**同济大学电子与信息工程学院实验中心实验报告**



**实验课程名称： 电子可编程器件**

**任课教师： 徐和根**

**实验项目名称： 时钟芯片设计**

**实验指导教师： 蔚瑞华**

**姓名:\_\_\_\_石卓凡\_\_\_ 学号:\_\_\_\_\_1751051\_\_\_**

**同组同学：\_无\_\_**

**实验日期:\_ 2019.12.13及12.20\_ 实验地点 电信楼348室**

**实验名称：\_\_时钟芯片设计\_\_\_\_\_\_**

**姓名:\_石卓凡\_ 学号:\_\_\_\_1751051\_\_\_\_ 同组：\_无\_\_ 实验日期:\_2019.12.13及12.20\_**

**一、实验目的**

1. 掌握用Verilog语言编写程序，在Nexys4开发板实现时钟芯片设计；

2. 掌握用模块化设计思路编写程序。

**二、实验内容**

1. 根据设计流层图画出各模块框图；

2. 掌握分层次、分模块进行各模块设计和仿真的方法；

3. 使用Verilog语言描述分频电路、十二进制计数器、六十进制、计数器等模块；

4. 选用硬件验证实验装置上芯片型号，对所设计的模块和总图在开发软件上进行设计实现和仿真；

5. 最终，将时钟芯片设计程序下载到硬件验证实验装置上，通过外围接口设备的运行观察设计结果。

**三、实验方案及流程图**

（一）设计功能说明

本实验设计的时钟芯片的功能如下：

1. 具有时、分、秒三级的计时，能实现正常24小时的计时（00:00:00——23:59:59）；

2. 能实现时钟计时的重置（重置回00:00:00），即清零；

3. 能实现时钟信号的时间设置，能分别设置时、分、秒，用户输入合适的数值来指定要设置的时间；

4. 时钟的时间通过六个共阴极数码管来显示；

5. 能实现计时的暂停，取消暂停后时钟从暂停前瞬间的计时数继续开始计时。

（二）工作原理说明

本实验基于Verilog硬件描述语言、Nexys4开发板以及Vivado软件，不选用ISE软件开发的原因是个人认为Vivado软件开发更加方便、操作更加简便，本人也更加熟悉Vivado软件的开发方式（尽管ISE和Vivado之间有很多相似和继承关系）。

工作原理的说明主要体现在实验代码的几个模块上：

1. 控制模块：作为计时控制模块，可以设置时、分、秒，并可以让时钟计时暂停但不重置的功能；

2. 计数模块：本实验通过1s时钟计数，时钟范围为小时0-24、分0-60、秒0-60。需设计24进制模块1个、60进制模块2个；

3. 分频器：通过分频器将Nexys4开发板的系统时钟频率（100MHz）分频，得到扫描频率200HZ-1KHZ（供给扫描电路作为扫描频率用于六个七段数码管的显示）和时钟计数频率1HZ；

4. 七段译码器模块：将输入的四位二进制BCD码转化为相应的七段数码管的输入信号，使得数码管上显示出与BCD码相对应的十进制数；

5. 数据选择器（扫描电路）：为了节省I/O管脚和内部资源，常用扫描方式进行显示。

6. 本实验有六位扫描数码显示，将时间划分为6个扫描周期，在周期1显示第1个数字，周期2显示第2个数字依次类推，完成6个扫描周期后，重新开始6个扫描周期，当扫描速度超过一定数值，人的眼睛感觉6位数码显示是同时的。数据选择器的功能是对个位秒、十位秒、个位分、十位分、个位时、十位时6个BCD数据进行：扫描。数据选择器由6进制计数器、3/6译码器、24选4多路开关组成。

**（NOTE）**

1. 为了确保功能上的简易，暂停功能在我设计的时钟芯片里实际上是设置时间的一种特殊情况；

2. 同样为了功能上的不重复，我取消了时钟控制模块的“清零”功能，时钟的重置清零在我设计的电路中必须由且仅由全局reset信号负责；

3. 考虑到分（min）每计1，秒（sec）就要计60个，因此我还是将2个60进制模块分开来进行设计；

4. 为了方便代码的开发，在我的设计中，6进制计数器、3/6译码器和24选4多路开关都是集中在一个模块里实现的，并没有单独作为一个模块，否则过多过分简单的模块会导致端口数量的增加，从而增加main文件中的工作量；

