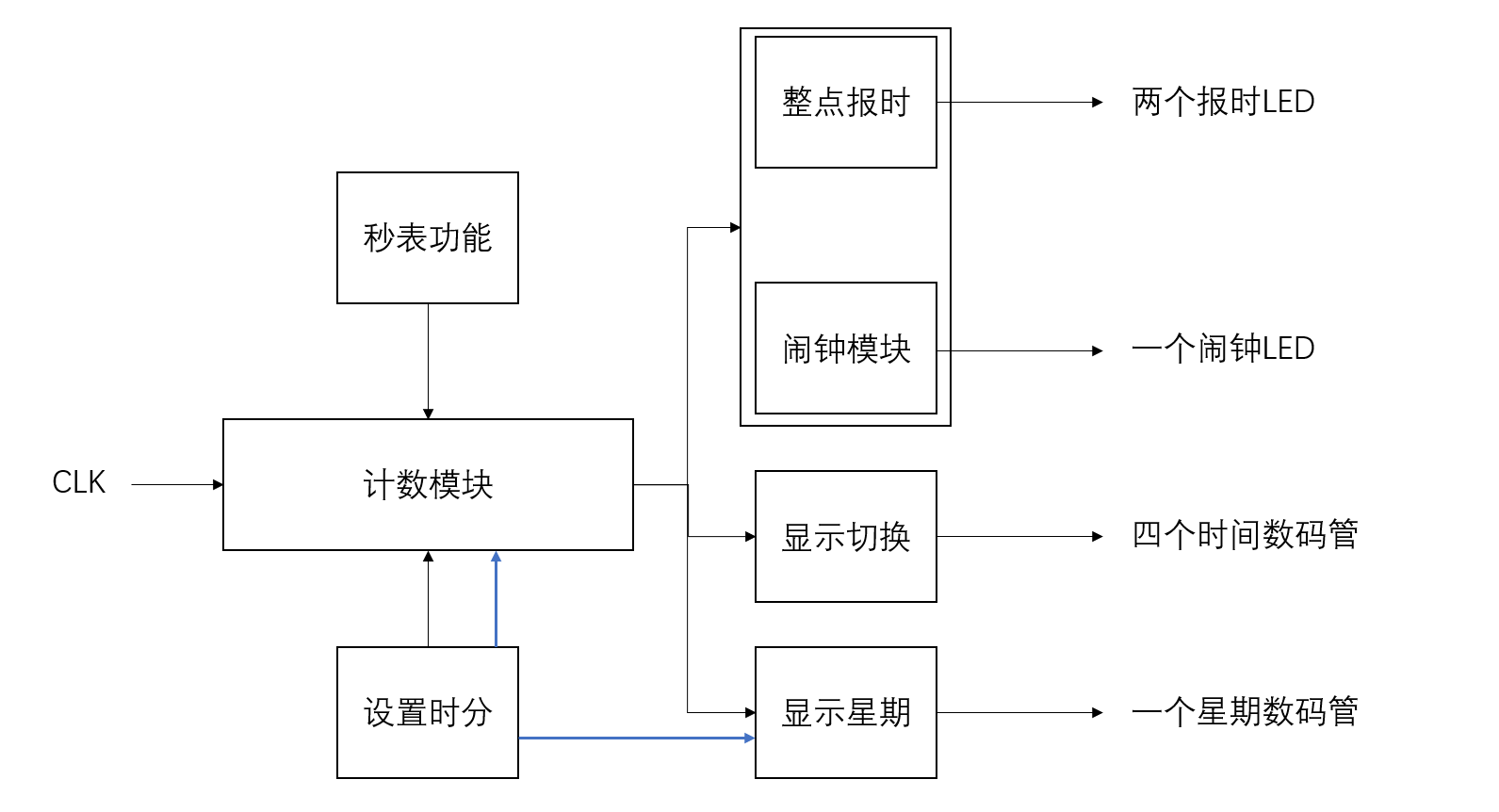
开关为01时，计时模式-显示时和分；

开关为10时，秒表模式-显示时和分；

开关为11时，计时模式-显示时和分。

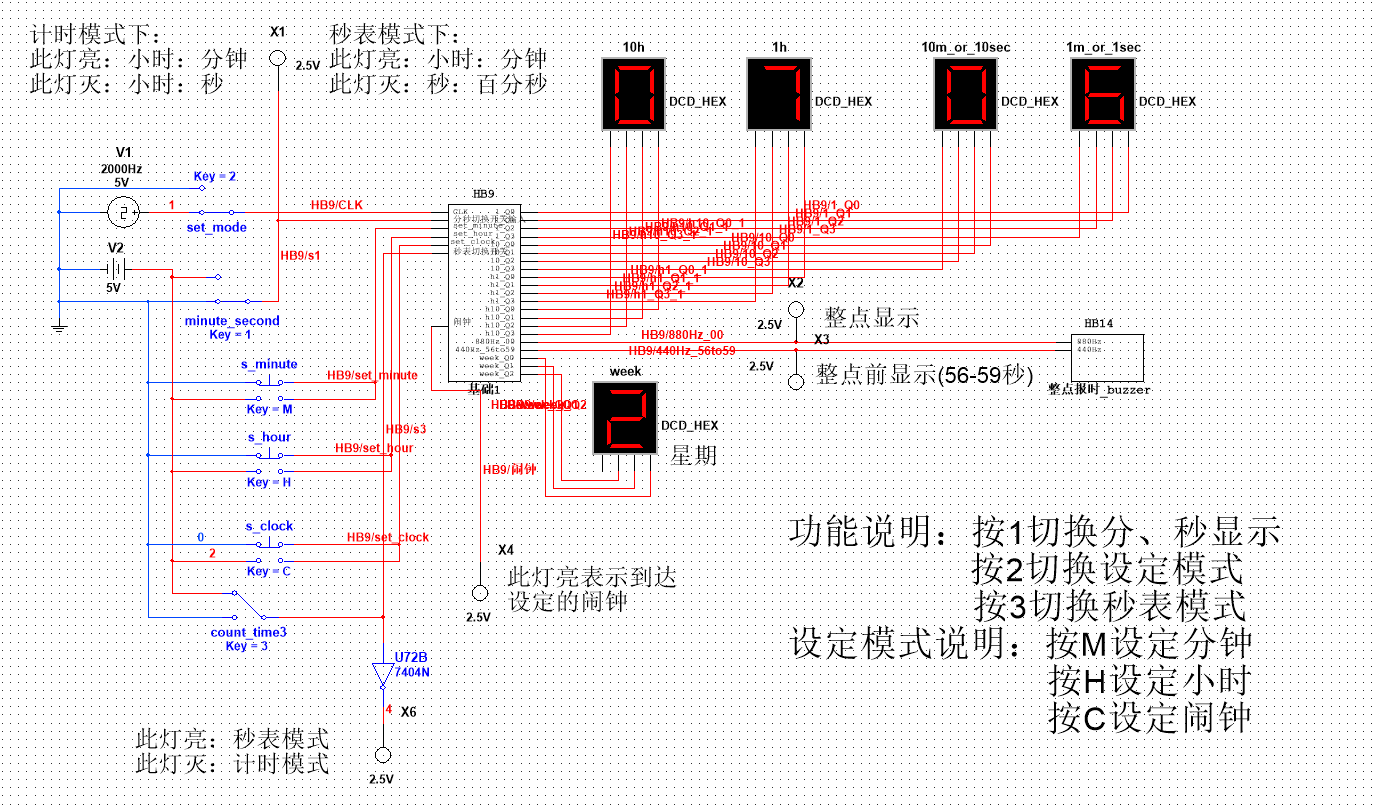
按照上述思路进行设计即可。

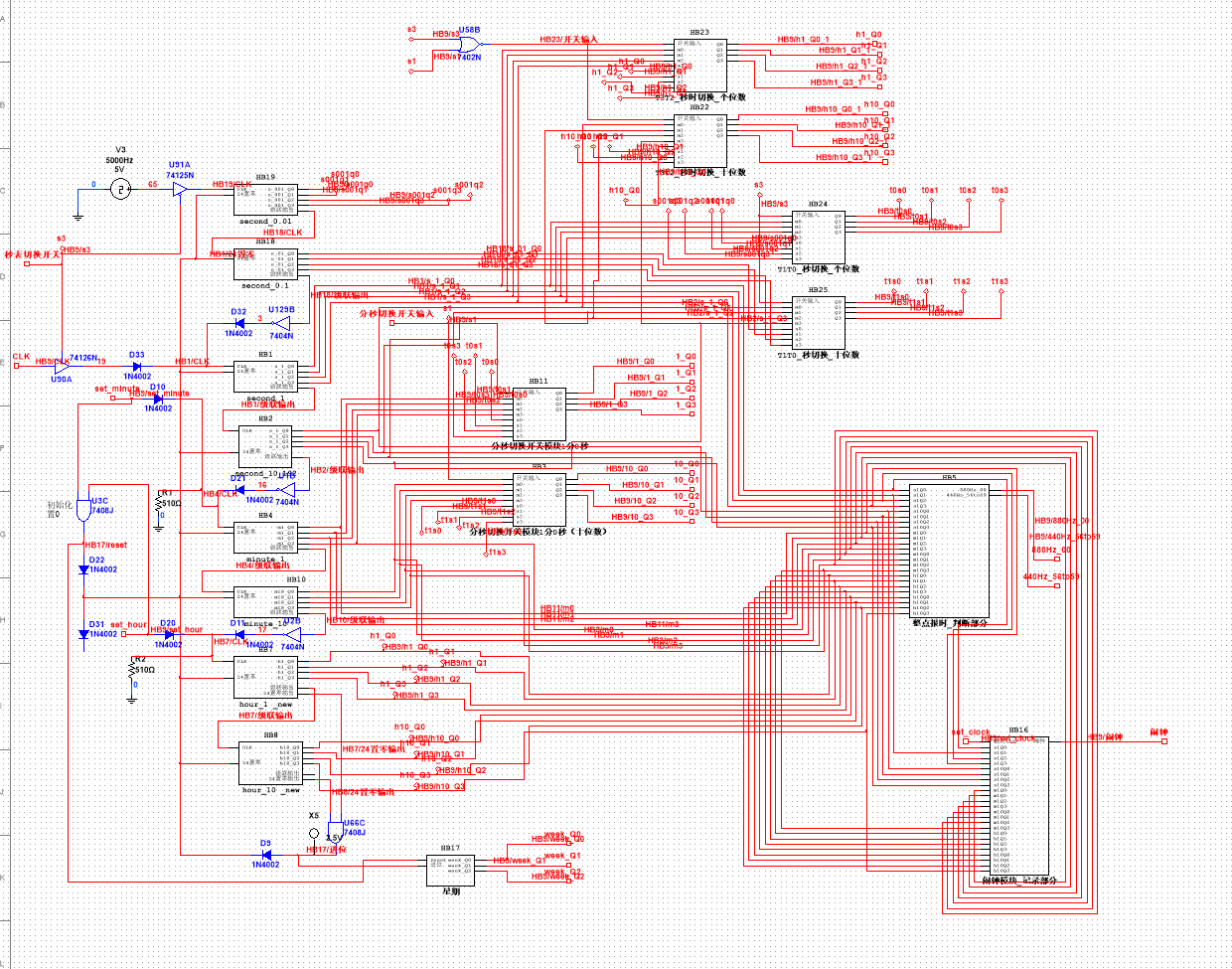
