**北京科技大学数字逻辑实验报告**

学院：计通 专业： 计算机 班级： 计1701

姓名： 吴思懿 学号： 41724010 实验日期: 2018 年 12 月 17 日

**1、实验名称：**实验三 时序逻辑实验

**2、实验目的：**

本实验的目的是学习时序逻辑模块在数字系统中的综合应用与程序编写并且掌握实验平台的外部功能模块在数字系统设计中的应用。学习SRAM的使用，为设计复杂的数字电路，尤其是CPU设计奠定基础。

**3、实验内容：**

1. **实验3.1——在七段数码管上滚动显示学号**：按照“4 实验步骤——在七段数码管上滚动显示学号” 完成本实验。以学号40123456为例：
2. 首先将学号中的数字被存储在一个32位的寄存器msgArray中；
3. 4个数码管始终显示寄存器的高16位数据；
4. 用频率为3Hz的时钟控制7段数码循环显示：在时钟的上升沿进行向左循环移动4位，并显示。注意：记得要把msgArray中的内容，即msgArray [31:28]的内容移到msgArray [3:0]中。
5. 复位时，寄存器恢复原始存储状态，7段数码管显示第一组4位字符（寄存器中的高16位），即4012；
6. **实验3.2——通过按键输入学号，并循环显示**：

电路功能描述：通过Ego1上的按键输入自己的学号（8位10进制数），并存储在32位的寄存器中；8位10进制数输入完成后，实现类似实验3.1中的滚动显示效果。

除了要求实现上述功能外，还包括如下具体实现上的要求：

1. 整个电路具有复位功能；
2. 8个数码管中，4个数码管用于输入数据后的循环显示，另外4个用于显示当前正在输入的数据。
3. 输入过程可控、实时可视：
   * 用按键或者开关控制输入开始和结束，输入开始后用于循环显示的4个数码管停止循环显示，用于显示输入数据的4个数码管开始显示输入数据；输入结束后用于循环显示的4个数码管开始循环显示输入的学号，用于显示输入数据的4个数码管停止显示输入数据。
   * 用按键切换当前要输入的数据位，切换结果通过用于显示输入数据的4个数码管展示出来，即用户能够通过显示输入数据的4个数码管看出来当前正在输入的是哪一位10进制数。
4. 输入数字的时候进行按键消抖（关于按键防抖的原理，请参考本实验手册的“5.1 实验3.2 相关说明”部分）；
5. **设计具有开放性**，下图仅仅为参考的一种实现效果，不作强求实现一样的效果。
6. 数码管、按键、开关的管脚和控制方式等信息请参考本文件“5.1 实验3.2 相关说明”以及”EGo1使用手册.pdf和EGO1电路原理图.pdf”
7. **实验3.3——SRAM芯片数据存取**

在实验3.2的基础上实现将输入的数据存入FPGA的SRAM芯片上，当进行读取时将数据从SRAM芯片中取出来。SRAM读写请参考本文件“5.2 实验3.3的相关说明”、EGo1使用手册.pdf、SRAM数据手册.pdf以及SRAM读写时序与代码解释.pptx。