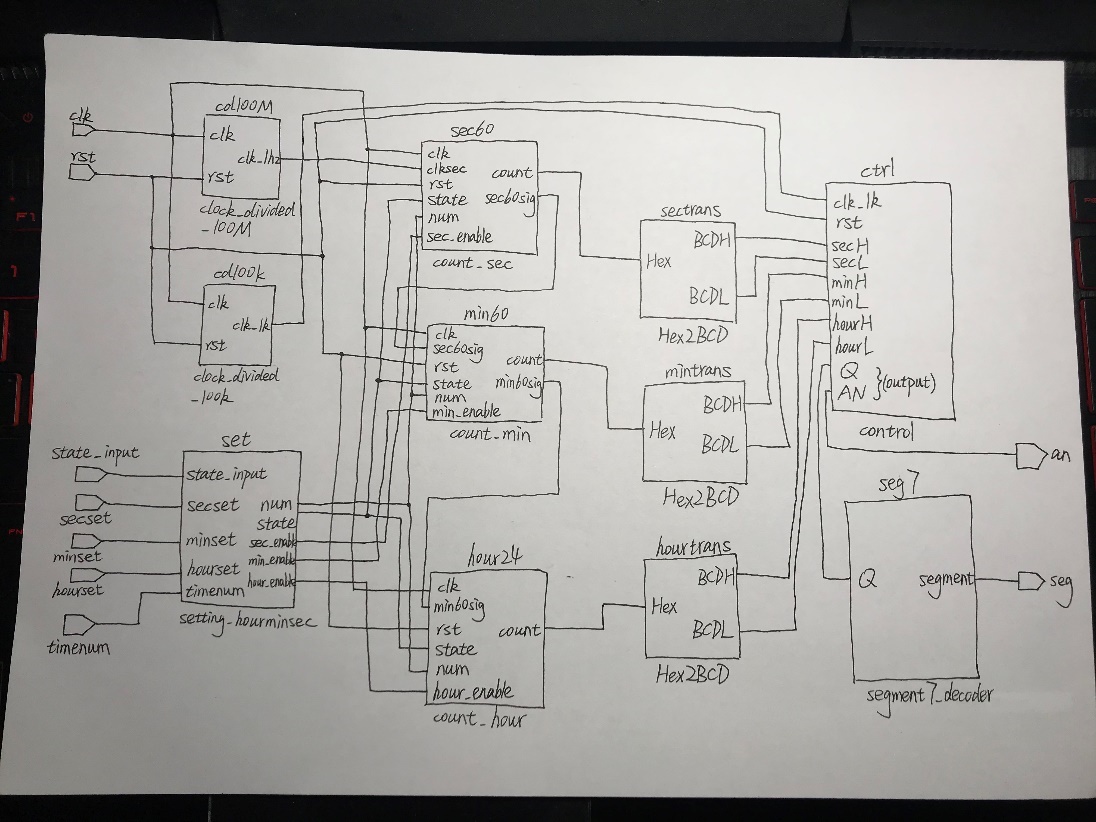
5. 3/6译码器：将六进制计数器的计数（0~5）译成六位独热码的反码——因为Nexys4的数码管是共阴极数码管；

6. 24选4多路开关：经过Hex2BCD后，24、60进制计数器转化为个位秒、十位秒、个位分、十位分、个位时、十位时6个BCD数据，其大小分别是从0到9，因此是四位二进制BCD码，总共24位，每次选一个BCD数据（即4位），再把该BCD数据转化为七段译码管的七个段的亮灭。

上述注意事项会在之后进行详细说明。

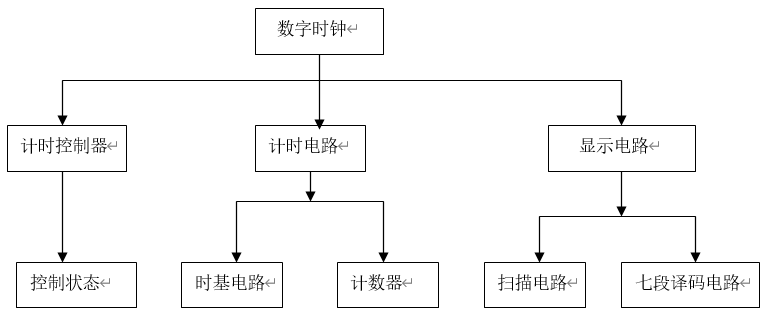
（三）设计原理图和框图

由于我的设计是基于Vivado软件，且用的是全硬件（PL）以及main文件实现整个时钟芯片的功能，因此我并没有引入ISE的电路设计工作，也没有封装成IP核，并绘制BLOCK DESIGN。



