**云南大学信息学院2020年至2021年上学期**

**《数字电路与逻辑设计实验》实验报告**

**实验名称：综合实验——数字时钟的设计** **教师：** **官铮**

**学号： 20191060136 姓名： 龙萍 序号： 24**

**上课日期： 2021年6月9日 班级： 2019级通信工程**

1. **实验器材（芯片类型及数量）**

|  |  |
| --- | --- |
| 芯片型号 | 数量 |
| 4位二进制计数器74LS160 | 6 |
| 4选1数据选择器74LS153M | 2 |
| D触发器 DFF2 | 1 |
| 七段译码器74LS48 | 1 |
| 可编程分频器74LS292 | 1 |
| 2线-4线译码器74LS139 | 1 |
| 3输入与非门74LS00 | 3 |
| 非门 | 1 |
| 5引脚与门 | 1 |
| VCC | 5 |
| 两引脚与门 | 1 |
| GND | 3 |

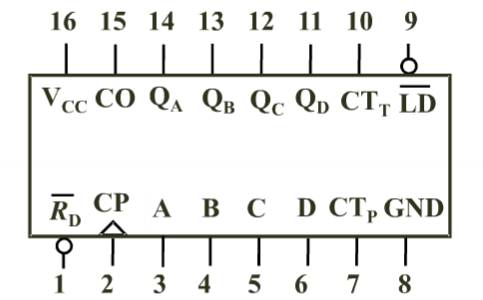
1. **实验原理**

1、七段译码器：因为计算机输出的是BCD码，若需在数码管上显示十进数，就须先把BCD码转换成 7 段字型数码管所要求的代码。我们把能够将计算机输出的BCD码换成 7 段字型代码，并使数码管显示出十进制数的电路称为“七段译码器”。其输入：8421BCD码，用A B C D表示（4位）；输出：七段显示，用a ~ g 表示（7位）。

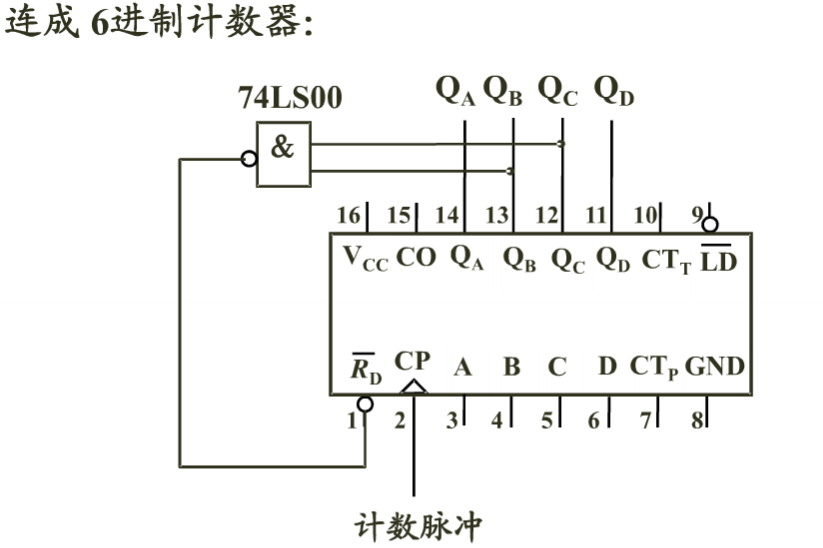
2、2-4译码器:用来产生片选信号00-11,分别控制译码后的7段数据在哪一个数码管上显示，同时也控制输入的4组数据哪一组被选出来进行译码。

3、四选一数据选择器：D0、D1、D2、D3是四个数据输入端，Y为输出端，A、B是地址输入端。利用指定AB的代码，能够从D0、D1、D2、D3这四个输入数据中选出任何一个并送到输出端。因此，用数据选择器可以实现数据的多路分时传送。此实验可以通过四选一数据选择器，选出四对输入信号送入七段显示译码器，进而显示相应字符。