**4、实验3.3系统设计**

（1）系统输入输出信号定义

input clk100mhz,时钟信号输入

input clr,清零信号输入

input rst,复位信号输入

inout [15:0]dataBus,16位读写数据总线，输入时为高阻

input key0, 按键S0，控制输入时左端数码管亮灭，按一次，右移一次

input key1, 按键S1，控制输入时数码管上数据值，按一次，数据减一

input key3, 按键S3，控制输入时左端数码管亮灭，按一次，左移一次（十进制）

input key2, 按键S2，控制输入时数码管上数据最终值，确认后按下，该位数据将被存储，并写入sram

input key4, 按键S4，控制输入时数码管上数据值，按一次，数据加一（十进制）

output [3:0]po,控制右端数码管亮灭

output [7:0]se,控制右端数码管显示数字

output [3:0]pos,控制左端数码管亮灭

output [7:0]seg,控制左端数码管显示数字

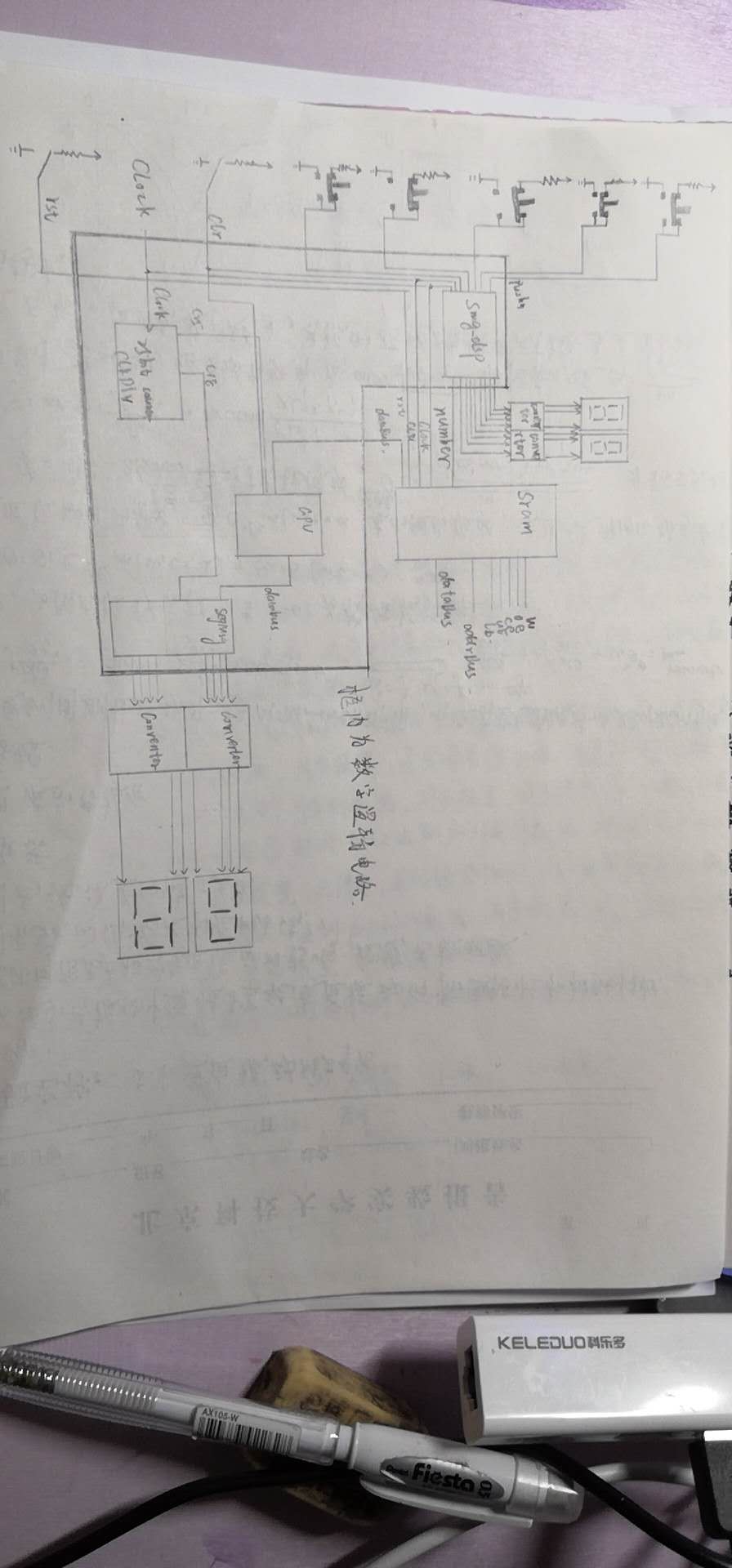
output [18:0]addrBus,19位地址总线

output ce,ub,lb, w,oe输出到芯片的控制信号

（2）系统行为描述

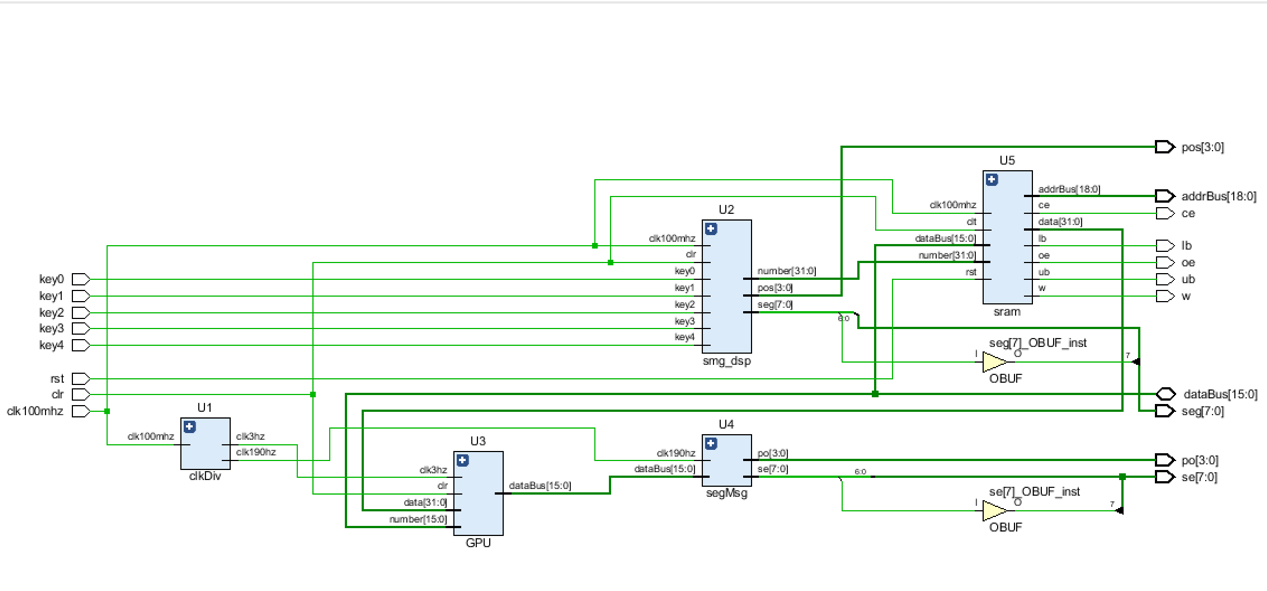
首先是学号的输入状态，将sw7开关拨至高电平，因为我的系统设置为高电平时输入数据有效，所以分别控制五个按键可以实现数码管显示数字的左右移以及数字的加减，中间按键用于最终确定数字，并传入sram中，进行写操作，在开关拨至低电平后，sram进行读操作，每次读四位，并将读出的四位传入另一模块GPU中，此后GPU模块进行数据移位操作，并将移位后最终结果传给segMsg模块，实现在右端数码管的循环滚动效果。