上图所示为整个时钟芯片的设计电路，即功能原理图，总共有7个输入和2个输出，包含11个模块（9种模块）。

思路框图如下所示：



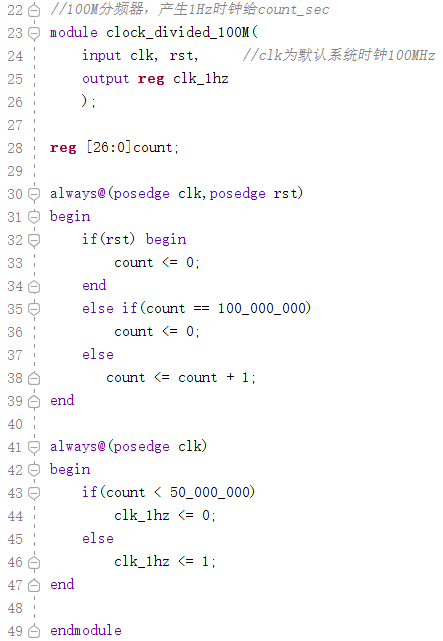
（四）设计输入

1. 原理图如（三）所示。

2. Verilog语言描述各模块

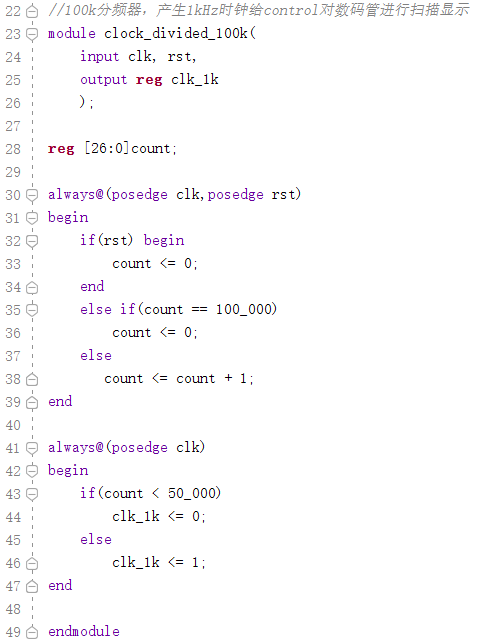
参照功能原理图：

（1）clock\_divided\_100M模块



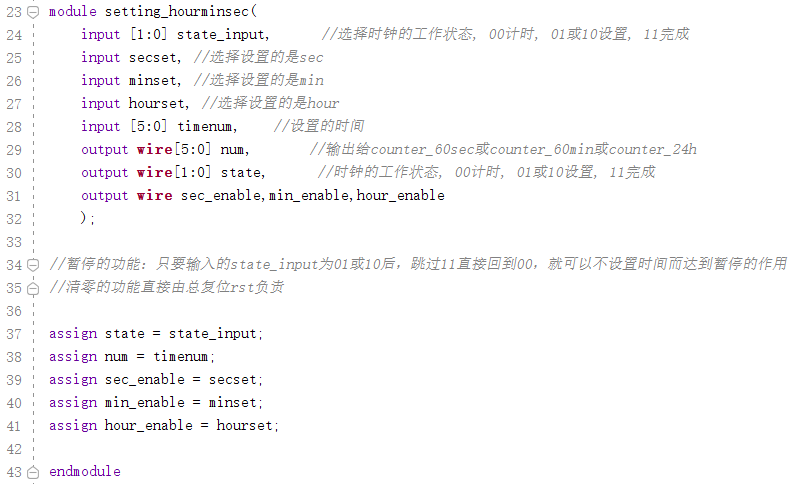
输入信号是系统时钟100MHz和全局reset信号rst，经过100M的分频之后输出频率为1Hz的时钟信号，它将作为秒计时器的计时输入时钟，指示后者一秒计1.

（2）clock\_divided\_100k模块



输入信号是系统时钟100MHz和全局reset信号rst，经过100k的分频之后输出频率为1kHz的时钟信号，它将作为control模块的输入时钟，实现对六个七段数码管的扫描。

（3）setting\_hourminsec模块



此模块的五个输入信号均为用户输入，其中state\_input为模式选择，当为00时时钟计时，01或10时为设置时间模式，11时为FINISH状态，此时将01或10时赋的值输出到七段数码管上；secset、minset、hourset分别为设置选择，当secset为1时设置秒，0时不设置，minset、hourset同理，允许同时设置时、分、秒；timenum为六位二进制数，它是用户设置的时间的二进制形式，用户输入二进制形式，在数码管上显示对应的十进制数，时、分、秒都用timenum进行设置。

（4）count\_sec模块

此模块是秒second计数模块，输出的秒为count，当输入的状态state为00时，每来一个clksec的上升沿（上升沿用clksec\_now和clksec\_before进行判断）时count就加1，计慢59时sec60sig输出高电平，下一次clksec到来后重置为0，同时sec60sig重置为低电平，sec60sig就是分minute计数模块的计数时钟；当输入的状态state为01或10且设置时间的使能信号sec\_enable为高电平时，时钟暂停，并且将用户输入的设置时间num存入寄存器count\_reg；当输入的状态state为11且设置时间的使能信号sec\_enable为高电平时，寄存器count\_reg的值传给输出信号count，将用户设置的时间num显示到数码管上，完成整个时钟设置的过程。