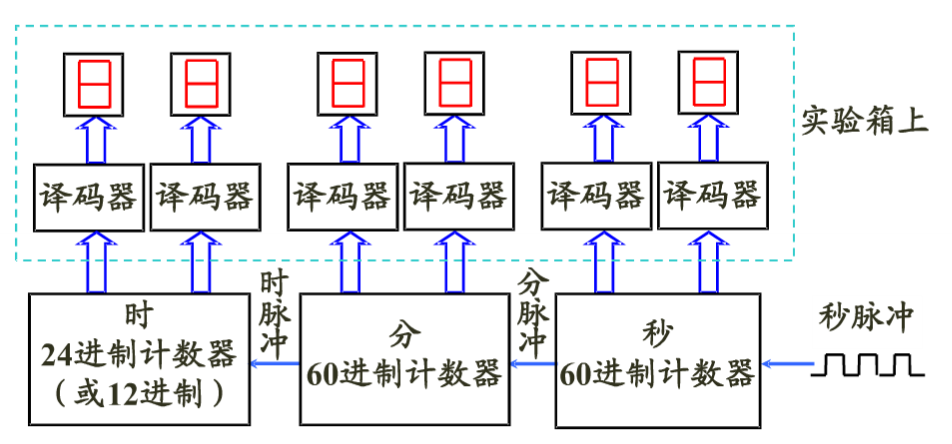
4、74LS160计数功能（脉冲引入方式，计数模式，清零方式，预置数方式）



5、应用74LS160设计6进制计数器



6、数字钟的总体结构



7、实验基本步骤

（1）从时间显示出发，时、分、秒三级时间均按照数字外形进行显示，即七段数码管显示。为使七段数码管能够显示十进制数，需要使用到显示译码器与之相配对。计数结果通过七段数码管显示，为满足显示需要，计数结果需要按照 BCD 码的形式进行输出。

（2）显示的时间数值则划分为计时部分，根据时间描述结构，可以采用 24 进制计数器对时一级时间单位进行计数，分、秒时间单位均采用 60 进制计数器。因此，我们采用集成计数器构成对应的 24 进制和 60 进制计数器。

（3）以秒脉冲为时钟源，令秒计数器对秒脉冲进行计数，记满 60 后形成分脉冲与分计数器级联。同理，分计数器按分脉冲进行计数，记满 60 后向小时计数器进行进位。小时计数器则按小时脉冲进行计数，根据选取的时间制式，满 24 后计数器清零。

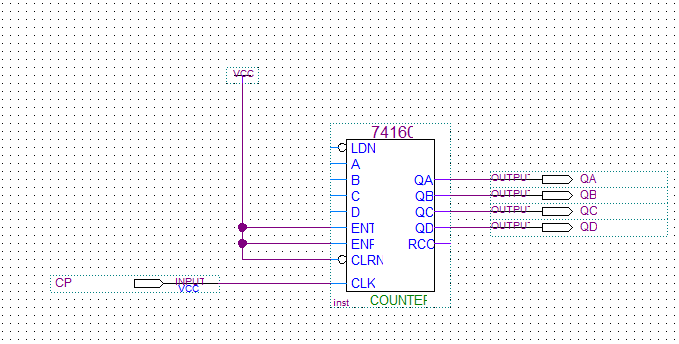
（4）用数据选择器构成多路复用模块分别读取不同进位的数据并传输到显示译码器中，实现计数器与数码管的连接。数码管的动态显示需要通过 2-4 译码器进行控制构成选通电路模块。