（3）系统级电路结构设计图

****

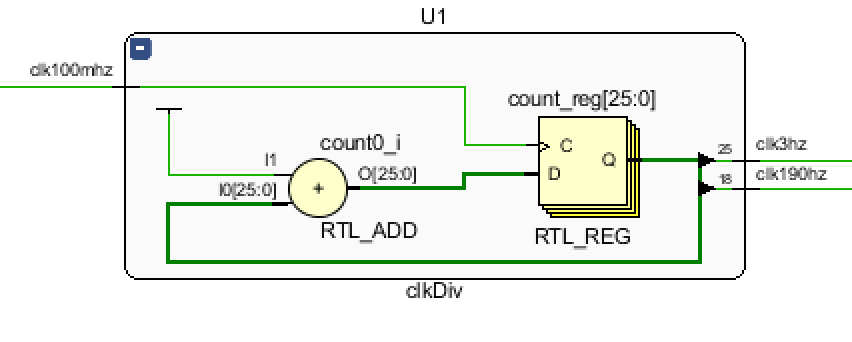
**5、实验3.3实现**

（1）顶层模块RTL电路结构图及说明

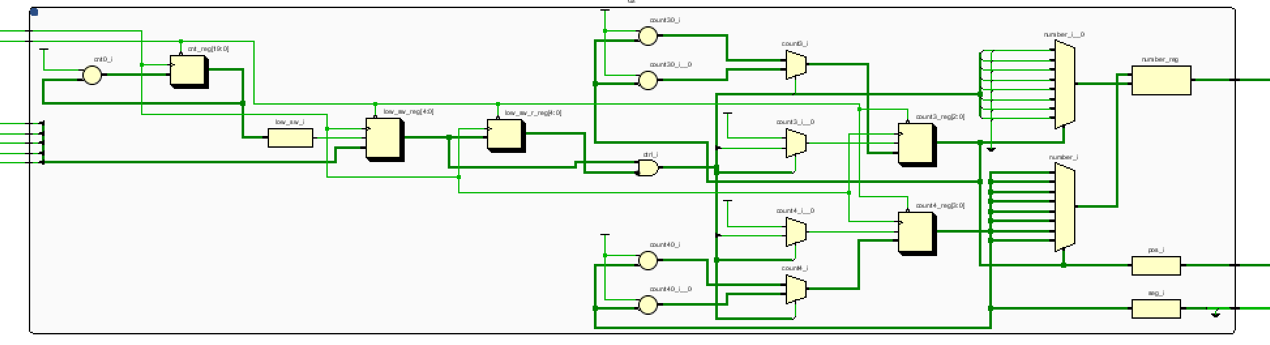


说明：在Verilog描述生成的顶层模块RTL电路结构图中，并没有数码管按键等外部电路，而且在两个数码管输出端还多了一个输出缓冲标志（带三角形），在我设计的电路中，没有这个输出缓冲，并且还画出了外设。其他大致相同。

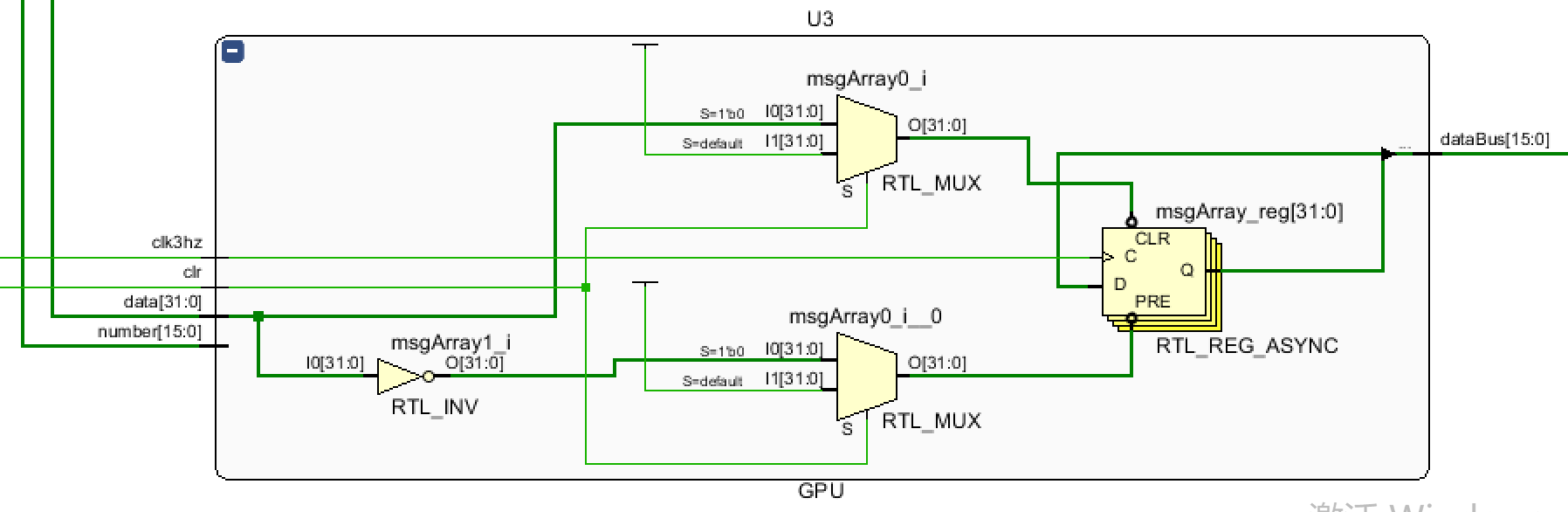
(1)子模块1 RTL电路结构(clkDiv)



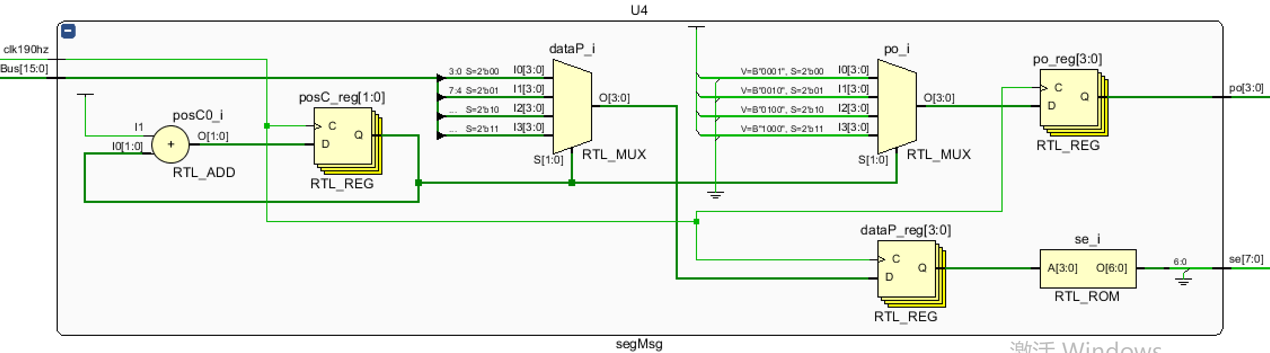
（2）子模块2 RTL电路结构(smg\_dsp)