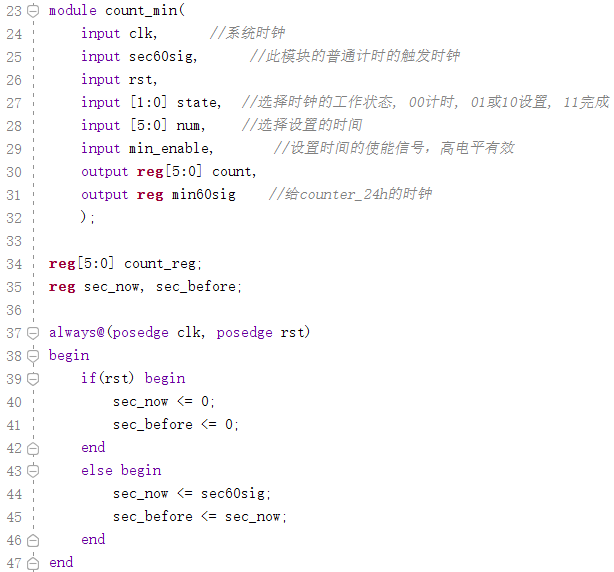
值得一提的是，如果用户在sec\_enable为1时将state设置为01或10，那么由代码可知，此时count不再受到计数时钟clksec的控制，count会保持为原来的值一直不变，达到了暂停的效果。之后，如果用户直接跳过state = 11而将state重新置回00，并将sec\_enable置回0，那么count会从原来保持的值开始继续计数，从而完成了暂停时钟的整个过程。

由于sec\_enable和state都赋给了秒、分、时计时器，因此秒、分、时的工作状态是同步的，秒在计时时，分、时也在计时，时钟处于计时状态；如果秒暂停，则分、时也会暂停，时钟便会处于暂停计时的状态。





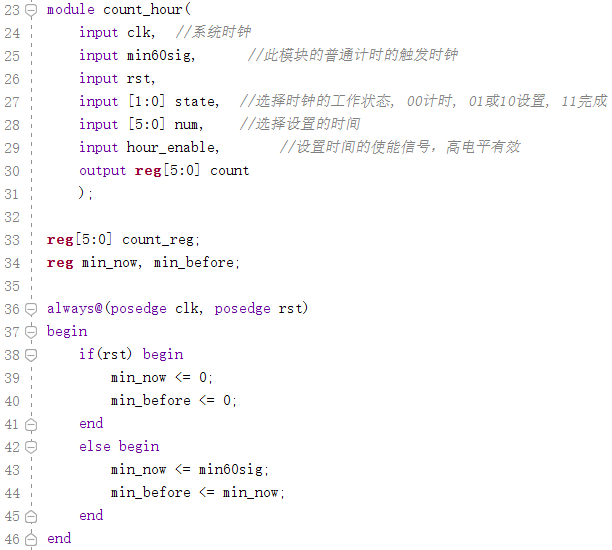
（5）count\_min模块

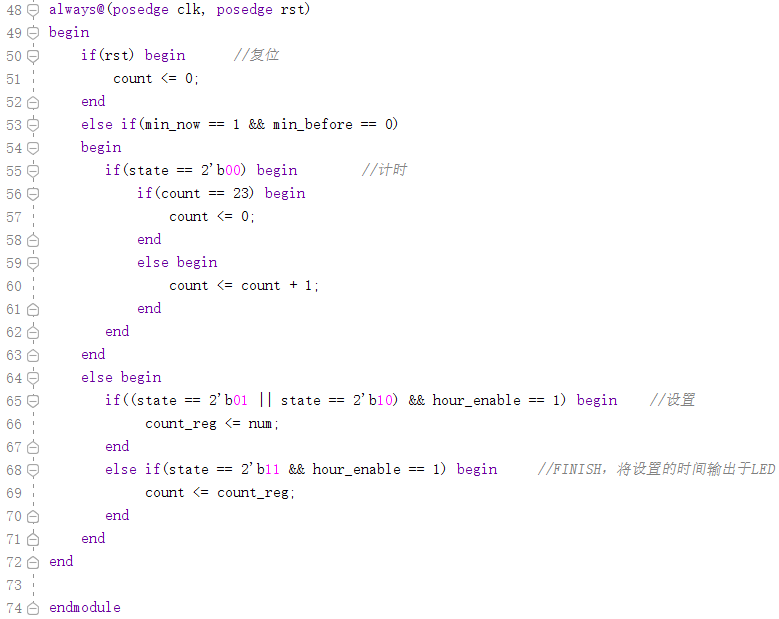




同理如count\_sec模块，唯一的不同是触发count计数加1的计数时钟上升沿变为sec60sig的上升沿了，并且本模块也会输出一个min60sig，作为时hour计数模块的计数时钟。