（5）数据选择与计时频率的不同通过分频器实现，保证数码管动态显示频率高于计时频率的四倍以上，并保证肉眼看不出闪动显示过程。

1. **实验内容及原理图**

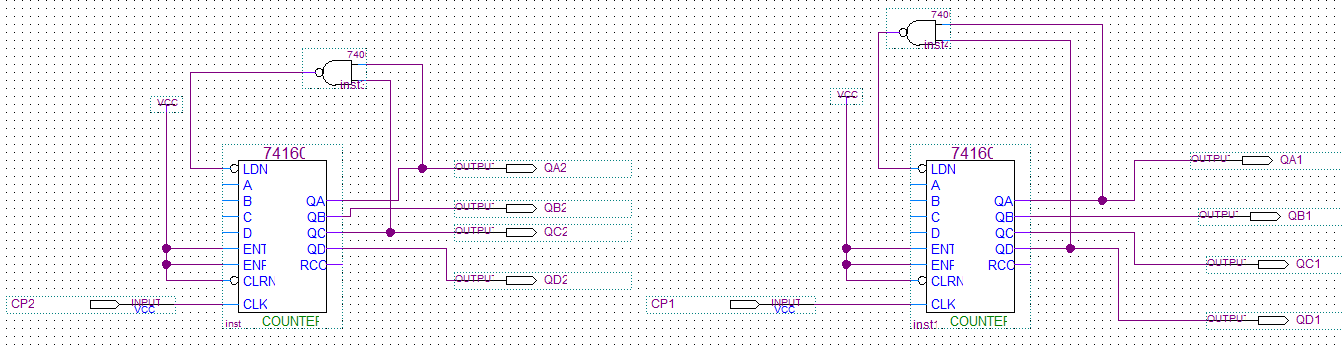
1、对74LS160计数功能进行测试

电路图：



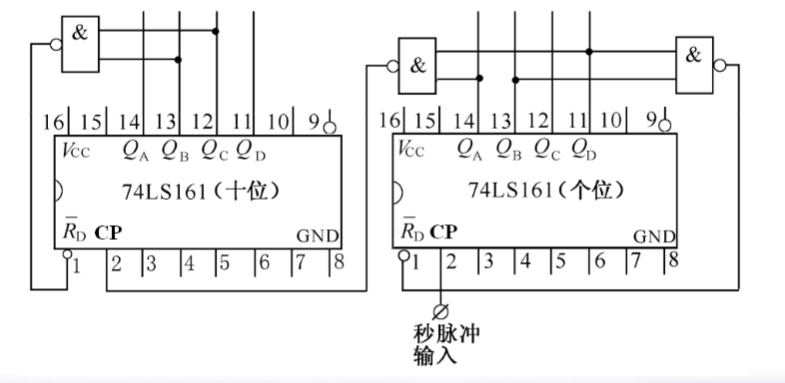
2、分别连接10进制和6进制计数器

电路图：

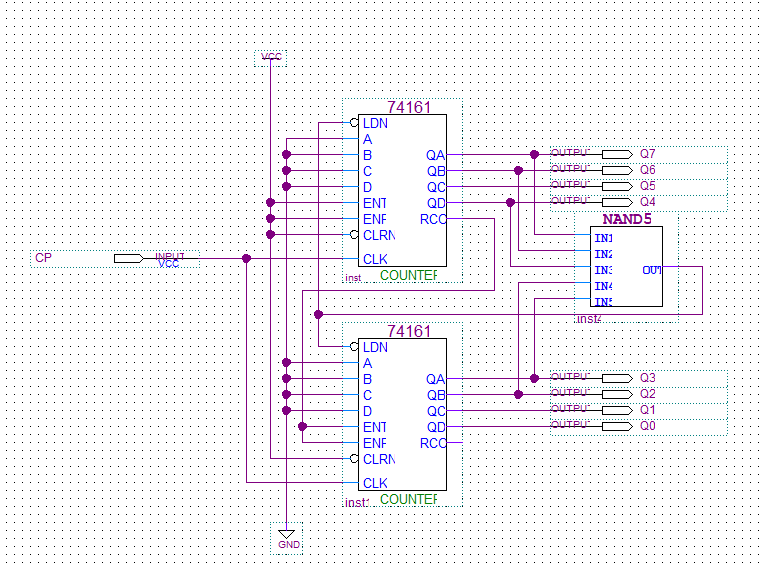


3、60进制计数器的设计

电路原理图：

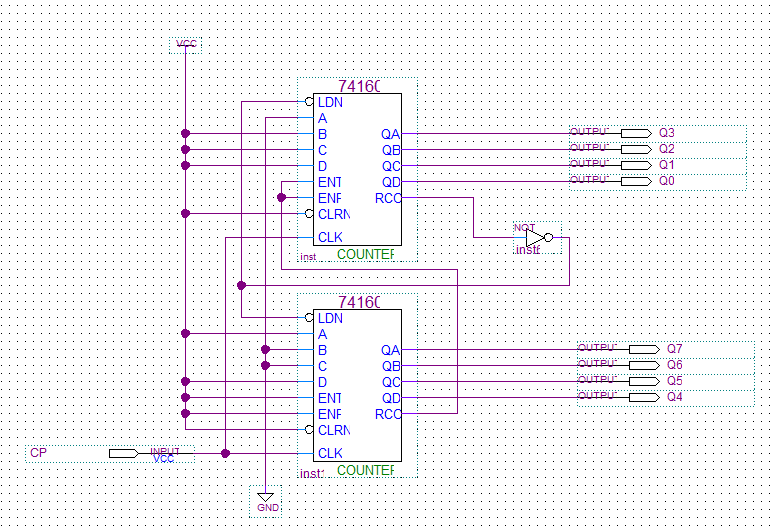


实验电路图：

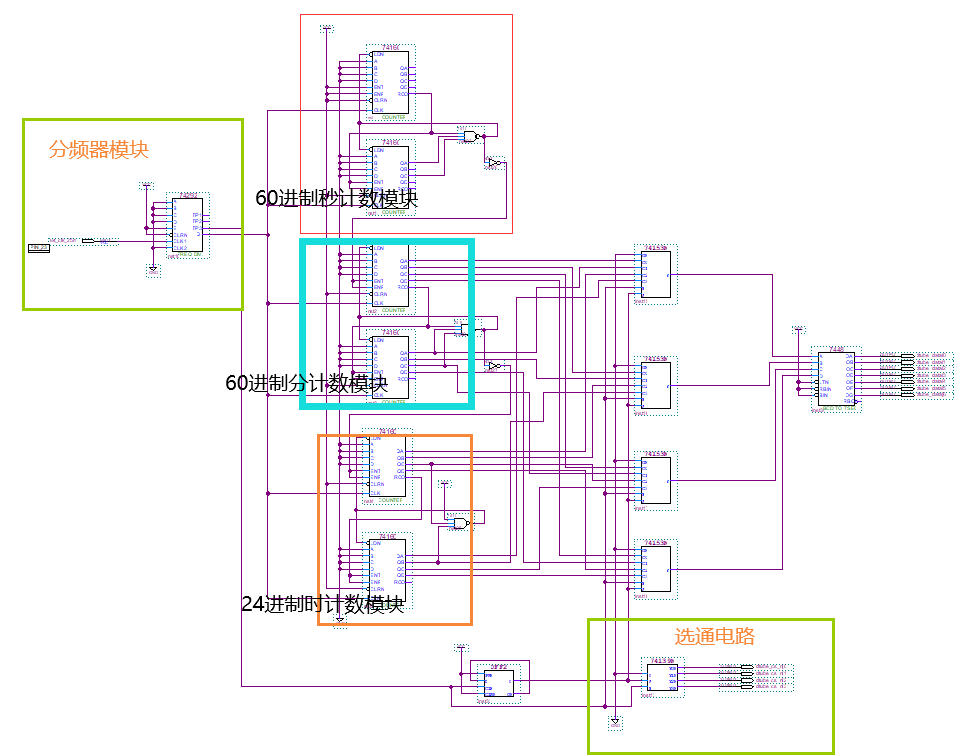


一片74LS161计数器可以实现0-15的计数功能，60进制计数器为大规模计数器，需采用多片计数器级联得到模为60的计数器。将两片74LS161计数器同步级联，采用整体置零法构成模为60的计数器，计数范围为00000000-00111011。

4、24进制计数器设计



一片74LS16计数器可以实现0-10的计数功能，24进制计数器为大规模计数器，需采用多片计数器级联得到模为24的计数器。将两片74LS160计数器同步步级联，采用OC整体置数构成模为24的计数器，计数范围为233-256，计数到255（Oc=1）时使两片 均为0，下一个CP来到时置数，预置输入=256-M=233，故D0C0B0A0DCBA=（233）10=（11101001）2。