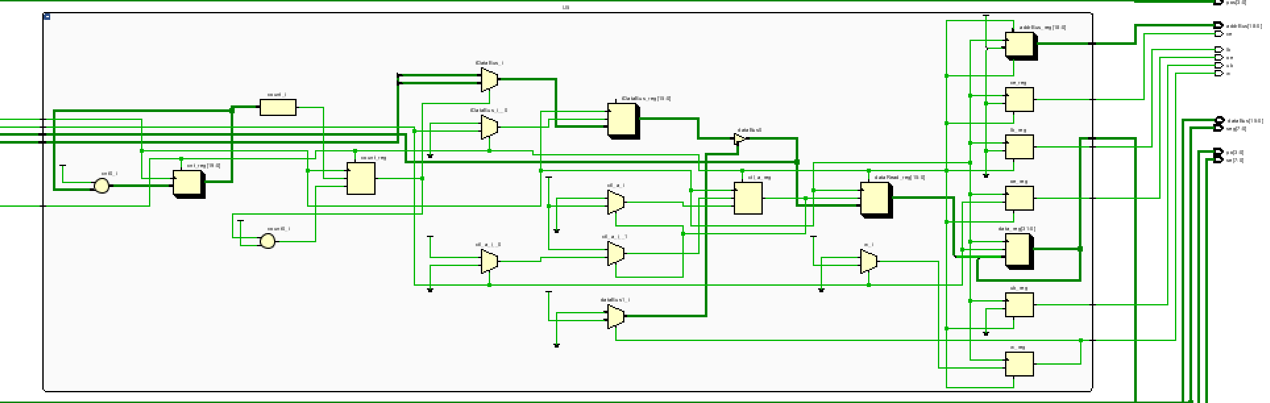
（3）子模块3 RTL电路结构(GPU)

****

（4）子模块4 RTL电路结构(segMsg)

****

（5）子模块5 RTL电路结构(sram)