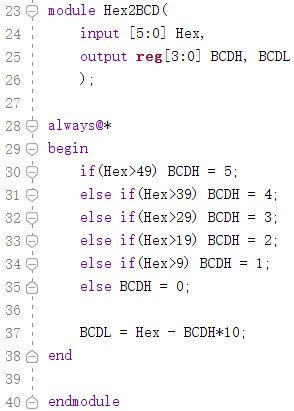
（6）count\_hour模块





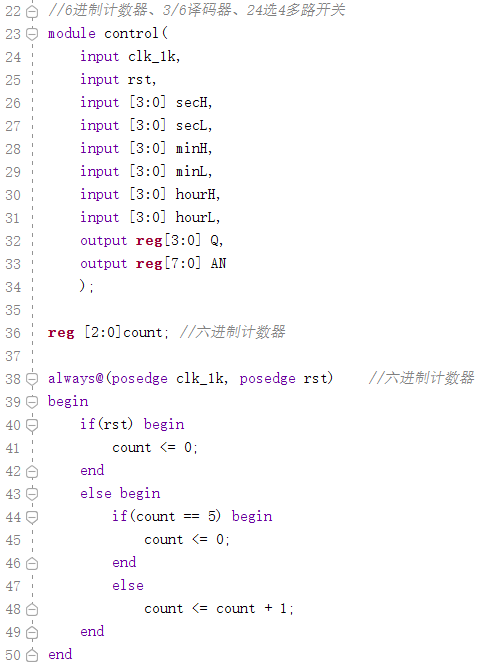
由于没有更上级的计数单位，因此时hour计数模块不需要输出hour24sig这种信号。

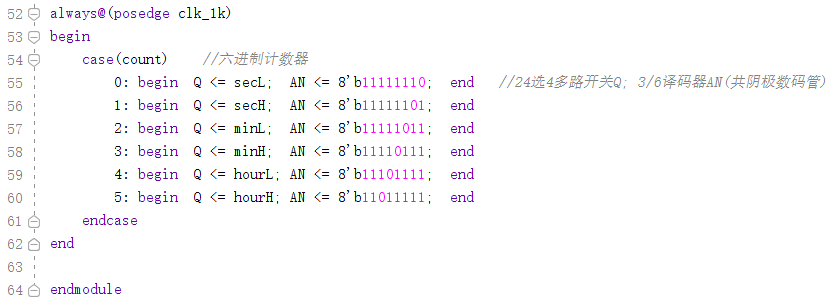
（7）Hex2BCD模块



本模块需要在电路中布置三个，分别将秒、分、时计数模块的输出count进行处理。模块接收count（即Hex），将其对应的十进制数的十位和个位对应的四位BCD码输出，BCDH和BCDL分别为十位和个位对应的四位BCD码。

（8）control模块





本模块的输入时钟是经过100k分频之后得到的1kHz时钟信号，每个时钟信号到来时，count计1。变量count作为一个六进制计数器，当count为0时，它选择将秒的十进制形式的个位数对应的BCD码输出给Q，并且AN将秒对应的数码管的引脚置0（共阴极数码管，置0说明数码管能亮，否则不能）；当count为1时，以此类推。因此，count的六个值分别可以将秒、分、时的个位、十位总共六个值分别依此显示在对应位置的数码管上，并且数码管显示的是Q的值。

虽然选中某一个位时，其他五个数码管不亮，但是由于时钟频率（扫描频率）为1kHz，考虑到人眼的暂留视觉效应，人眼在同一刻会看到六个数码管同时亮起，这样子就实现了数码管的显示工作。