5、实验最终总电路

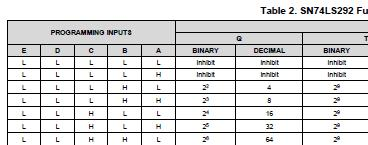
（1）进制计数器间的级联部分

如60进制分计数器与60进制秒计数器之间的级联，采用同步级联的方法进行，当低位秒计数器计数到59（01011001）时，取秒个位计数器的进位信号和秒十位计数器的第1、3 位输出的与，当这些信号全为 1 时表明计数器记到 59，后将信号传送至高位分计数器的 P、T计数器允许控制端，高位分计数器进 1。

（2）分频器部分

因实验中我们用到的FPGA开发套件提供的频率过慢，在观察FPGA开发套件上的实验结果动态显示也会变化的很慢，所以得采用分频器，加大动态显示变化的频率，使我们能较为方便清晰的观察到实验结果，如我们想看到秒计数器进位时，如果频率过慢我们可能等待的时间较长，但是利用分频器，就可以以我们想要的变化频率变化，可以较快且清晰的看到秒计数器向分计数器进行变化。

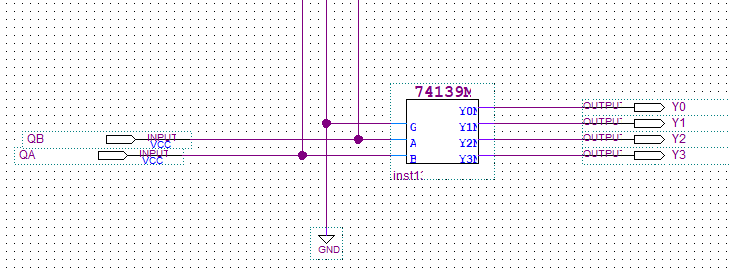
根据分频器功能表中显示，EDCBA 引脚分别接 10000时控制分频器实现 Q 输出端实现215倍分频。



（3）动态显示部分

设计电路时，4 位十进制分别由四个七段数码管进行显示，每一个七段数码管分别对应数字时钟的时十位、时个位、分十位和分个位。出于本次实验软件无法正常观察到数码管的显示成果，我们采用输出端子观察数字信号的输出结果从而进行人工对照，检验实验成果是否成功。本次实验我们将通过观察 DCBA 的输出情况来判断是哪一个数码管在进行显示。

DCBA 输出端由 2-4 译码器进行输出信号控制，使能端保持低电平有效，输入端 K0-1通过 2 位二进制进行顺序控制，使4个输出端能按顺序控制数码管显示。其中译码管的输出端为低电平有效，哪一个输出端输出信号为 0，所对应的数码管进行显示。



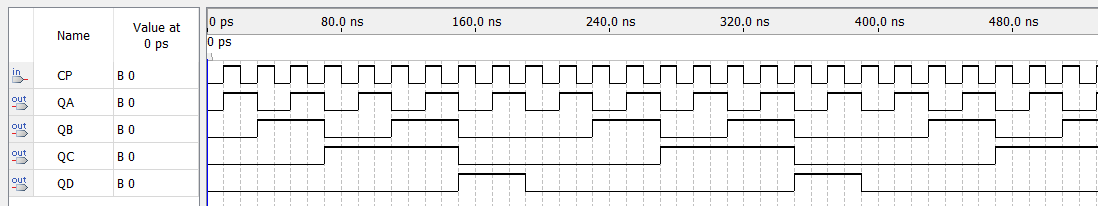
**四、实验数据记录（真值表/时序波形图/状态转换图）**

1、对74LS160计数功能进行测试

功能表：

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | | | 输出 |
| CP | P T | D C B A | QD QC QB  QA |
| × | 0 × × × | × × × × | 0 0 0 0 |
| 🡩 | 1 0 × × | d c b a | d c b a |
| 🡩 | 1 1 1 1 | × × × × | 计数 |
| × | 1 1 0 1 | × × × × | 保持 |
| × | 1 1 × 0 | × × × × | 保持（QC=0） |