****

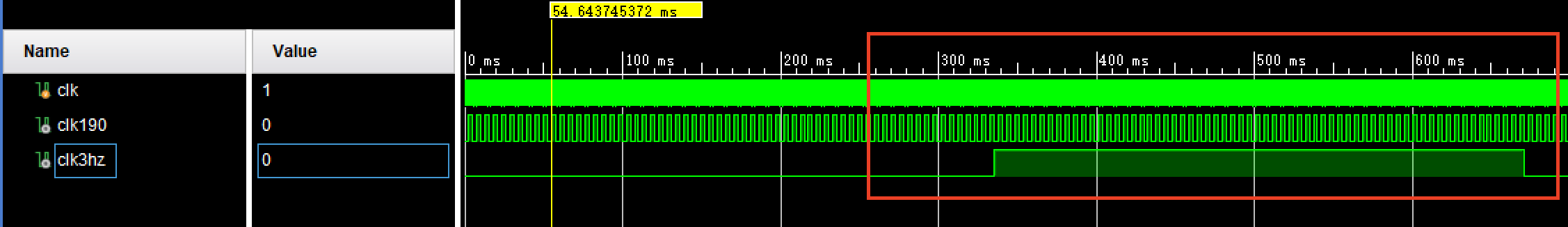
**6、实验3.3仿真验证**

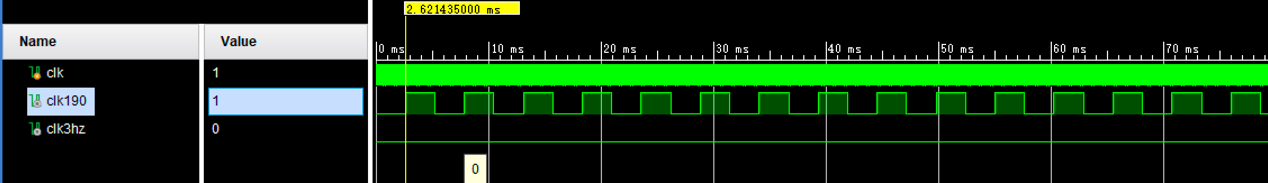
（1）模块级仿真

模块1测试方案设计说明

模拟输入clk100mhz,设置为周期10ns，占空比50%的时钟信号，测试输出clk3hz,clk190hz,这么设计就是还原了板中真实状况。

模块1仿真结果及说明



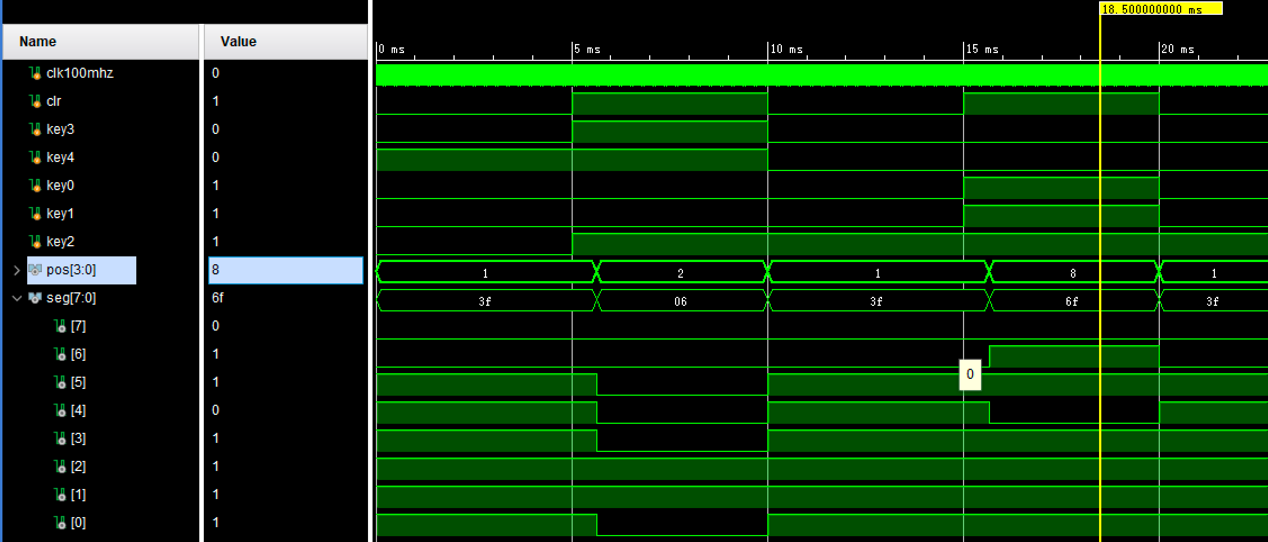


二图中，第二张图记录了clk190第一次上升沿时间，算得2^18=262144，而clk一个时钟周期10ns，经换算，刚好为2.62144ms，检验正确，第一张图中，同理，clk3hz经同样方法测算，2^25\*10^(-5)=335.54432ms，等到加到第26位为一，第25位为零时，刚好为671.08864ms，验算成立。

模块2测试方案设计说明

模拟输入时钟信号，清零信号和五个键值，由于难以控制时间，这里采取起初为左移加一模拟，累加一段时间后，模拟右移减一，之后输出数值number和数码管选通值[3:0]pos以及数码管显示数值[7:0]seg，number用于判断每次按键值是否都有效存入，pos和seg则用于检测数码管上显示是否能按照键值变化。

模块2仿真结果及说明

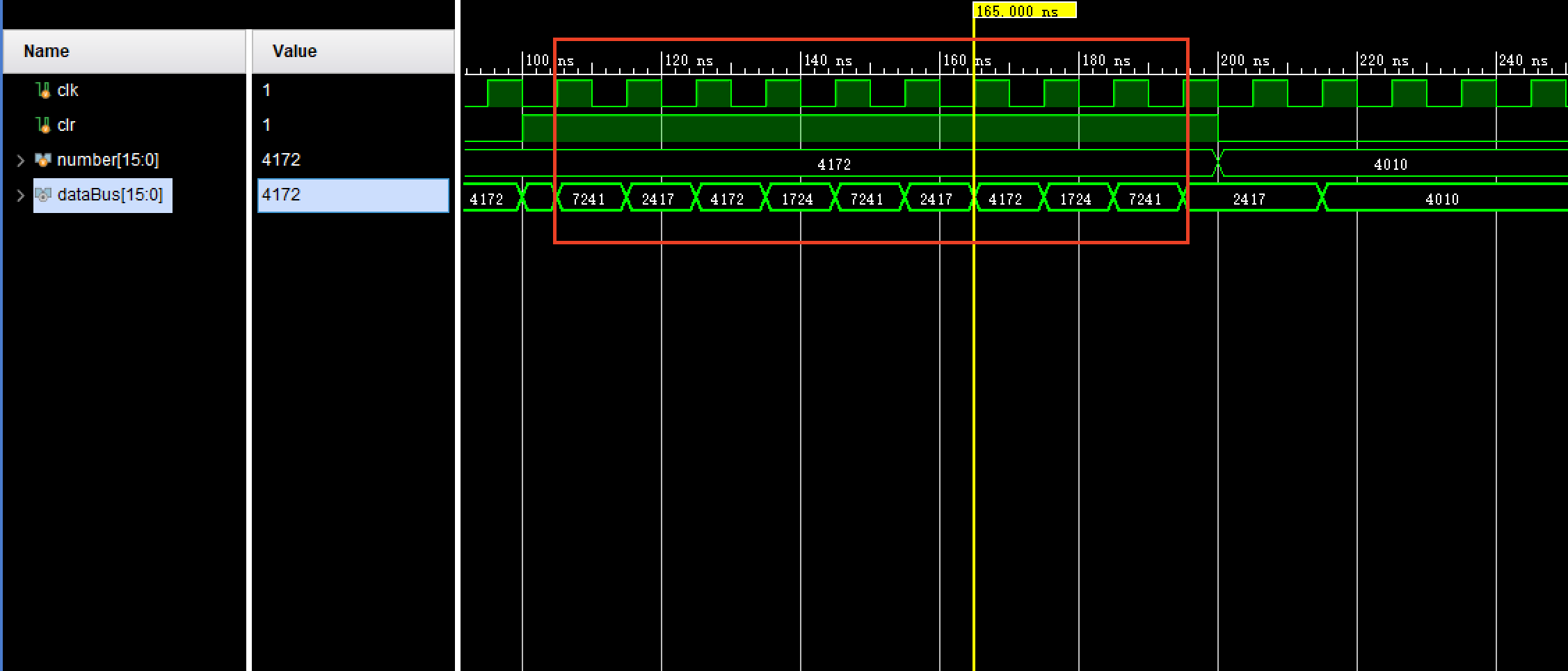


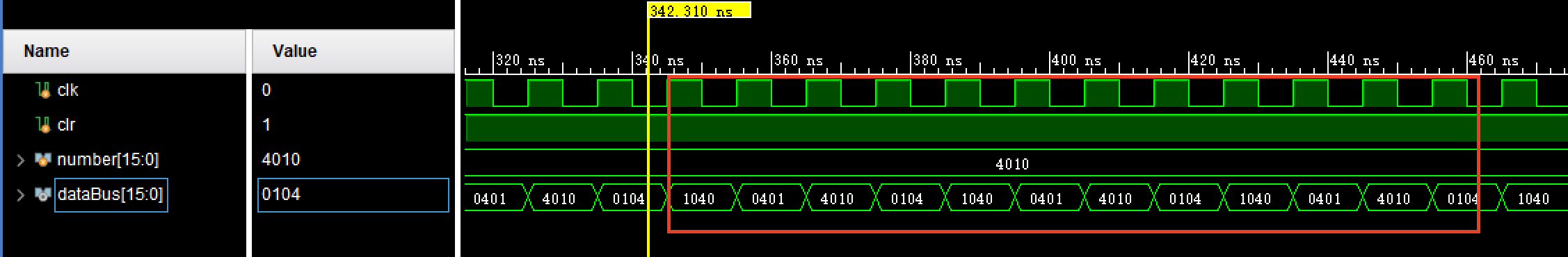
图中，因为刚开始为持续的按键所以左移加一，之后右移减一，由于持续时间过长，所以自动移位位数较多，基本体现了思想。

