**中国海洋大学计算机科学与技术系**

**实验报告**

**姓名：岳宇轩 年级：2019** **专业：** 19慧与

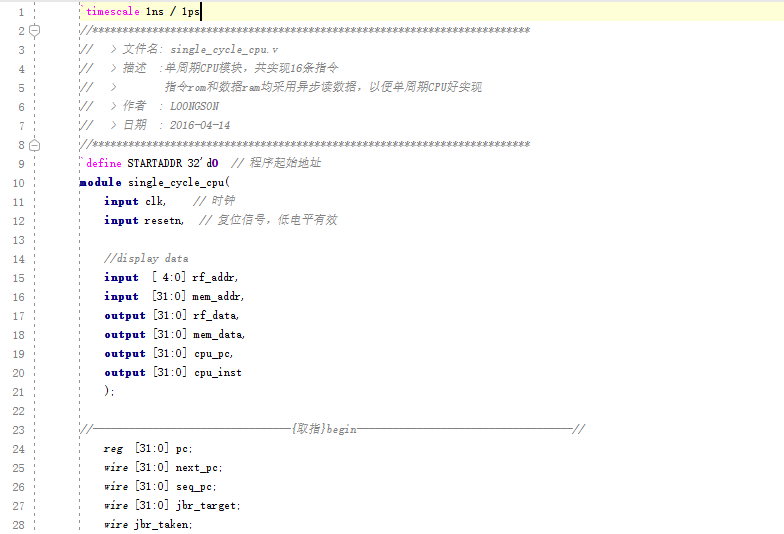
**科目：计算机组成原理** **题目：单周期cpu**

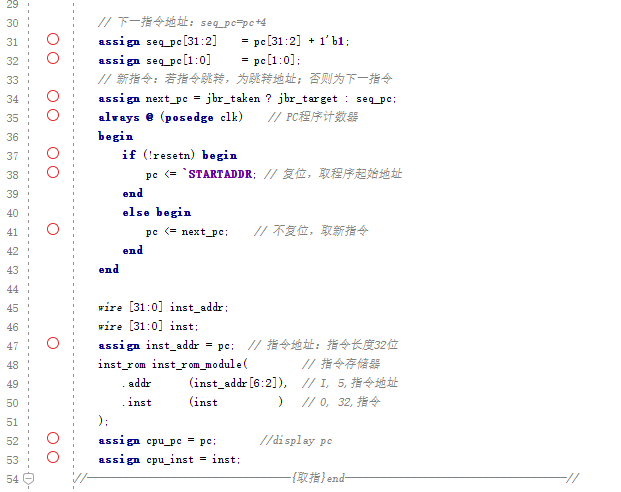
**实验时间:** 2021 年5 月20日  **实验教师:** 张巍

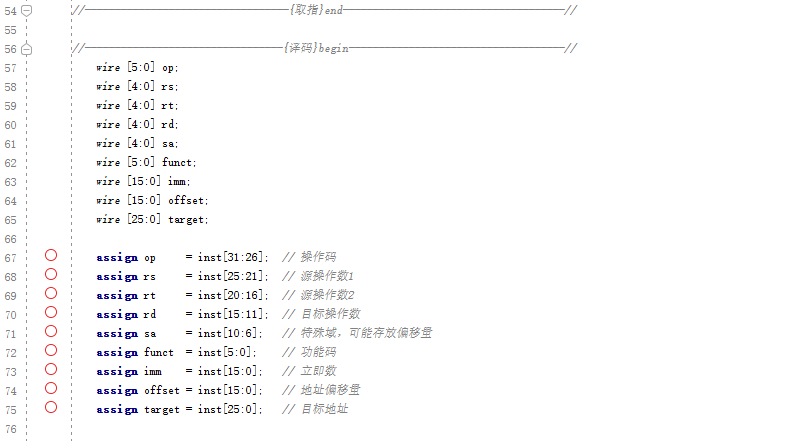
**一、实验结果及截图分析：**

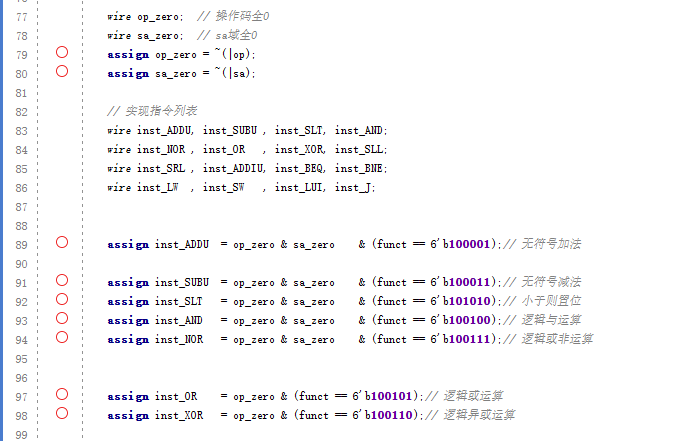
（※代码挖空的部分必须截图或复制）

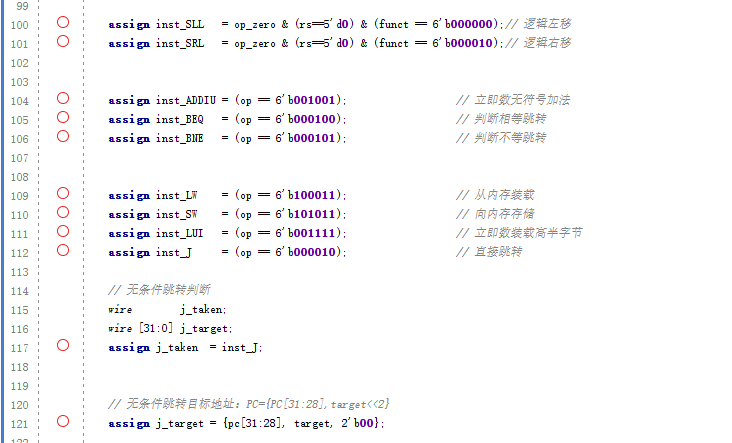
1. **代码补全**

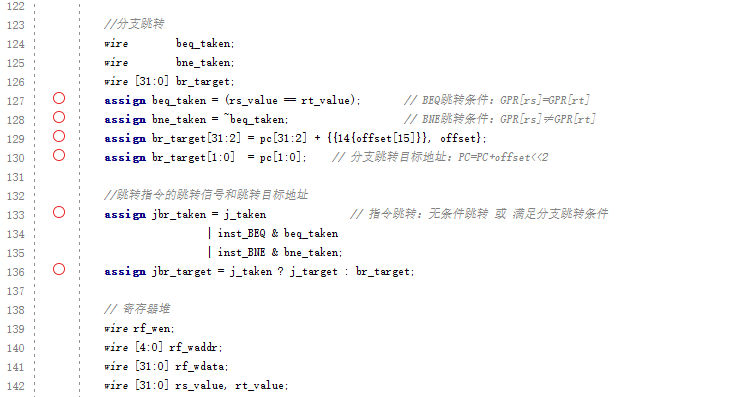


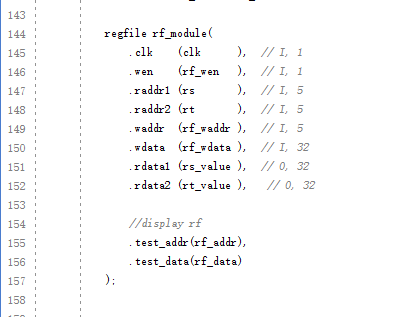