**三、总体设计框图**

以上介绍了每个功能的分模块功能与详细框图。进行归纳，有如下总框图：

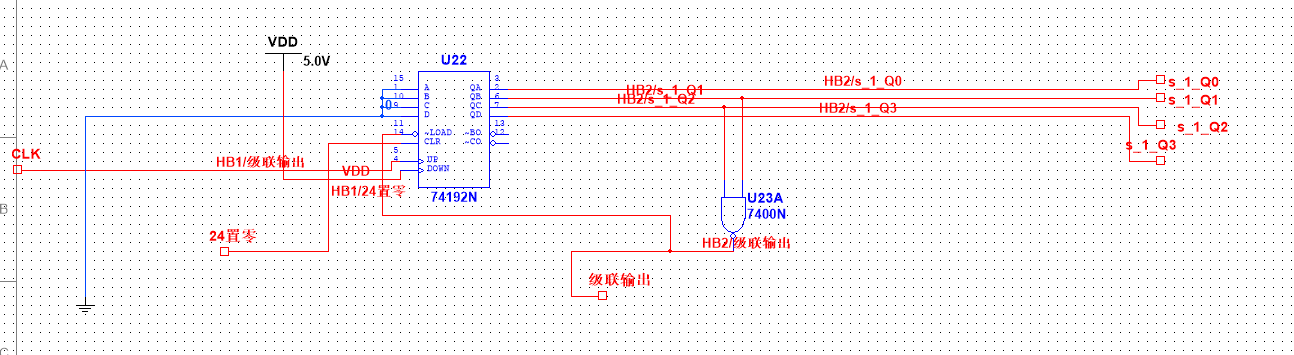
以上框图展现了数据的流动方向。

**四、模块电路截图**

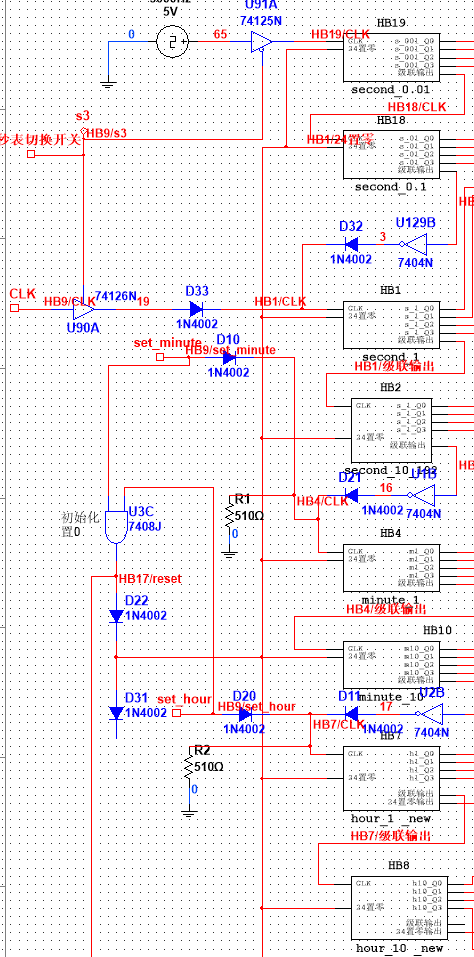