模块3测试方案设计说明

模拟仿真时钟信号，每5个时间单位改变一次，清零设置为仅初态时为0，其余为1，设置传入data值即为已设好常量，如学号41724010，有关动态数据的改变在另一模块输出中模拟检测，模拟输出databus，这么设计可以充分检验每个输入的作用。

模块3仿真结果及说明



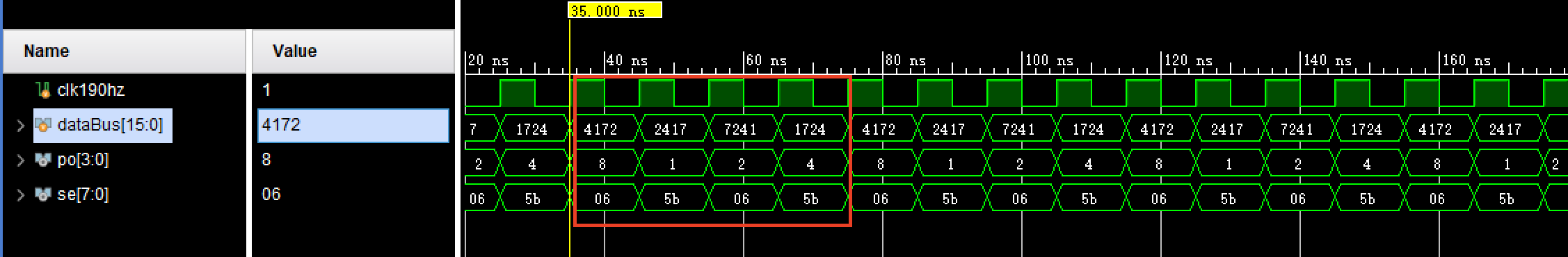


由图中可看出，模拟信号分两次加入，表示sram接口每次只传输四位，两次叠加即可传输完整学号，第一次传4172，并进行循环，第二次传4010，并开始循环，结果得到验证。

模块4测试方案设计说明

对于这一模块的测试，主要在于判断数码管上选通端的亮灭以及数字对应的数码管端是否能相应输出正确数字的对应七根亮灭状况。所以模拟输入时钟信号与变化数据，相应模拟输出选通端和显示数值。所以选择模拟随着时钟信号，数据4172循环，以此模拟现实状况。

模块4仿真结果及说明

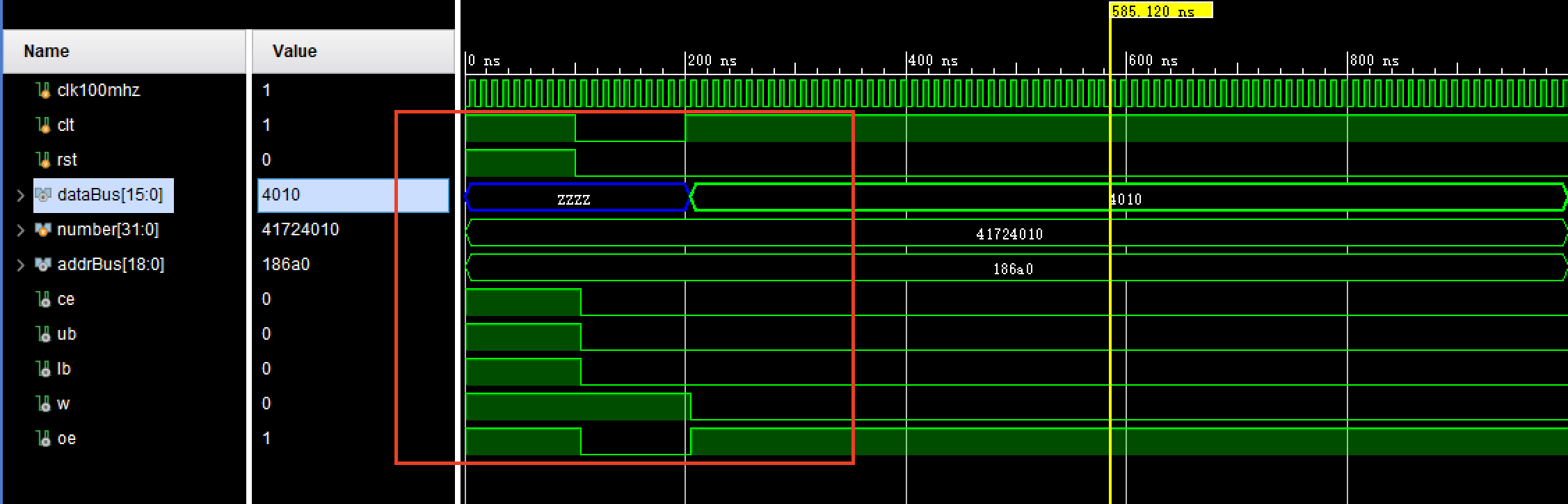


通过图中圈出，就可以判断检验成果成功，虽然看似对应的是4172，但考虑延时的存在，所以实际值为1724，当出现1724时，选通1000，即最左边的数码管，显示数字1，以此类推，皆成立，所以结果正确。

模块5测试方案设计说明

模拟输入时钟信号，每5个时间单位改变一次，rst置位刚开始为一，后为零，clt控制读写状况，刚开始为写状态，为1值，之后因为sram读写很快，很快即可进入读状态，此时clt为零，另外则为模拟输入数据number，固定为某一值，以及inout读写数据总线，测试输出数据以及地址总线，和使能信号，这么设计可以很好的验证sram的真实输出。