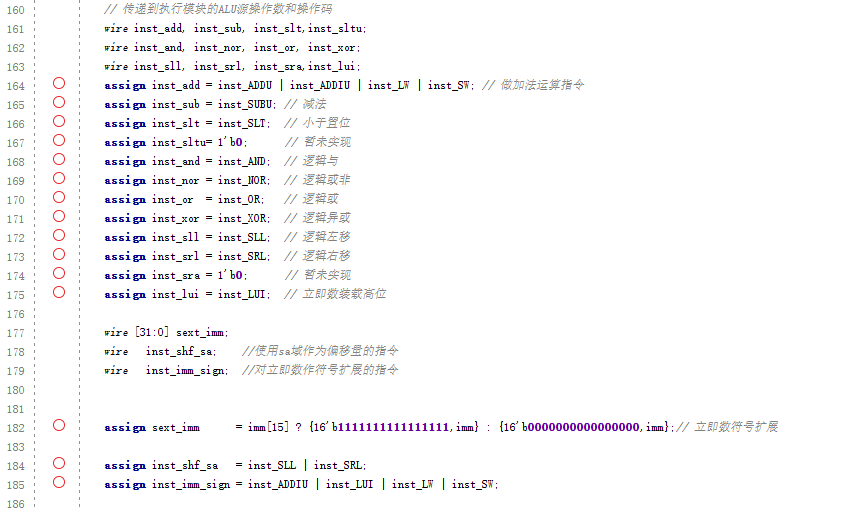


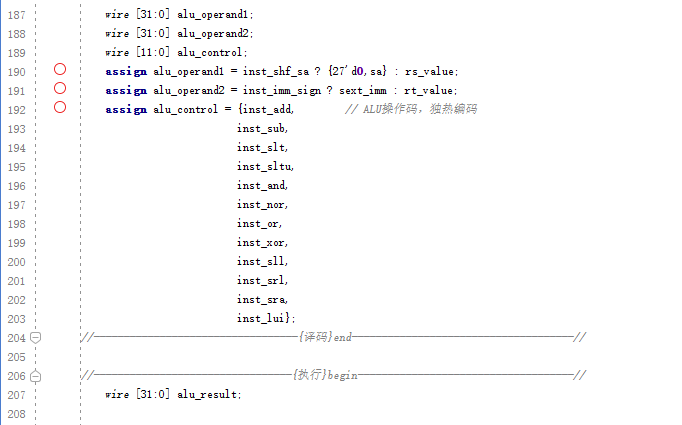


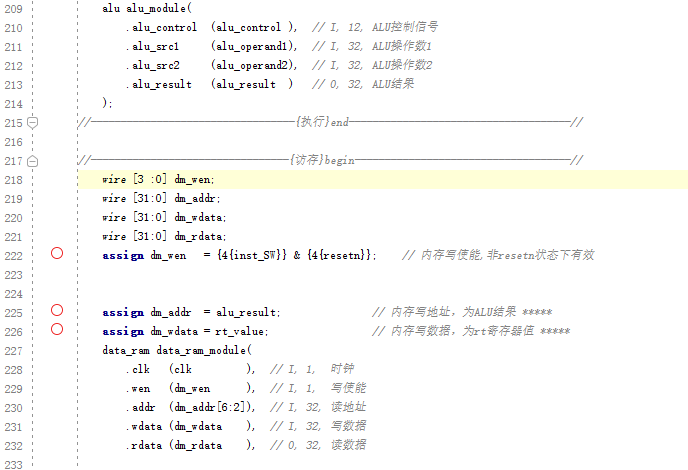


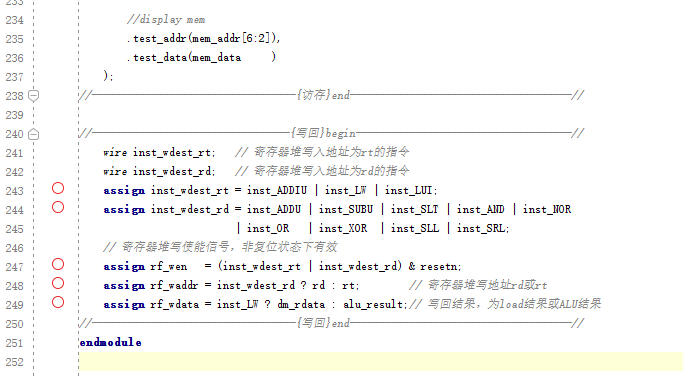








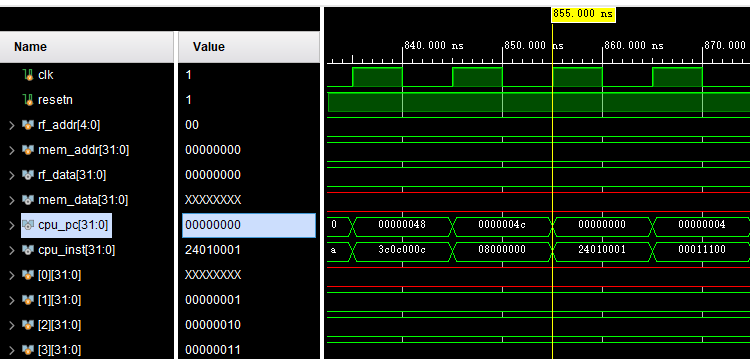




1. **仿真图像**

**为了方便看实验结果，我把所有寄存器也显示了**

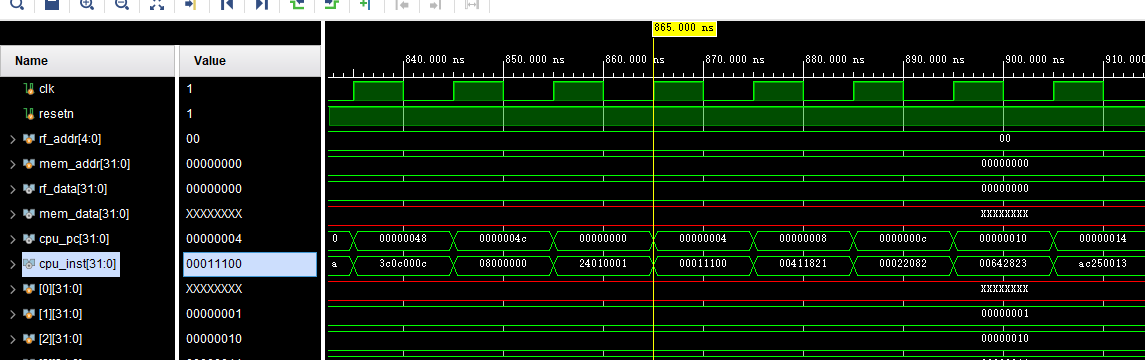
**00H**





从指令寄存器中取出第一条指令，其指令编码为24010001h, 指令地址为00h,对指令进行汇编得到汇编指令addiu $1 ,$0,#1 ，指令的操作是零号寄存器（0）无符号加 立即数1，结果送1号寄存器。 运行结果为1号寄存器值变为1H，通过仿真图像观察，结果正确