**（1）用户界面**

**（2）数字钟内部**

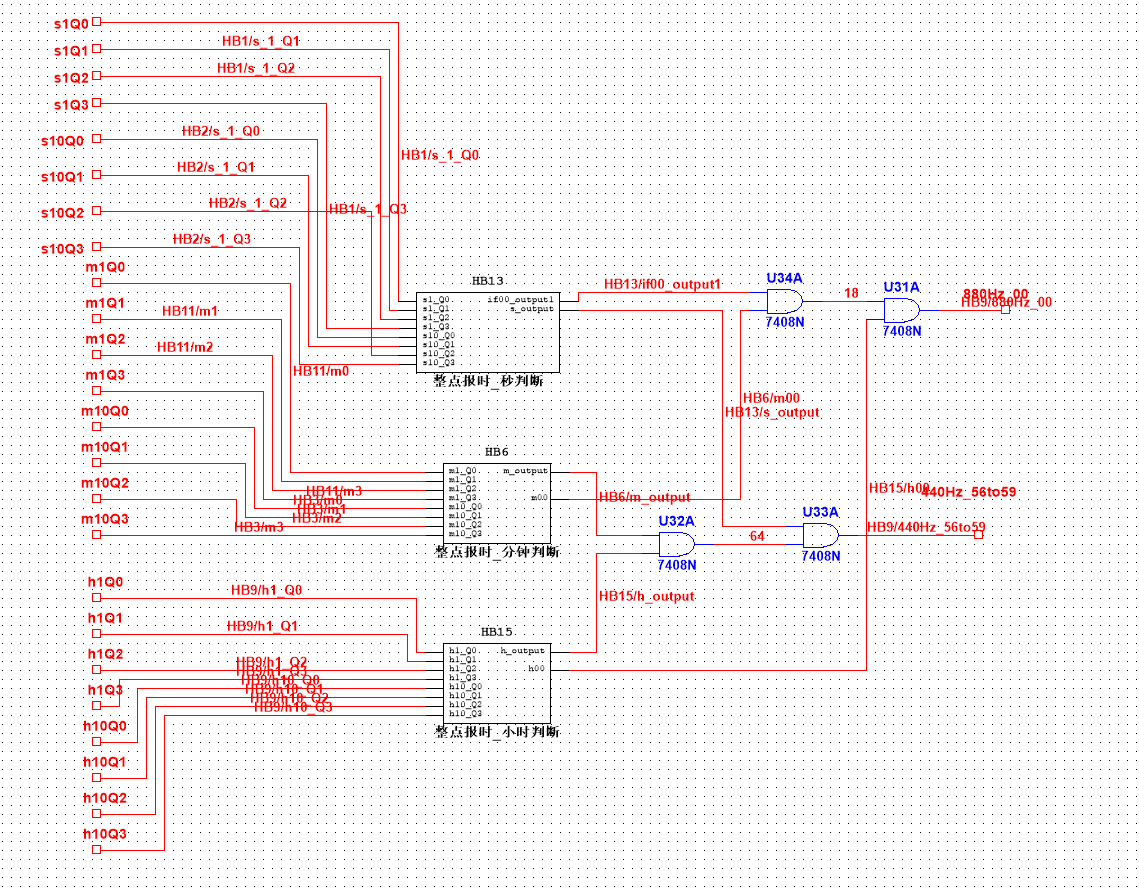
**（3）计数模块与设置时分**

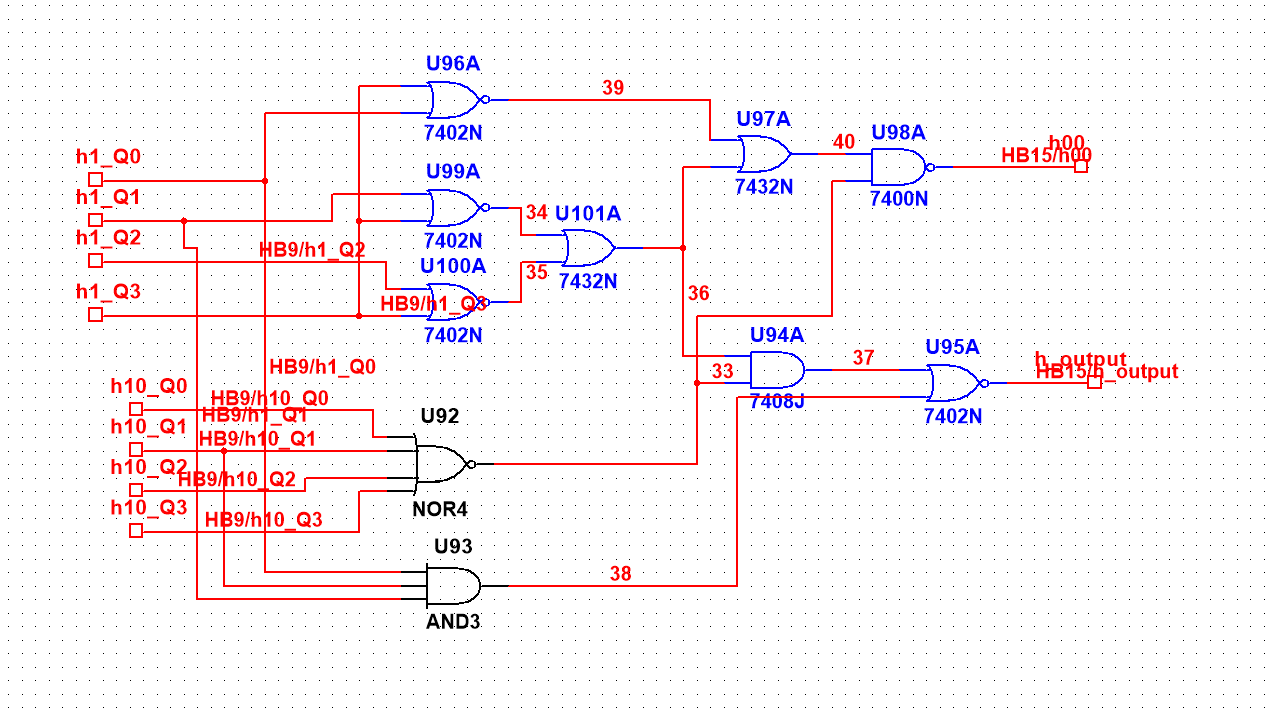
以秒的十位为例，小计数器：

计数器的级联与设置时分：

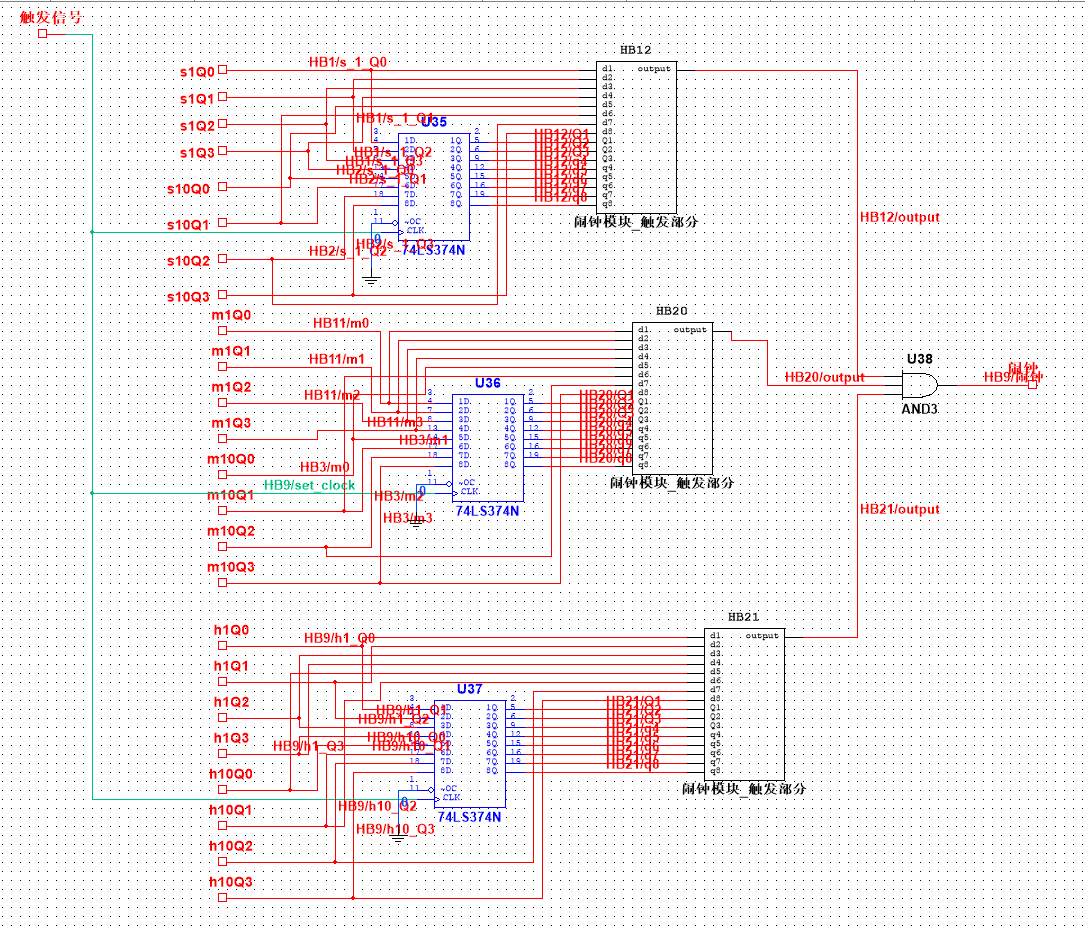