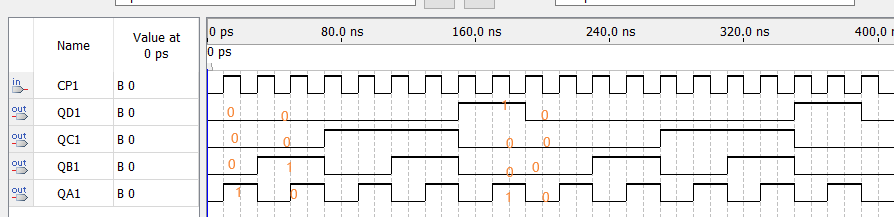
时序波形图：



2、分别连接10进制和6进制计数器

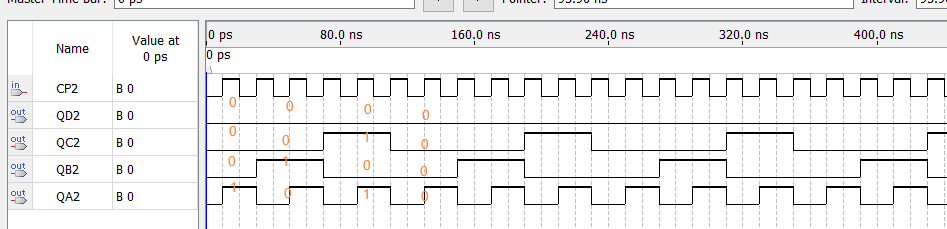
（1）10进制计数器（0000--1001）

时序波形图：



（2）6进制计数器（0000--0101）

时序波形图：



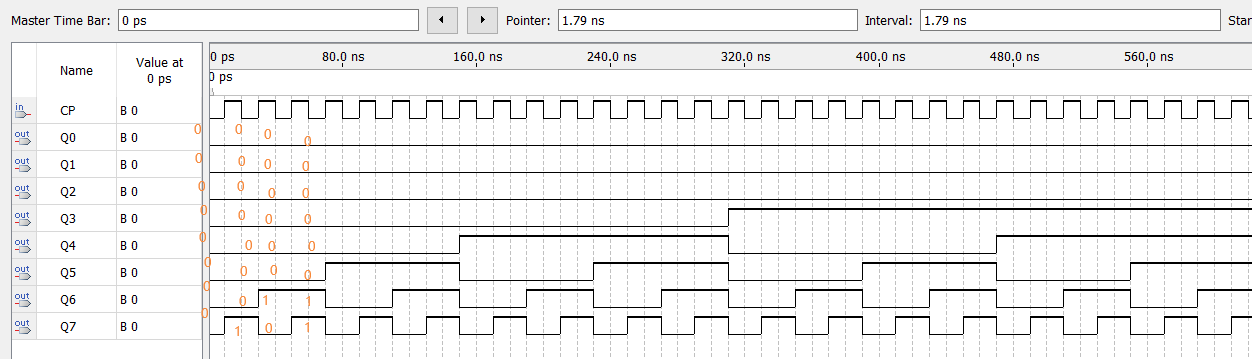
态序表：

|  |  |  |  |
| --- | --- | --- | --- |
| QD | QC | QB | QA |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| QD | QC | QB | QA |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |

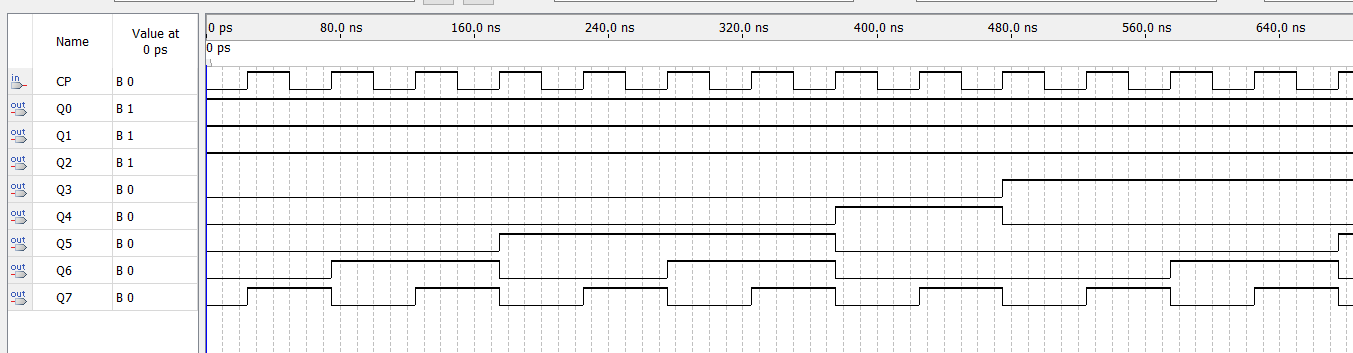
3、60进制计数器的设计

时序波形图：



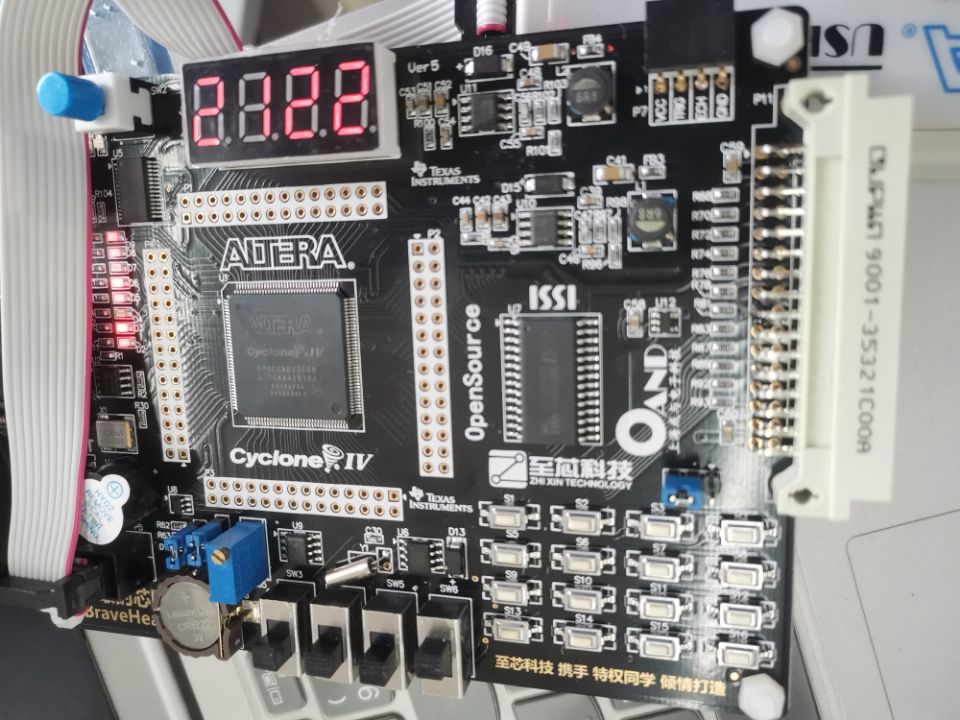
如图所示，该电路实现了从0到59的计数功能，当计数到00111011时，通过与非门将信号传送到端用预置实现清 0。

4、24进制计数器设计



如图所示，电路从11101001计数到11111111，计数到11111111时，通过非门将预置实现置数，实现从0-23的计数。

5、实验最终总电路

实验结果图：

**五、总结**

1、通过本次实验，我熟悉掌握了集成计数器的运用，设计电路、进行仿真、运行结果都得自己实际操作完成。在数字时钟设计中，根据老师上课所讲的内容，可以用两片集成十进制同步计数器74LS160级联为100进制，再利用其异步清零功能，可以分别实现小时的24进制和分秒的60进制。当然，在仿真过程中也遇到了很多困难和问题。比如说,无法直接从秒进位到分和分进位到时，并且在仿真中总是出错。于是自己请教了一些同学，同时在网.上查找了相关资料，最后终解决了这个问题。