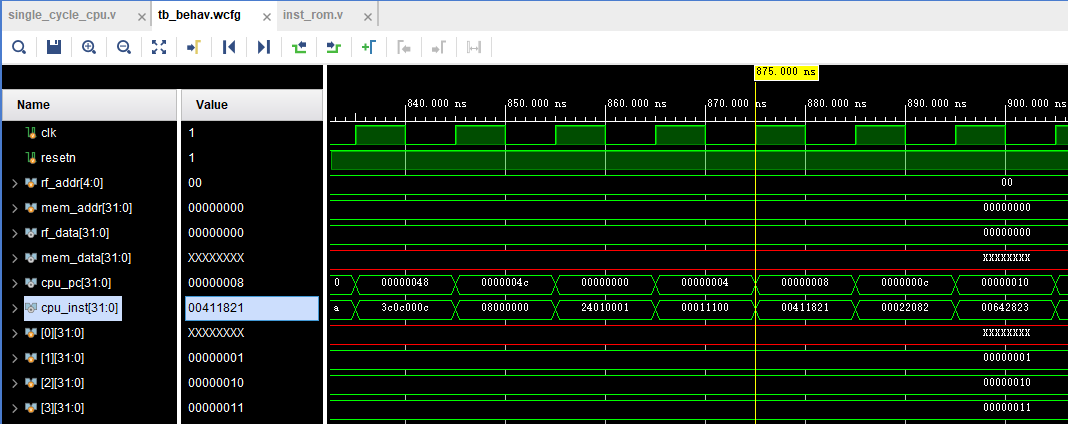
**04H**





第二条指令地址为04H，从IR中取出的指令编码为00011100H，对应的汇编指令为sll $2 ,$1,#4，指令操作为1号寄存器中的值逻辑左移4位后送2号寄存器。01H左移4位变为10H，指令结果为2号寄存器值变为10H 。 通过仿真图像观察实验结果正确。

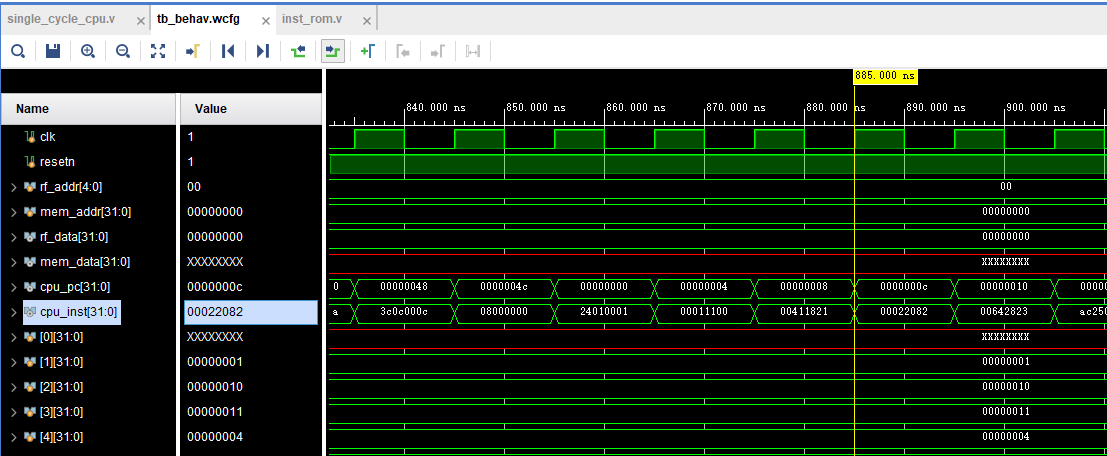
**08H**





下一条指令的地址为08H， 取出的指令编码为00411821H，对应的汇编指令为addu $3 ,$2,$1，指令操作是将2号和1号寄存器中的值无符号相加，结果送3号寄存器。01H+10H，结果为11H，通过仿真图像可以看到3号寄存器值为11H。

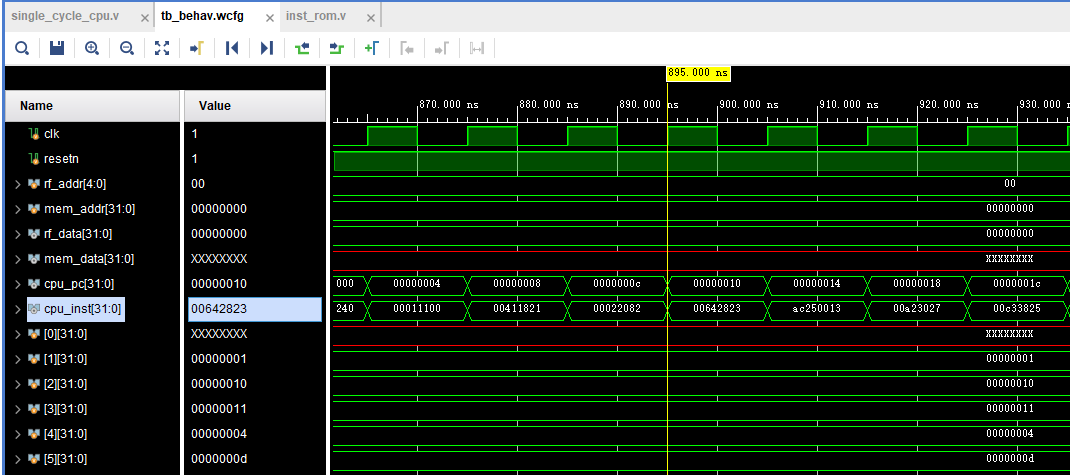
**0CH**





指令地址为0CH，指令编码为00022082，对应汇编指令为 srl $4 ,$2,#2 ，将2号寄存器中的值逻辑右移两位后放入四号寄存器。通过仿真图像可以看到，2号寄存器中的值为10H，即0001 0000，逻辑右移2位后为0000 0100，即04H，04H送入4号寄存器，实验结果正确。