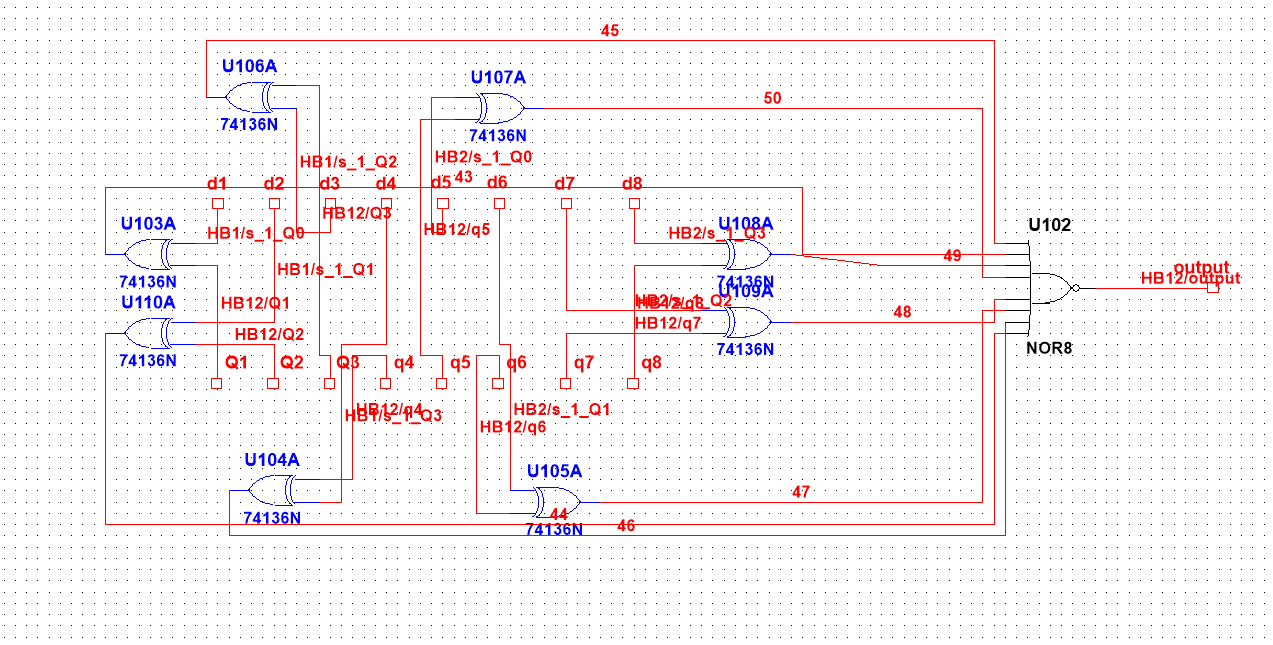
**（4）整点报时模块**

模块内部：

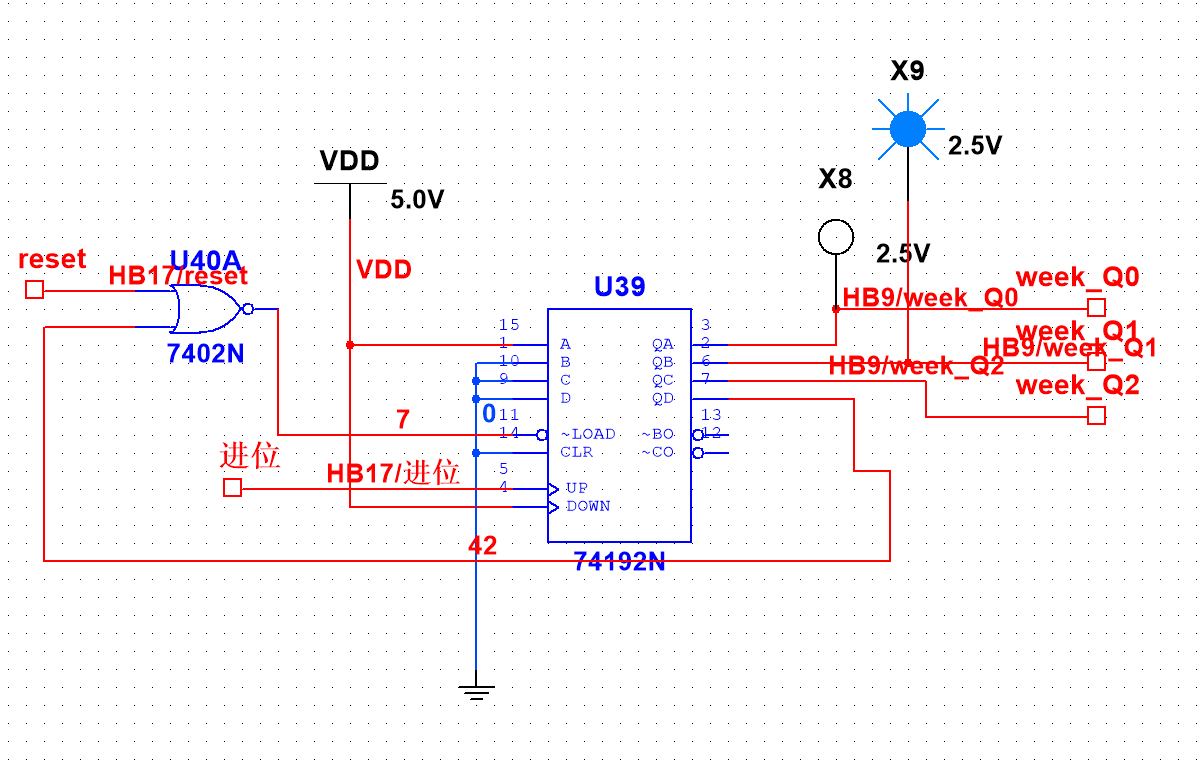
****判断模块（以时模块为例）：

**（5）闹钟模块**

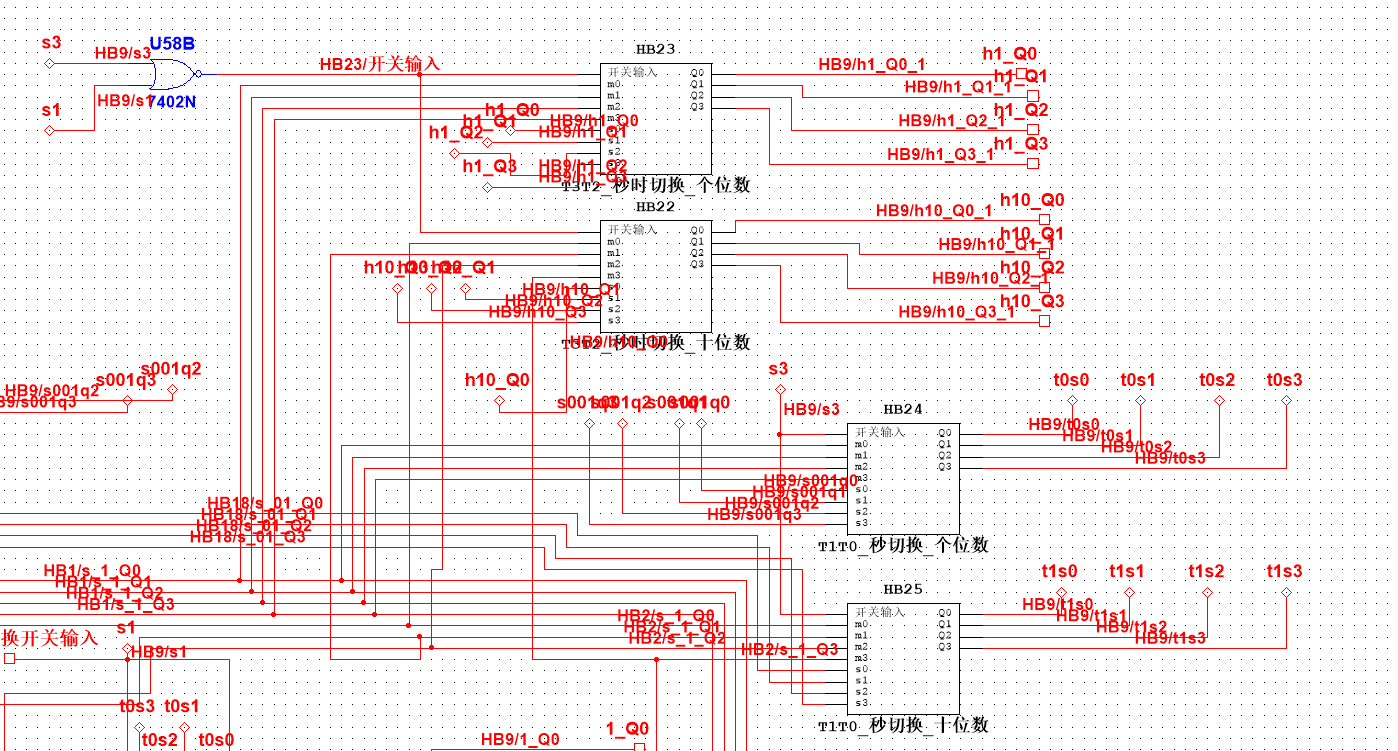
记录部分+判断模块：