模块5仿真结果及说明

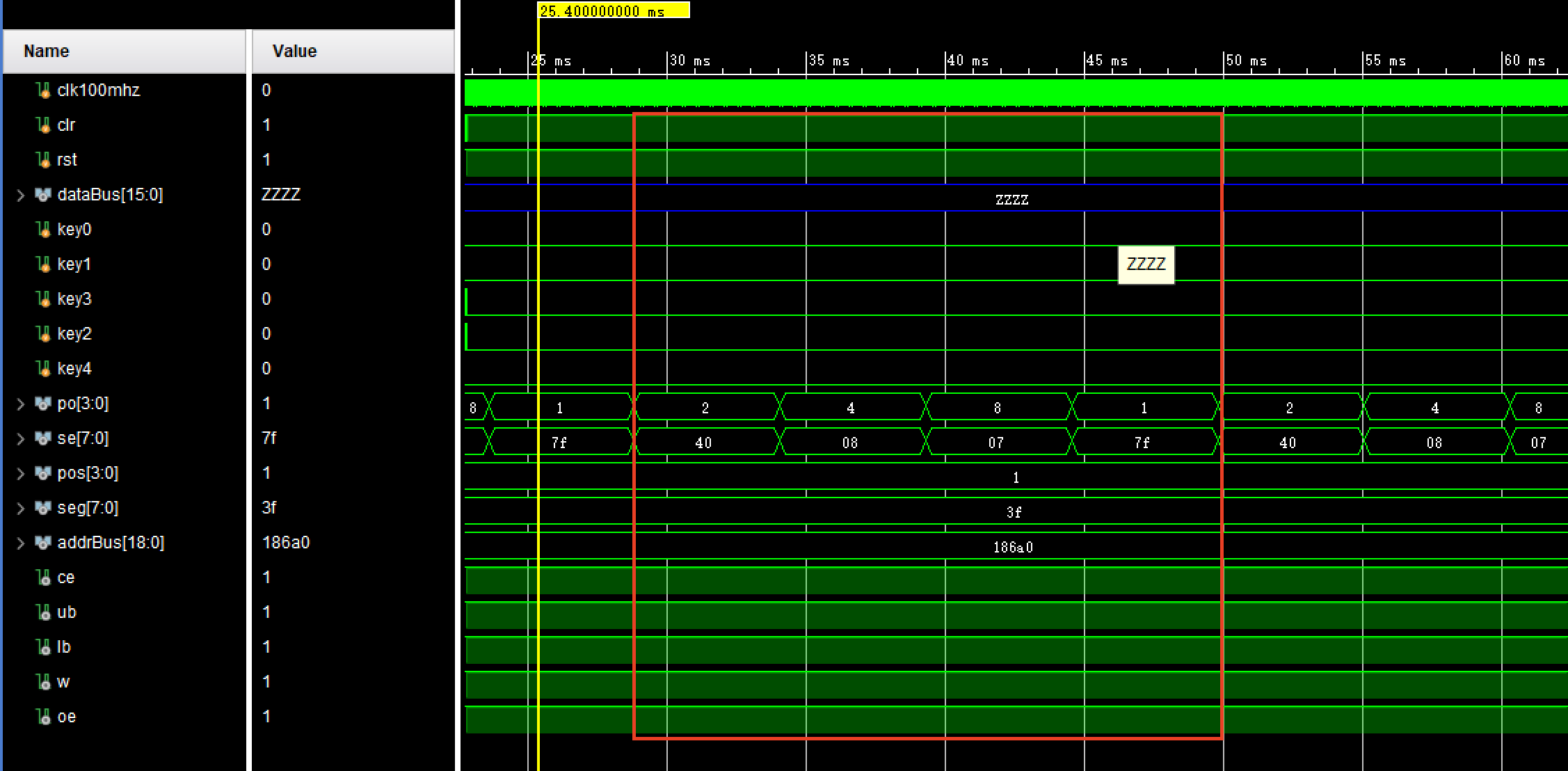


在红框中，可以清晰看到，起初rst信号起作用，使得使能端皆为1，此时不做任何操作，并且dataBus为高阻，此后clt信号起作用，先为1，意味着写数据进sram，此时w仍为高电平，所以dataBus仍为高阻，之后为零，此时读状态，使能端使得高八位低八位都能读出，并且成功读出4010.而地址经运算即为存储位19’d100\_000，检验成功。

顶层模块测试方案设计说明

模拟输入时钟信号，清零置位端，以及五个按键随时间的变化，输出相应数码管以及sram的有关输出，这样可以检验整体情况，得知下到板中实际情况，因为具体情况已经在各个模块中试验过，所以在顶层模块中着重看每个量的对应关系。