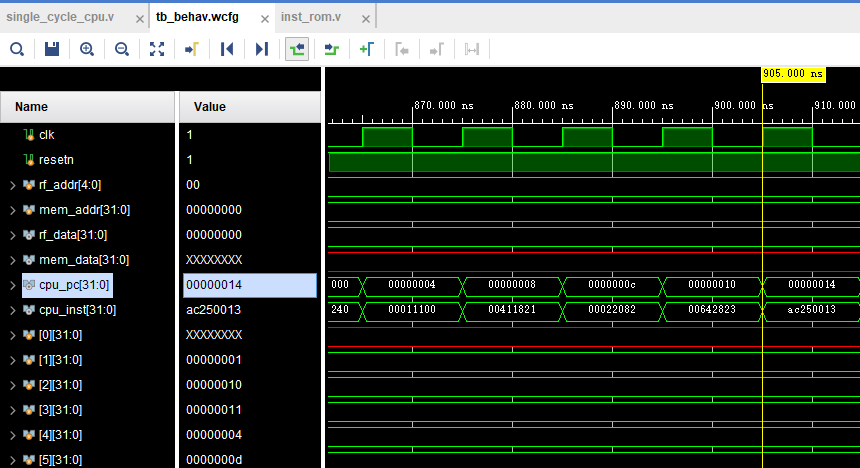
**10H**





PC+4来到了10H，取出指令的编码为00642823，对应汇编指令为subu $5 ,$3,$4 ，指令操作为3号寄存器中的数无符号减4号寄存器中的数，结果送5号寄存器。从仿真图像中可以看出，3号和4号寄存器值分别为11H和04H，结果为0DH，送5号寄存器，5号寄存器值为0d，结果正确

**14H**



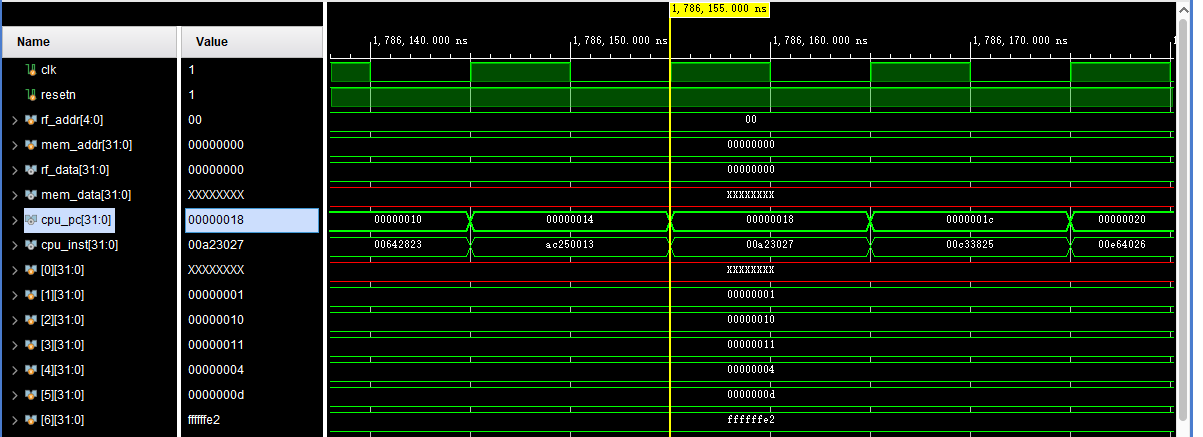


指令编码为AC250013，对应汇编指令为sw $5 ,#19($1)，进行操作为：寄存器1中数+19作为访存地址，将5号寄存器中的数据加载进该地址的内存中



alu运算结果为访存地址14H，dm\_wdata是要写入的数据，与5号寄存器中数据一致，实验结果正确

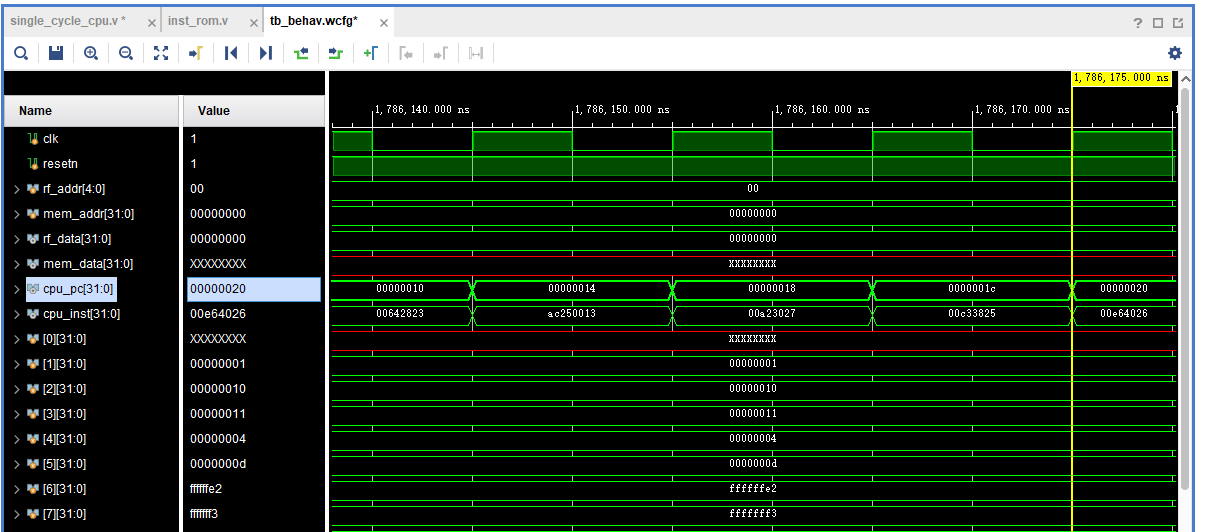
**18H**





指令编码00A23027H，对应汇编指令为 nor $6 ,$5,$2 。寄存器5和2中的值分别为0000 000dH和0000 0010H，进行或非运算结果为ffff ffe2H,放入6号寄存器中。实验结果正确。