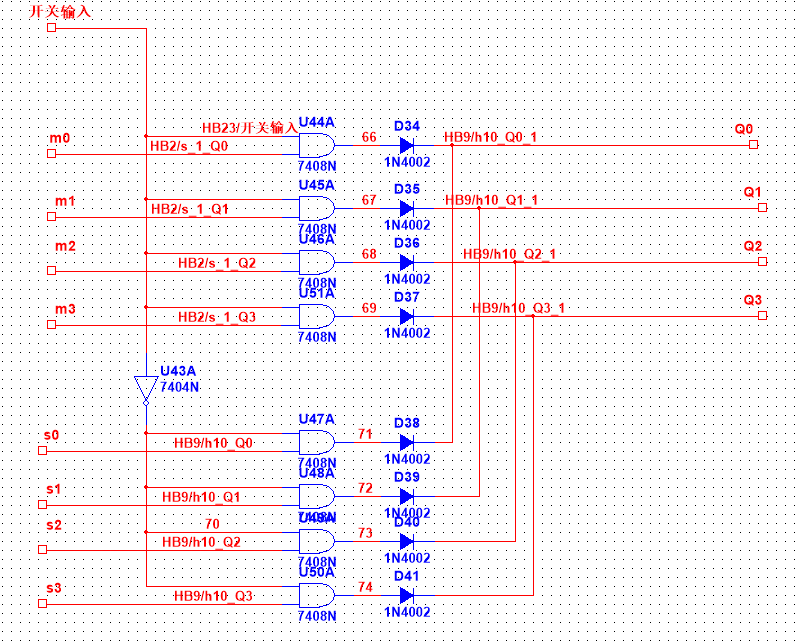


判断模块：

**（6）显示星期**

**（7）设置切换（秒表功能）**

秒表功能并没有设计一个单独的模块，但是秒表功能的计时部分在之前的计数模块中已经体现，现在展示显示切换模块外部：

切换模块内部：

当然，仅靠截图是无法展现具体接线细节的，还是建议在原文件中进行查看。

**五、系统运行情况与改进思路**

**（一）基础要求**

**运行情况：**总共使用了4个按键，几乎完成了所有任务，美中不足的是在切换显示秒的时候，数码管高二位显示的是小时，违背了现实逻辑。同时，老师批评进入设置模式是把CLK信号切断，不太妥当，我个人认为“切断CLK”这件事本身是可以包装的更好的，例如将CLK信号和设置模式信号经过一个与门，设置模式为1的时候CLK正常输入，为0的时候CLK则无法输入。