顶层模块仿真结果及说明

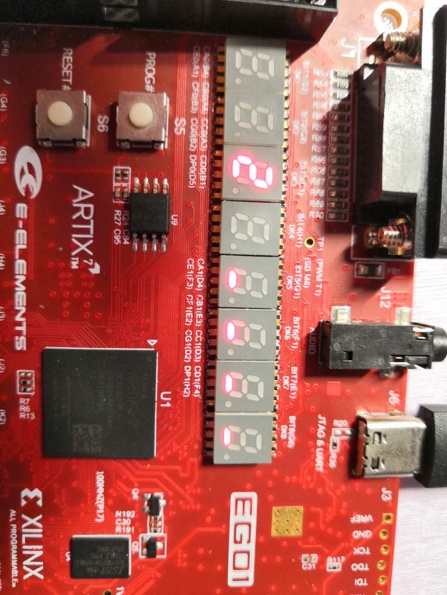


图红框中，可以注意到，在输入按键结束后，将rst置位，sram变成不读不写状态，而值已经传至segMsg中，显示值并随时间改变，检验成立。

（2）板级测试验证

（a）功能1测试

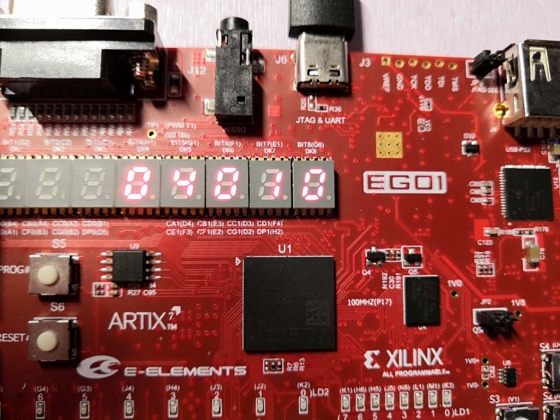
操作：在左端数码管中输入学号

现象：

现象说明：在左端输入时，一次只显示一位，此时确定该位数据，右端等待sram写入读取数据之后显示。

（b）功能2测试

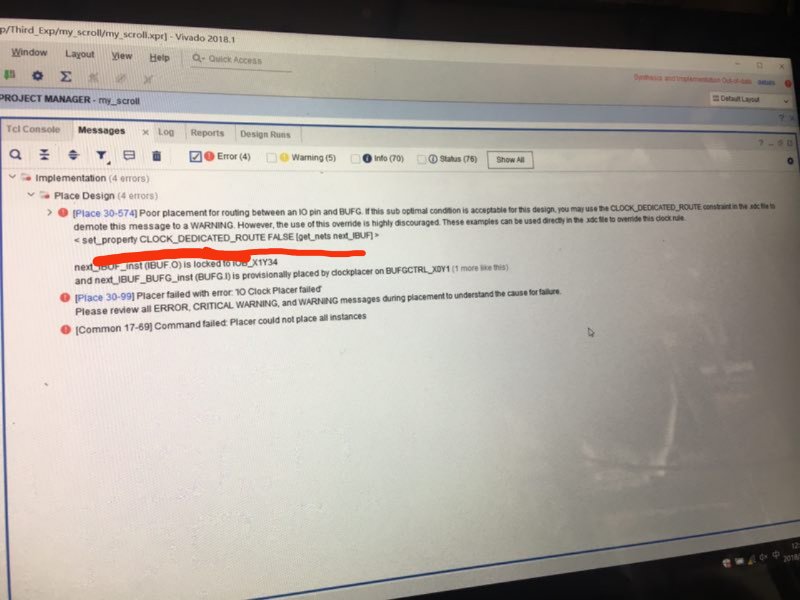
操作：观察右端显示数据是否正确

现象：

现象说明： 在显示数据以后，左端数码管置零，右端数码管开始滚动显示学号，如图4010则为我的后四位学号。

**7、实验中遇到的问题、现象及解决方法**

问题1：写的过程中，首先遇到了.xdc文件的报错，提示时序错误，建议加上set\_property开头的两行，加上后，果然没有报错，但仍存在警告消息，具体如图，



现象：系统报错，就算加上报错提示加入信息后，仍然有critical warning

问题原因：在时序逻辑中，always块里加入了非时钟信号，时序存在差异，系统报错。

解决方法：将always里判断条件改为posedge 或negative，也可以按照报错信息加入提示信息，但实际效果差强人意，所以建议采用第一种方法。

问题2：帮他人解决：报错显示multiple driver nets

现象：报错显示某个变量有多个地方赋值，导致系统报错

问题原因：因为时序问题，一个reg变量在always里被赋值后，若在别的地方仍然被赋值，不管逻辑上两者是否相斥，系统都默认报错

解决方法：设计一个中间变量作为赋值，或者将同一个变量分条件直接在一个always里执行。

问题3:帮他人解决：不知如何延时，在延时功能处使用#500等表示方式

现象：编译不会报错，但在板上综合实现不了，时间会乱。

问题原因：使用#500等待时间，虽然逻辑上讲得通，但是在时序电路中，#500这样的电路完全无法综合，导致所有延时都没用，相当于消抖根本没有用。

解决方法：使用一个计数器代替延时，当计数器值满足某一条件时，才执行按键相应操作，这样可以很好的解决消抖问题。

问题4:帮他人解决：&与&&的区别

现象：当只有一位运算时，两者无区别，但位数增加，两者区别很大，如10&01=00，10&&01=1算出结果不同。

问题原因：&为位运算，&&是与运算，&一位一位得出结果，而&&根据最终值算出结果。

解决：具体问题具体分析，查看得出何值后，确定运算方式。

问题5: 系统报错：[DRC 23-20]规则违规（CFGBVS-1）在current\_design中未设置CFGBVS和CONFIG\_VOLTAGE电压属性。配置组电压选择（CFGBVS）必须设置为VCCO或GND，并且CONFIG\_VOLTAGE必须设置为正确的配置电压，以确定支持0的引脚的I / O电压。

现象：系统报错，并有提示信息在.xdc文件中加入某句。

问题原因：正如报错信号，配置组电压选择必须设置为VCCO 或GND。

解决：加入建议指定这些使用GUI中的“编辑设备属性”功能或使用以下语法直接在XDC文件中使用：

set\_property CFGBVS value1 [current\_design]

# value1是VCCO或GND

set\_property CONFIG\_VOLTAGE value2 [current\_design]

# value2是提供给配置库0的电压

**8、本次实验心得体会**

**之前接触过用C语言写按键的延时以及动态数码管数字的显示，但这次是第一次使用verilog来写消抖和数码管，相对来说，思路虽相近，但采取的方法不尽相同，如，C语言习惯上使用延时函数来判断按键前后变化，在延时函数中，用一个while不断实现某数的减法，直到减完，时间也到了某一特定值，而在verilog中，习惯上使用计数器，相当于一个数组从零不断的加，加到某特定值时，检查一次按键，当加到最大，又重新归零开始新的二进制加法，加深了对延时的认识。**