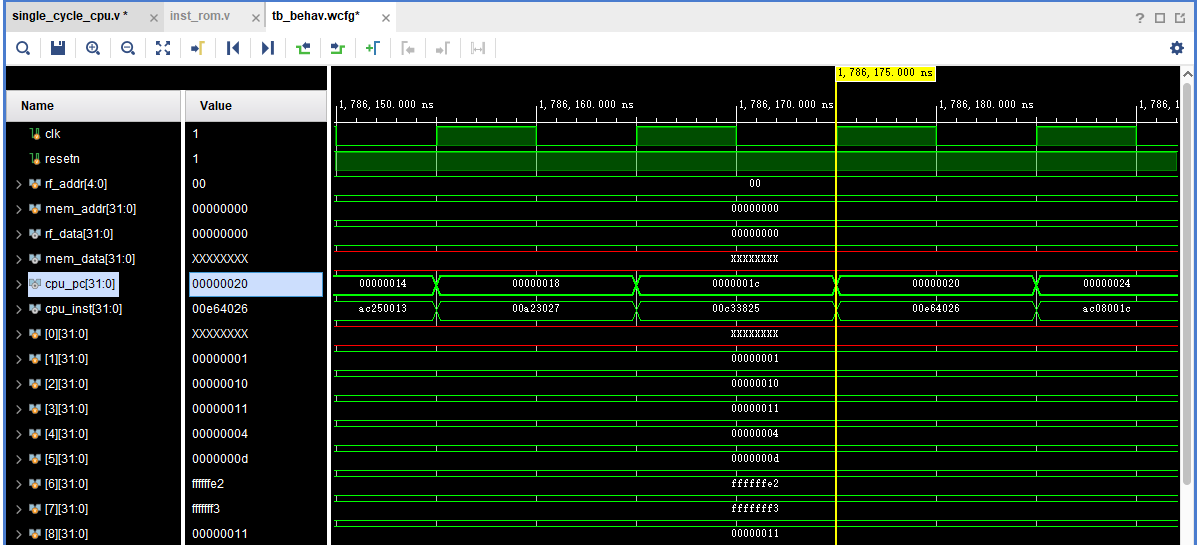
**1CH**





指令编码为00C33825H，对应汇编指令为or $7 ,$6,$3 ，寄存器6和3进行或操作，即ffff ffe2H与0000 0011H进行或操作，结果为ffff fff3 H,放入7号寄存器，7号寄存器为ffff fff3H，结果正确。

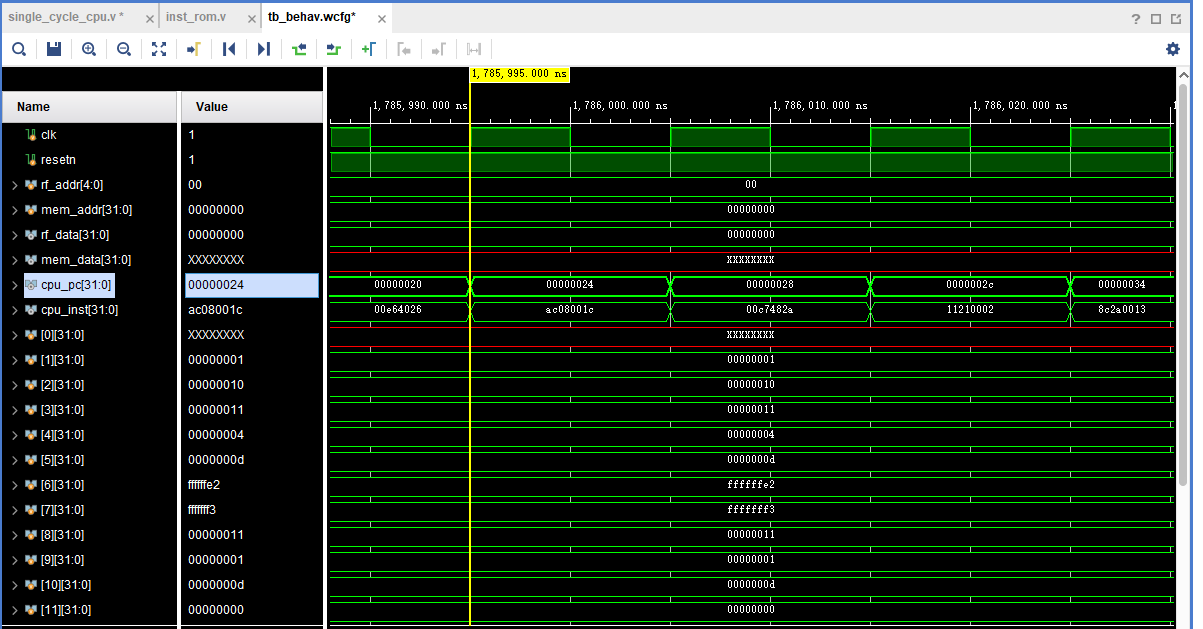
**20H**





指令编码00E64026H,对应汇编xor $8 ,$7,$6 ，寄存器7和6中内容（fffffff3 ffffffe2）进行异或操作，得到结果为0000 0011H，放入8号寄存器