AN的八个位分别控制八个数码管，某一位为低电平时对应的数码管点亮，否则不亮。

（9）segment7\_decoder模块

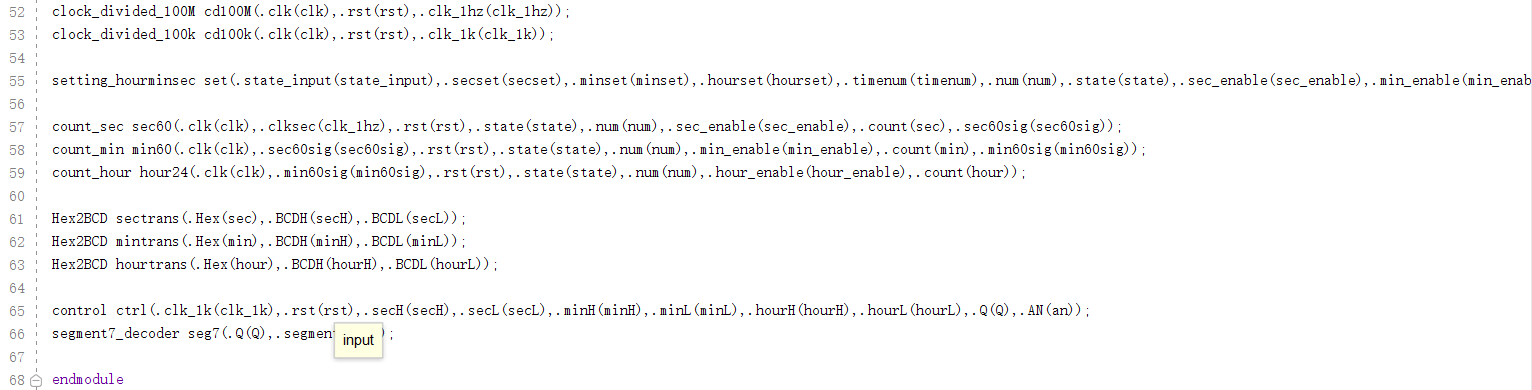


本模块是将某一时刻被control模块选中的Q值正确地显示到数码管上，segment的低七位分别代表数码管的七个段，最高位代表数码管右下角的小数点。由于是共阴极数码管，所以某一段是低电平时，该段被点亮。通过case语句，就可以设置Q从0到9时的数码管七个段被点亮的情况。