**改进思路：**可以仿照秒表模块的显示切换，把高二位数码管再做一个时/分显示切换，这一点是完全可以实现的，最大问题是麻烦。

**（二）进阶要求**

**（1）整点报时**

**运行情况：**正常运行，只是Multisim的蜂鸣器有些问题，只能改用灯。

**（2）闹钟功能**

**运行情况：**正常运行。

**改进思路：**目前只是让闹钟在闹钟时间那一秒有LED输出，可以做一个触发电路（如计时器）让LED输出更久。

**（3）显示星期**

**运行情况：**正常运行。

**改进思路：**没有和设置时间进行隔离，依靠设置小时到24进行进位，效率过低。整个设置时间的模块完全可以用两个按键解决：一个按键进行“进入设置模式”和“选位”的需求，另一个按键则进行选定位置的修改。

**（4）秒表功能**

**运行情况：**功能本身正常运行，但是进入秒表模式后，外部的计时模式就被终止，同时秒表的清零依托于初始化，牵连整个系统。

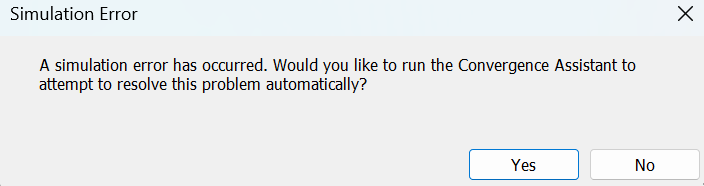
**改进思路：**直接复制一个新的计数模块，两套系统并行运行，实现彼此的隔离。

**（5）显示切换**

**运行情况：**正常运行。

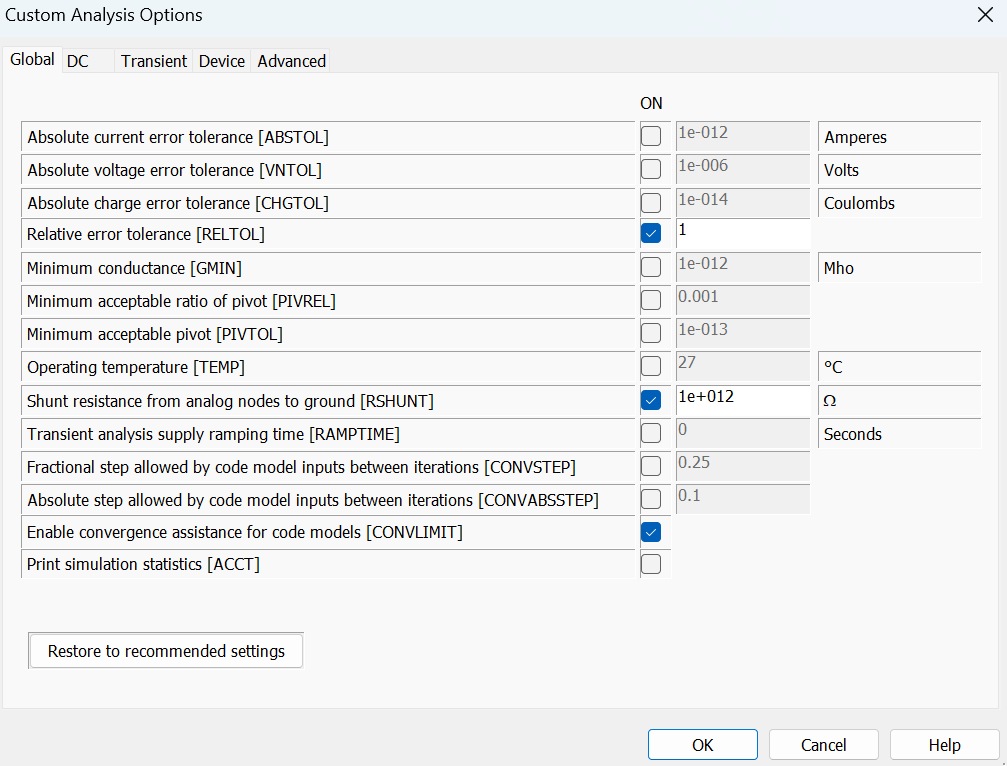
**六、实验中故障现象与解决方法**

实验中的故障数不胜数，但极大多数都是简单问题导致，容易排查。接下来，我分享两个或是逻辑无问题、或是难以排查的故障，以及它们的解决方法。

**（1）系统报错**

在设计初期就已经出现，但是无论怎么排查电路都没有问题。在互联网上搜索后，我根据这条帖子进行了设置，解决了问题。

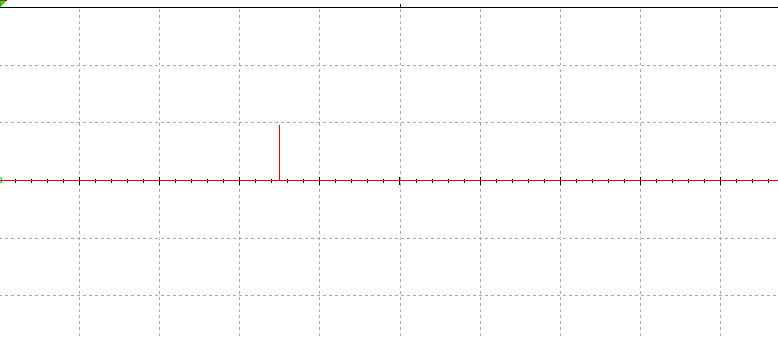
链接：<https://blog.csdn.net/mahoon411/article/details/117127050>

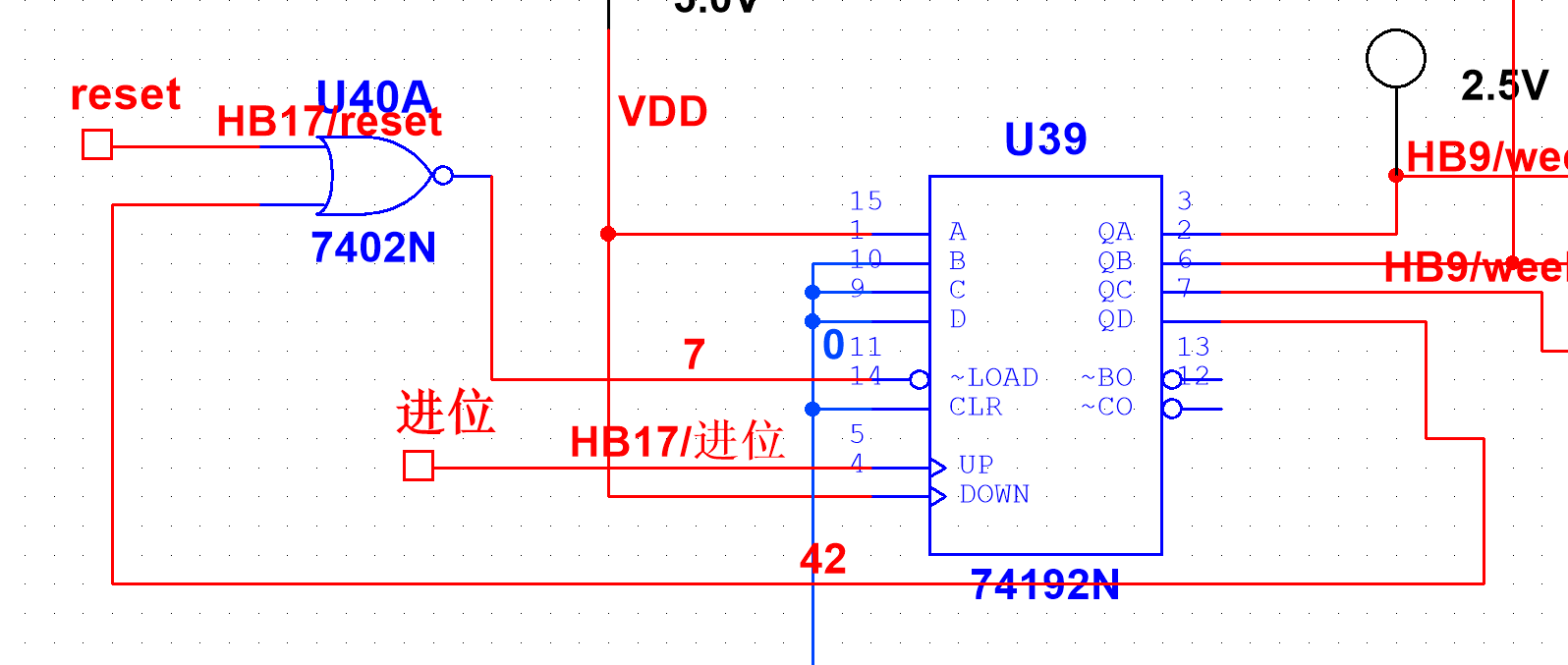
即将Relative error tolerance进行自定义设置，我设置为1，经过测试，1以下系统还算比较稳定，再往上系统会出现一些比较离奇的问题，例如一个CLK信号触发两次上升沿等。