**24H**



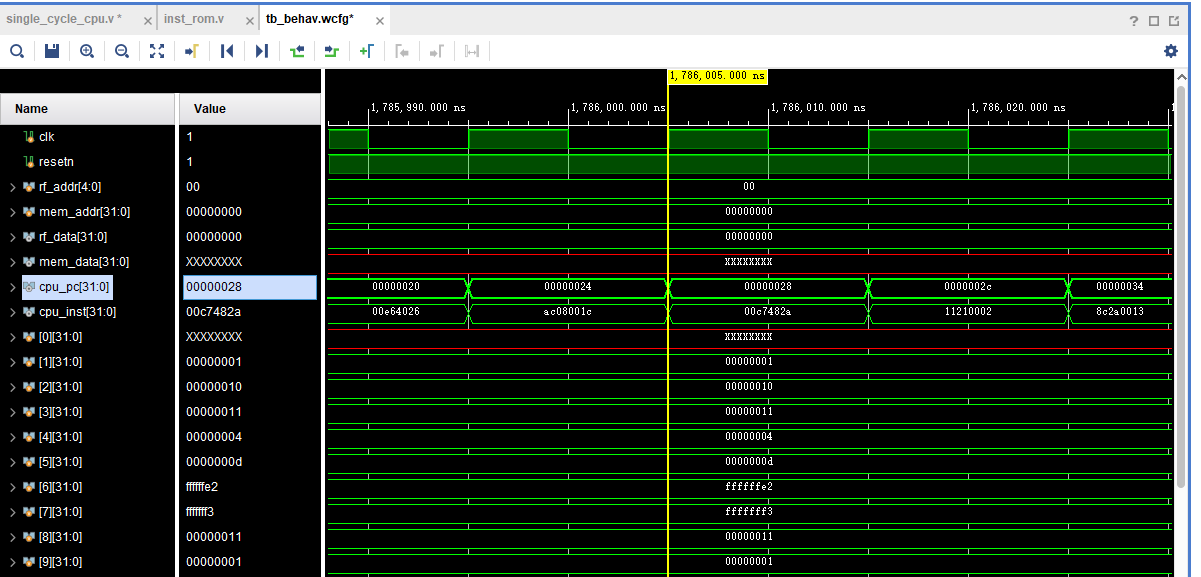


指令编码AC08001C,对应汇编sw $8 ,#28($0)，将8号寄存器中的内容放入内存中地址为28处



alu\_result为内存地址计算，0+28 = 28，即1cH,要写入的数据是8号寄存器的值0000 0011H

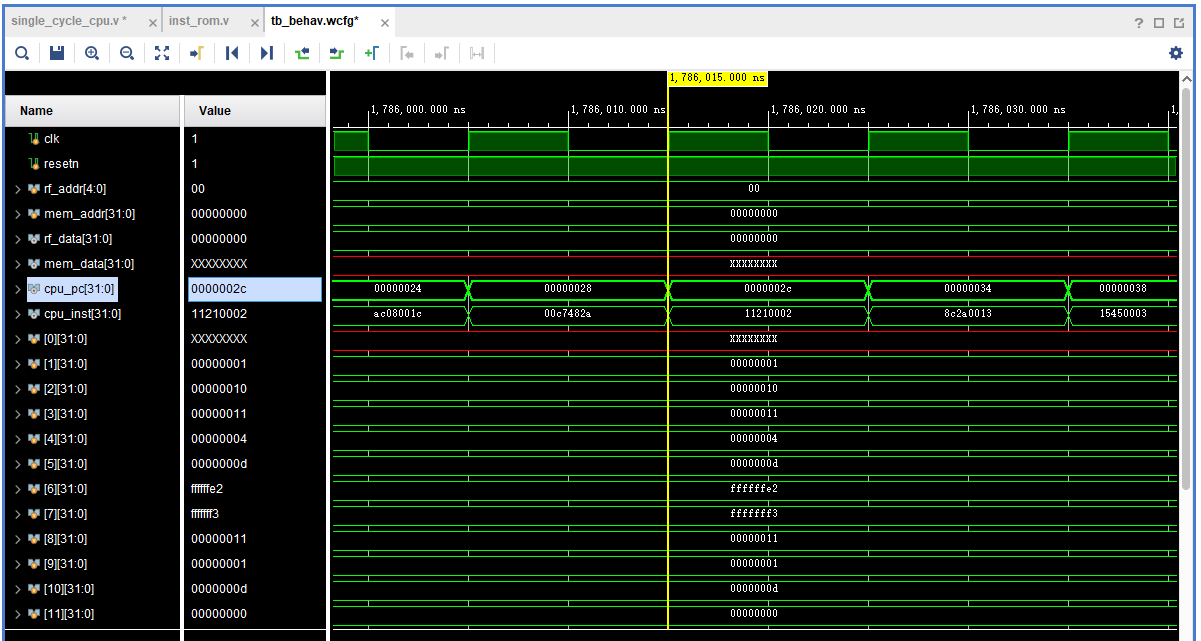
**28H**





指令编码为00c7482aH，对应汇编为slt $9 ,$6,$7 ，指令操作为小于置位。寄存器6中值ffff ffe2 减寄存器7中值ffff fff3，结果小于0，将1存入rd中（9号寄存器）

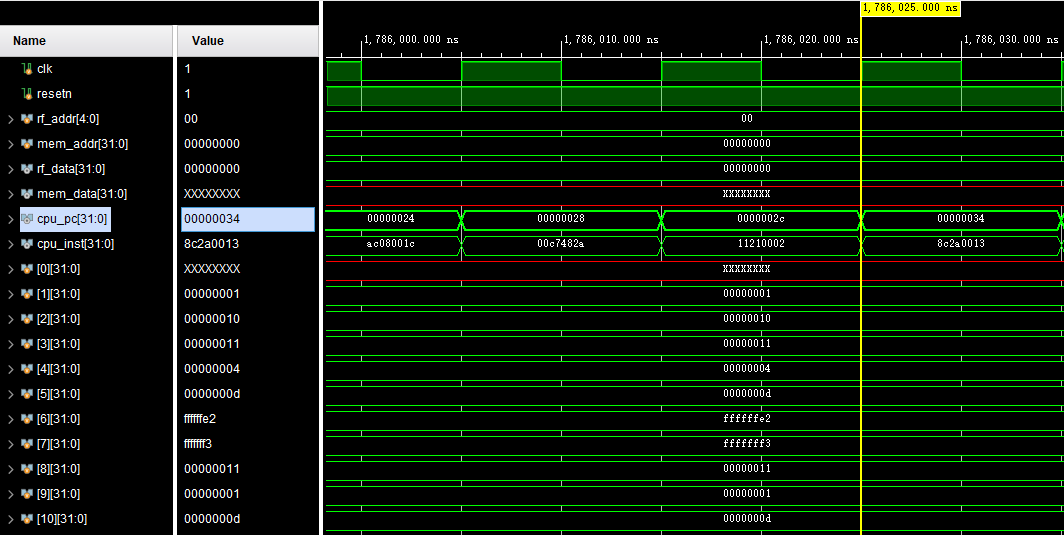
**2cH**





指令编码11210002H，对应汇编beq $9 ,$1,#2。执行的操作为：9号寄存器中的值01H与1号寄存器中的值01H比较，相等，跳转到34H（2cH + 两个4）

**34H**





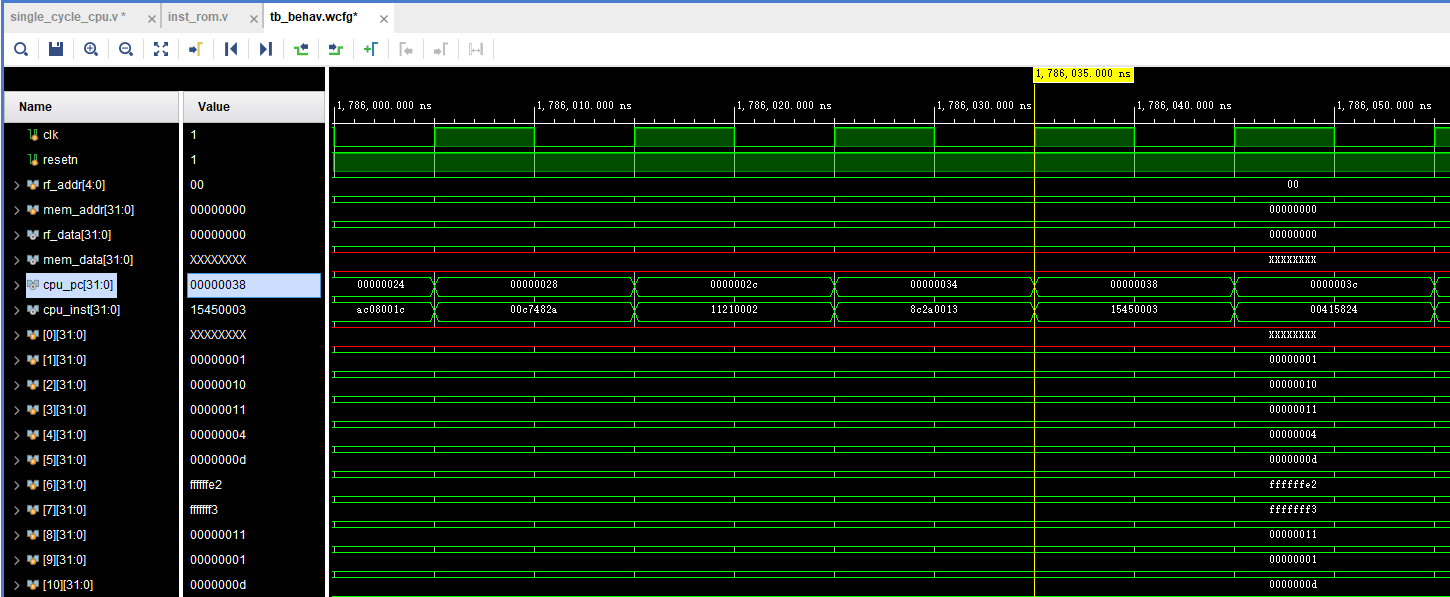
指令编码为8C2A0013，对应汇编指令为lw $10,#19($1)，1号寄存器中内容+19后（14H）作为访存地址，读取内存中数据放入10号寄存器