（10）main文件

此main文件为我的工程中的顶层文件，它引用上述的九种模块（其中Hex2BCD被引用了三次），相当于实现了我先前手绘的功能原理图。

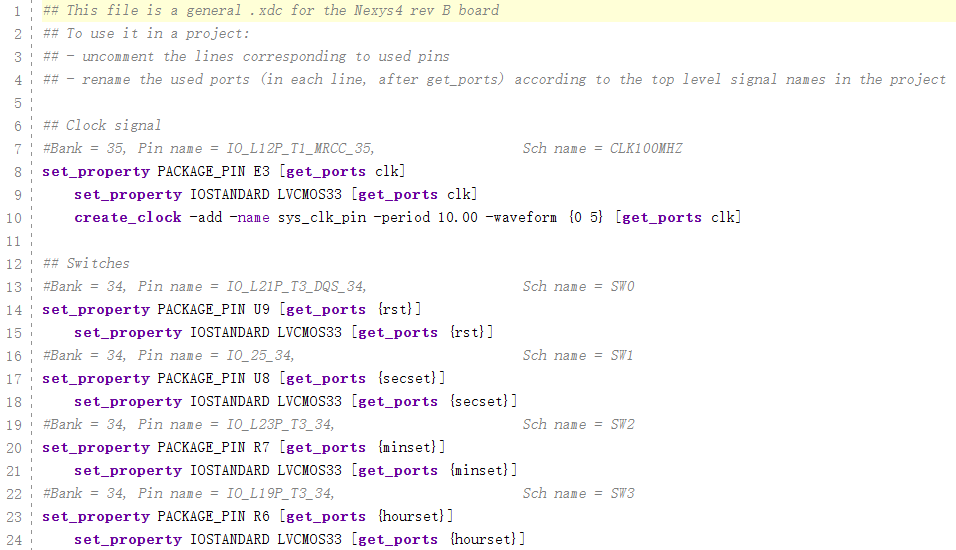




上述的十个Verilog文件详见实验报告的附件。

3. 约束文件

所有约束如下：







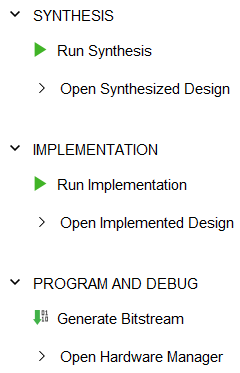


所有输入都是由Nexys4开发板的数码管下的开关设置。

**四、实验结果及体会**

（一）实验结果

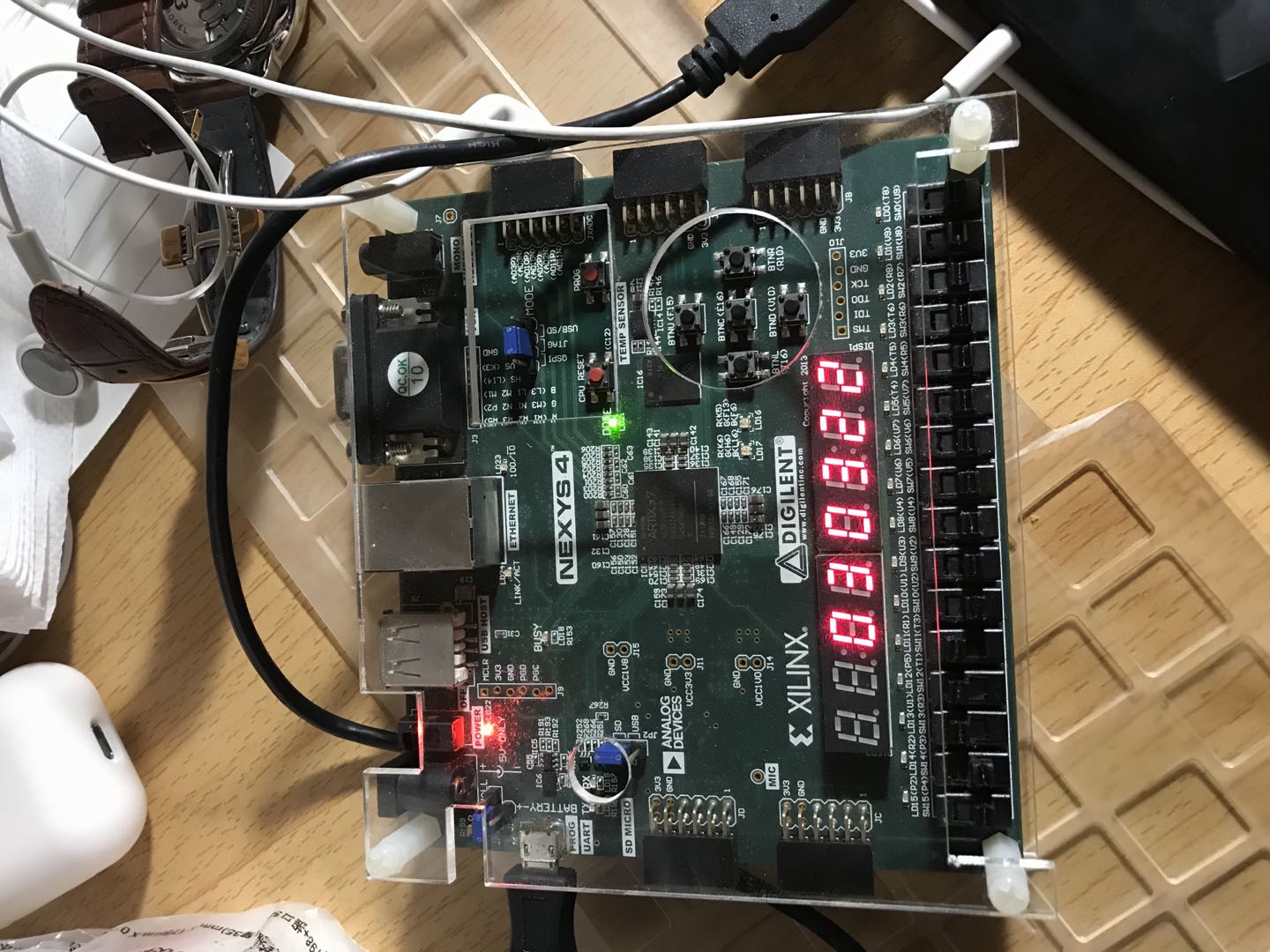
将工程保存，在Vivado中依此执行Run Synthesis、Run Implementation、Generate Bitstream三步操作，成功生成比特流。将Nexys4开发板通过USB线连接到电脑上，点击Open Hardware Manager，选择Auto Connect，最后将比特流导入开发板中。



由实验现象知，开发板一切正常，能实现所有设计初要求的功能，通过选择state\_input和secset、minset、hourset，我们可以控制时钟正常计时、暂停、设置某一位或者多位的时间，设置时间是通过六个开关代表的六位二进制数控制的。