推测是某部分电路逻辑上合法、但Multisim仿真不合法所导致。

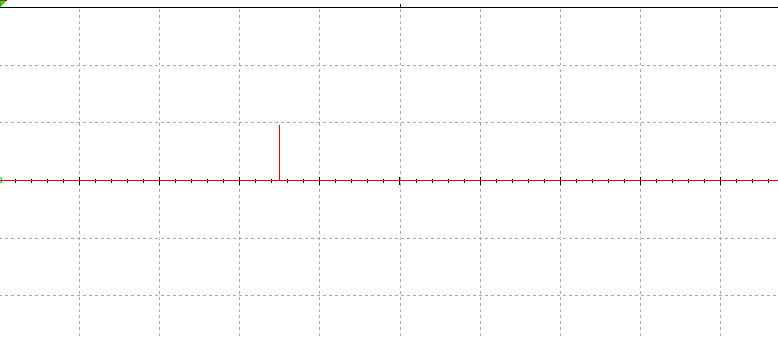
**（2）星期无法进位**

这是困扰我最久的一个bug，我选择的进位信号是计时模块24点自置零信号，

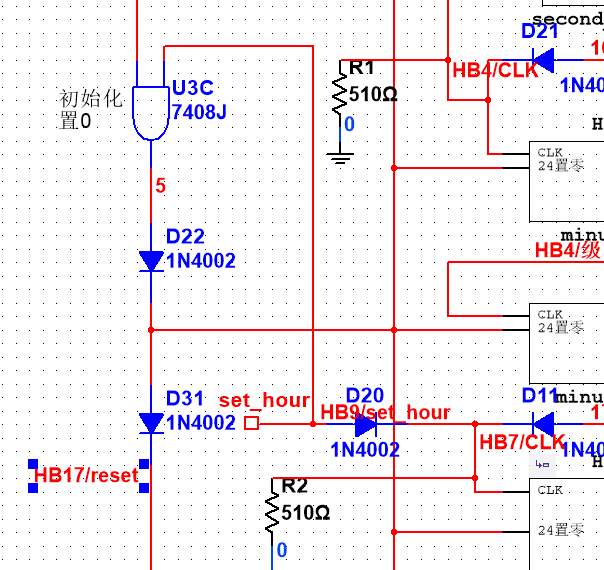
其信号如右图所示，尽管表现为一个冲激信号，经过测试，74192N的时钟端完全是可以检测到这个信号的。但是，此信号输入星期模块后，星期无法进位，被卡死在了“1”.

星期模块本身经过测试完全正常，没有办法，我只能进行逻辑上的倒推。既然卡死在了“1”，那么或许是初始化信号或者是星期模块中的逢8进位信号为1，导致我的模块被强制性设置在1.

以上是我的星期模块内部，当QD为1（即8以后的所有数据）或reset为1时触发LOAD端，输出置1。知道这一点后，我想读到这的读者也会觉得诧异：reset信号仅仅在按键M和按键H同时按下时才会输出为1，而QD仅仅在DEC8以后才会输出1，那哪来的信号让LOAD端为0，从而被触发呢？

我用示波器读取这两个信号，QD一直为0，符合逻辑。离奇的是reset信号，在我没有按下M和H的情况下，其信号在计时模块24点进位的一瞬间，出现了如下情景：

很眼熟的信号，这不就是进位信号的冲激吗？

于是顺藤摸瓜，发现了reset信号接在了此处：

reset信号和所谓的24置零信号接在了一起！我自以为用一个二极管就能实现隔离，但是忘记了24置零信号是有冲激的。换言之，我的reset信号和进位信号是分配在一起的。

*\*比较幽默的是，我到写下这一段的时候才意识到问题是这么简单，但是在这之前我一直以为这个问题是无解的。*

解决方法也相当简单，把reset信号直接接在序号5处即可，问题终于解决。因此，当读者打开文件或者放大之前的电路截图时，可以看见一个孤零零的D31，就将其当作debug的象征吧。

**七、实验总结与心得体会**

本次设计几乎将过去学过的数电知识全部用上，我也终于体会到何为“纸上得来终觉浅，绝知此事要躬行”。例如计数器的设计、计数器之间的级联、前后级信号的防串扰等等知识，在理论课学习的时候仅仅只是死记硬背，但实际用上时方觉前人的智慧真可谓无限。许多问题是不去做就不会发现的，例如将一个门的输出端与另一个门的输出端直接接在一起，逻辑上这似乎没什么，但是实际而言，这样会引起电流的倒灌，在仿真中表现为报错，在实际中则可能表现为爆炸。

本次实验，我想我收获的最珍贵的一点是：永远要相信事出有因。就像前文提及的星期无法进位，在当时我倾尽全力也无法明白故障何在，仅仅是改了个接线就万事大吉；但是几天以后，甚至几年以后，当再度思考这个问题时，或许很轻易地就找到问题所在。数电是逻辑的，世界也是逻辑的，因此永远要相信“真相只有一个”，只是暂时无法寻觅到而已。

数字钟是第一个我完全独立、没有借助任何工具而独自设计的课程作业，其中固然充满了不完善的设计缺陷，例如切换到秒表功能外部的计时模块就被干扰，如果老师没有提醒，或许我还认为这是理所当然的。

但同时，它也的确按照我的设想逻辑正常运行，给予我极大的满足感。这篇实验报告本也可以草草了事，但我选择静下心来，一个字一个字地将其中的苦恼与乐趣倾囊而出。找一份学长学姐的作业、报告是很简单的，但这过程中经历的种种却是独一无二的。或许这份数字钟本身对我的升学、工作而言没有任何帮助，倘若写进我的简历中或许还会被嘲笑一番；但是经由它我所领悟的却远超其本身。

行文至此，已经是2025年6月2日的凌晨1点38分了。感谢当初选择独自设计数字钟的自己，也感谢各位老师一学期的谆谆教诲，同时感谢助教老师辛苦批阅数量繁多的实验报告，我想我受益匪浅！