1号寄存器中内容为01H，加19后是20，也就是14H，



通过仿真图像可以看出，alu计算方寸地址为14H，读出数据为0dH，结果正确。

**38H**

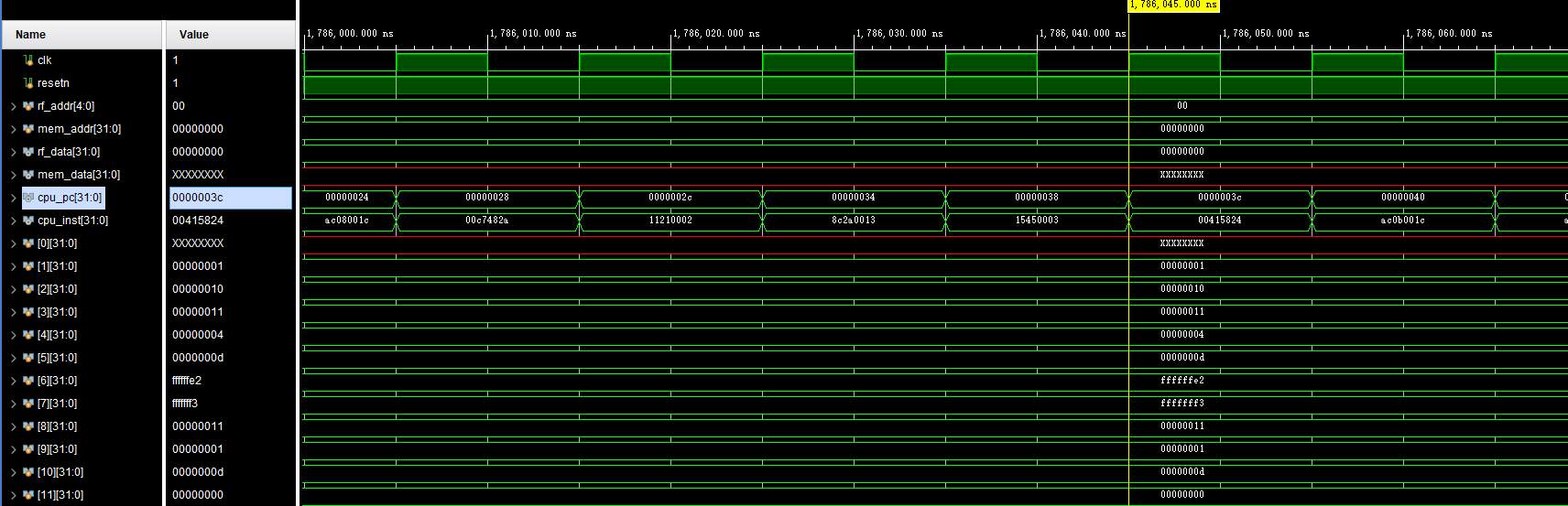




PC+4后指令地址为38H，取出的指令编码为15450003H,对应的汇编代码为 bne $10,$5,#3 ，指令执行功能为比较寄存器10和5中内容，不相等则发生跳转。

通过仿真图像可以看出寄存器10和5中的内容都是0dH，故不发生跳转，下一条指令地址应为PC+4，也就是3CH

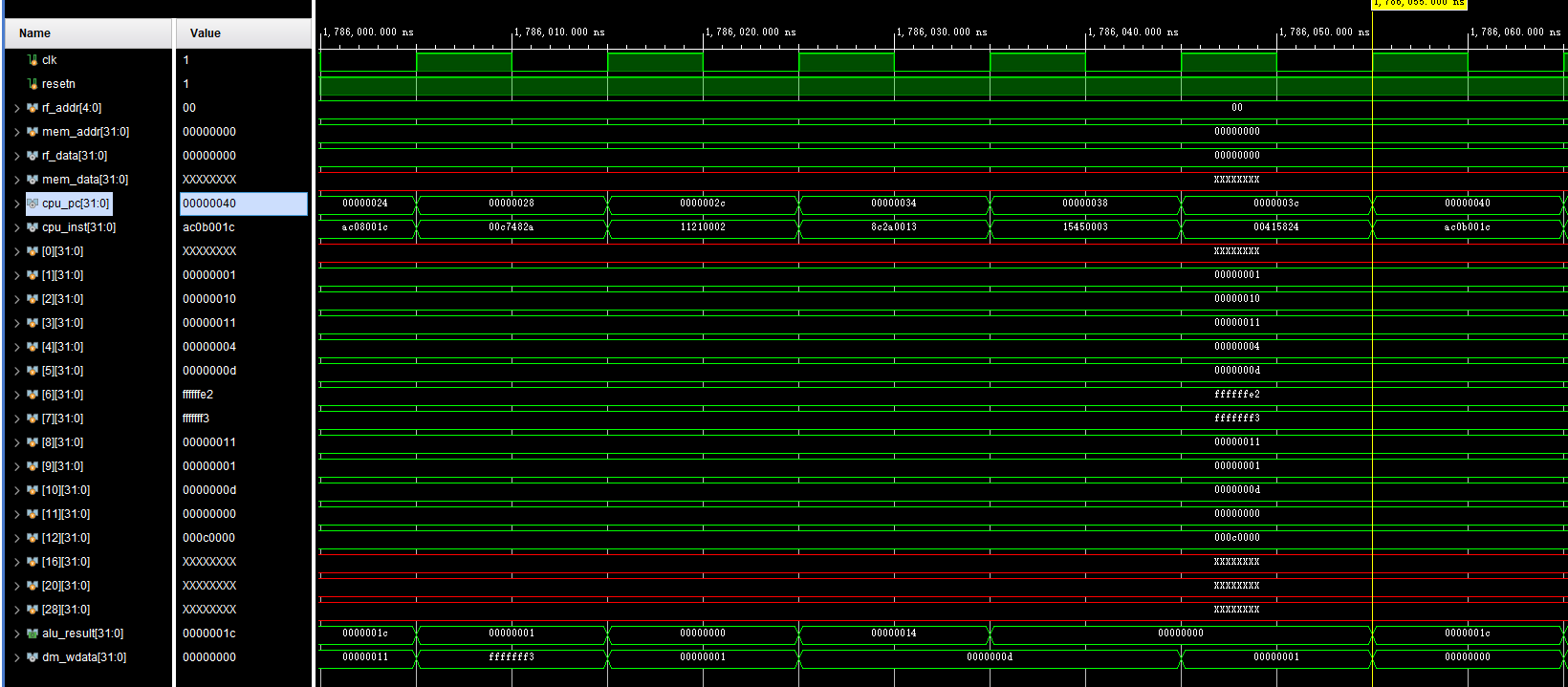
**3CH**





指令编码为00415824H，对应汇编指令为and $11,$2,$1，对2号寄存器和1号寄存器中的内容进行逻辑与，结果送11号寄存器，结果11号寄存器中值为0000 000H，观察仿真图像，结果正确