当时钟计数时，时钟能正常进位、重置清零，当时钟记到23:59:59时，下一秒便可以自己跳转回00:00:00，开始新一轮的24h计数。

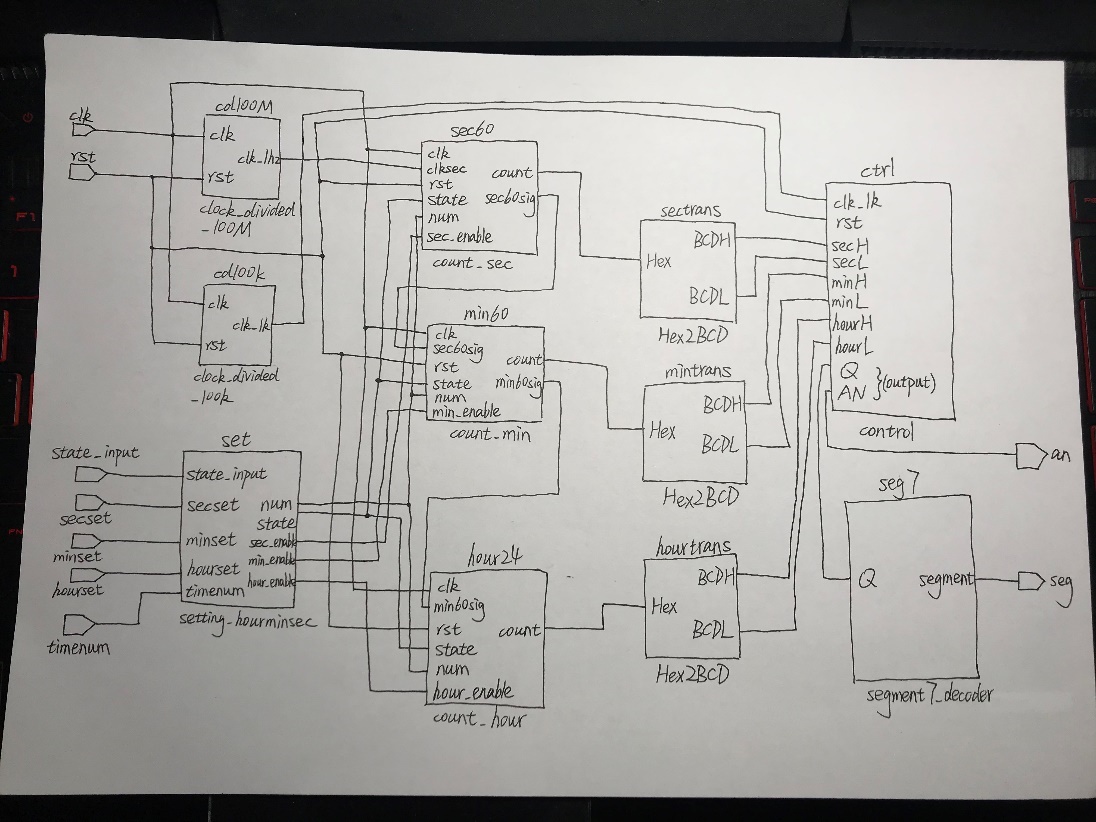
遗憾的是，由于进行实验的是我一个人，没有请其他人帮我拍摄演示视频，除了在16周课堂上的演示之外，我在提交文件中附带了一个5s的演示视频和一张照片，从演示视频上可以证明，我设计的时钟可以正常地实现秒到分的进位。



（二）设计总结

本实验基于Vivado以及硬件描述语言Verilog，通过Nexys4开发板验证了实验结果的正确。

设计中主要采用了模块化的思想，将电路功能的实现分为了11个模块，方便今后的功能拓展以及代码维护。本实验未采用Vivado的block design，未封装IP核，而是通过顶层文件main实现了功能原理图。



由功能原理图可以看出，我将实验大致分为了四步走，从左到右分别是四列部分：第一部分是设置部分，包含了分频器模块和功能设置模块；第二部分是设计部分，是设计的核心部分，它实现了暂停、设置计数、计数的功能，并为清零的rst信号提供了接口；第三部分是数制转换部分，它将实时的计时转化为六个数码管的数字对应的四位BCD码，为第四部分作准备；第四部分是显示部分，control模块选择显示的数字和数码管，segment7\_decoder模块将Q显示为数码管上的数字。

（三）实验体会

通过本次实验，我更加熟悉了Vivado软件的操作方式，并且将模块化的编程思想运用到了Verilog编程当中。实验中主要遇到了两个困难：

1. 功能的设计和实现方面。一开始我是想用Nexys4开发板上的button型按钮进行设置，每按一次button计数加一，从而实现对时、分的控制，但是在徐和根老师的提醒下，我想到button可能会有抖动型干扰，因此最后还是决定采用数码管下方的开关进行设置和功能选择，并要求用户手动输入设置的时间，而不是按一次按钮计1；

2. 设置时间和计数方面的冲突。由于Verilog是基于电路设计的语言，不允许在不同的always语句块对同一个变量进行赋值，因此对于时间的设置带来较大的困难，因为必须将这两个功能写在一个always语句块内，因此研究之后，我决定用上升沿触发。但是如果用clk\_sec、sec60sig、min60sig进行触发，则需要等到下一个clk\_sec、sec60sig、min60sig的上升沿到来时才能将设置的值赋进去；最后，通过用clksec\_now和clksec\_before判断上升沿才解决了这个问题。

实验中遇到的各种问题让我对Verilog语言的编程又加深了概念，让我意识到Verilog语言中各种与其他编程语言不同要求的限制，更加反映了它是一种基于电路设计的硬件描述语言，但同时具有一些软件编程语言的特点。

**五、实验原理的认识**

本次实验主要就是考查学生模块化编程的思想和逻辑思维能力，最让我印象深刻的还是显示电路部分的原理，通过24选4选择开关以及很高的扫描频率，运用了人眼的暂留效应，使得人眼在同一时间可以观察到六个数码管的信息，达到正常时钟计时的效果。