2、学习了分频器的原理分频器存在 4 个输出端口 Q、TP1、TP2、TP3，例：分频倍数为22倍，输出信号为原信号的 倍。不同输出端口在相同控制端 EDCBA 情况下也有可能存在不同的倍数。

3、通过本次实验对数字电路知识有了更深入的了解，将其运用到了实际中来，明白了学习电子技术基础的意义，也达到了其培养的目的。也明白了一个道理:成功就是在不断摸索中前进实现的，遇到问题我们不能灰心、烦躁，甚至放弃，而要静下心来仔细思考，分部检查，找出最终的原因进行改正，这样才会有进步，才会一步步向自己的目标靠近，才会取得自己所要追求的成功。当然，自己的仿真技术和应用能力还是很欠缺的，虽然完成了基本的设计要求，但是很多自己想要的扩展功能还未能实现。而且很多时候会走过很多弯路,浪费了很多不必要的时间。不过这次设计经历必将使我受益终身，让我明白如何更好的获取知识，如何更好的理论联系实际。今后的学习更需要不断努力，在获得知识的同时获得快乐，真正的主动探索，主动学习，形成自己的思维方式，不断应用，不断进取。

**六、思考题**

1、异步计数器为什么容易产生误动作？同步计数器能够避免误动作？为什么？

[同步计数器](http://www.so.com/s?q=%E5%90%8C%E6%AD%A5%E8%AE%A1%E6%95%B0%E5%99%A8&ie=utf-8&src=internal_wenda_recommend_textn):实现是将计数[脉冲](http://www.so.com/s?q=%E8%84%89%E5%86%B2&ie=utf-8&src=internal_wenda_recommend_textn)引至所有的[触发器](http://www.so.com/s?q=%E8%A7%A6%E5%8F%91%E5%99%A8&ie=utf-8&src=internal_wenda_recommend_textn)的CP端，使应翻转的触发器能够同时的翻转。

异步计数器:实现是不将计数脉冲引至所有的触发器的CP端，而是将其它的触发器的输出引至其他的触发器的CP端，是不同时发生的。

对于同步计数器，由于[时钟脉冲](http://www.so.com/s?q=%E6%97%B6%E9%92%9F%E8%84%89%E5%86%B2&ie=utf-8&src=internal_wenda_recommend_textn)同时作用于各个触发器，克服了[异步触发器](http://www.so.com/s?q=%E5%BC%82%E6%AD%A5%E8%A7%A6%E5%8F%91%E5%99%A8&ie=utf-8&src=internal_wenda_recommend_textn)所遇到的触发器逐级延迟问题，于是大大提高了[计数器](http://www.so.com/s?q=%E8%AE%A1%E6%95%B0%E5%99%A8&ie=utf-8&src=internal_wenda_recommend_textn)[工作频率](http://www.so.com/s?q=%E5%B7%A5%E4%BD%9C%E9%A2%91%E7%8E%87&ie=utf-8&src=internal_wenda_recommend_textn)，各级触发器输出相差小，[译码](http://www.so.com/s?q=%E8%AF%91%E7%A0%81&ie=utf-8&src=internal_wenda_recommend_textn)时能避免出现[尖峰](http://www.so.com/s?q=%E5%B0%96%E5%B3%B0&ie=utf-8&src=internal_wenda_recommend_textn);但是如果同步计数器[级数](http://www.so.com/s?q=%E7%BA%A7%E6%95%B0&ie=utf-8&src=internal_wenda_recommend_textn)增加，就会使得计数脉冲的[负载](http://www.so.com/s?q=%E8%B4%9F%E8%BD%BD&ie=utf-8&src=internal_wenda_recommend_textn)加重。

与同步计数器相比较，由于触发器不是共用同一个[时钟源](http://www.so.com/s?q=%E6%97%B6%E9%92%9F%E6%BA%90&ie=utf-8&src=internal_wenda_recommend_textn),触发器的翻转不能同时发生，所以[工作速度](http://www.so.com/s?q=%E5%B7%A5%E4%BD%9C%E9%80%9F%E5%BA%A6&ie=utf-8&src=internal_wenda_recommend_textn)慢。