**40H**

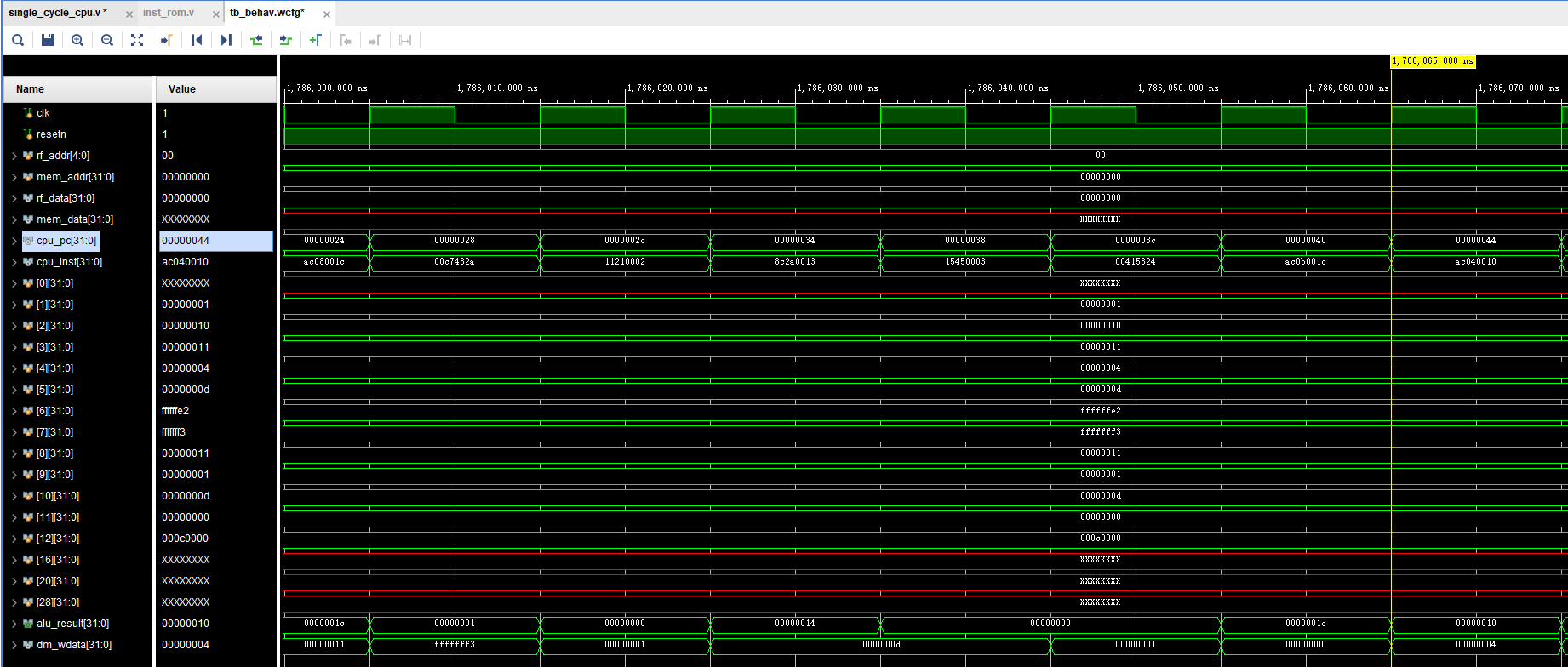




指令编码为AC0B001C，对应汇编指令为sw $11,#28($0)，将11号寄存器中的内容加载进0+28处地址的内存中

通过上图可以看到alu运算地址结果为1c，dm值为11号寄存器中的内容0000 0000H，结果正确

**44H**

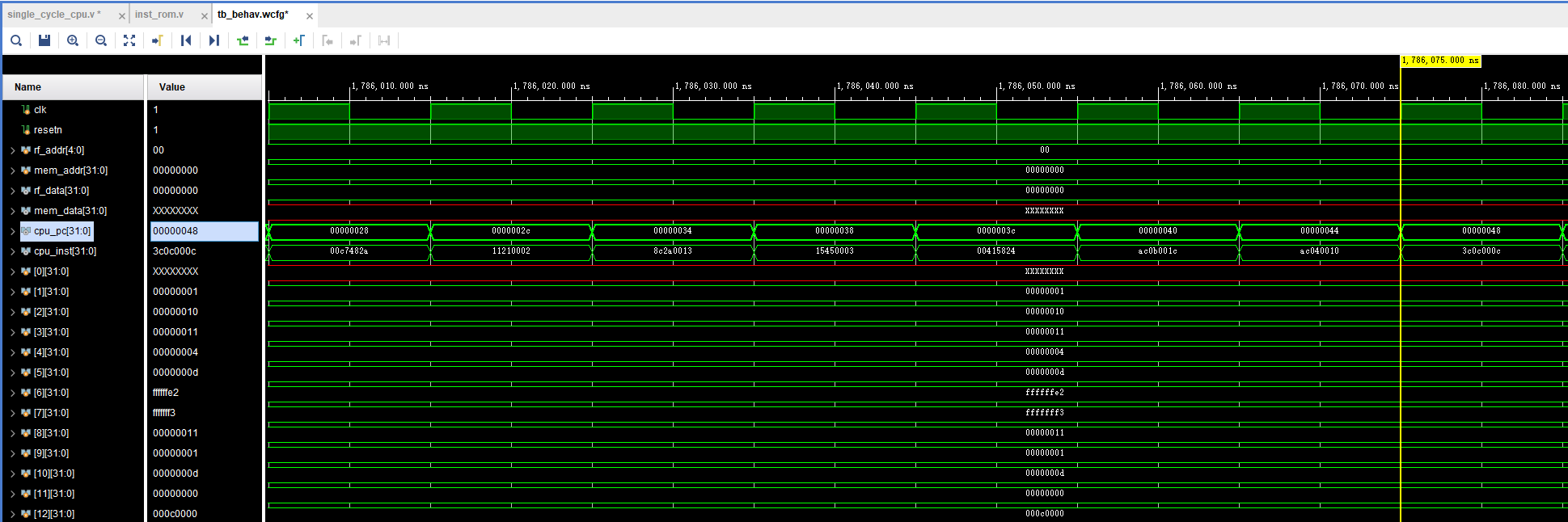




指令编码为AC040010H,对应汇编指令为 sw $4 ,#16($0)，加载4号寄存器中的内容进入内存地址为16处。

通过仿真图像可以看到4号寄存器中值为0004H，alu运算访存地址为10H，写入数据的值dm\_wdata为0004H。结果正确

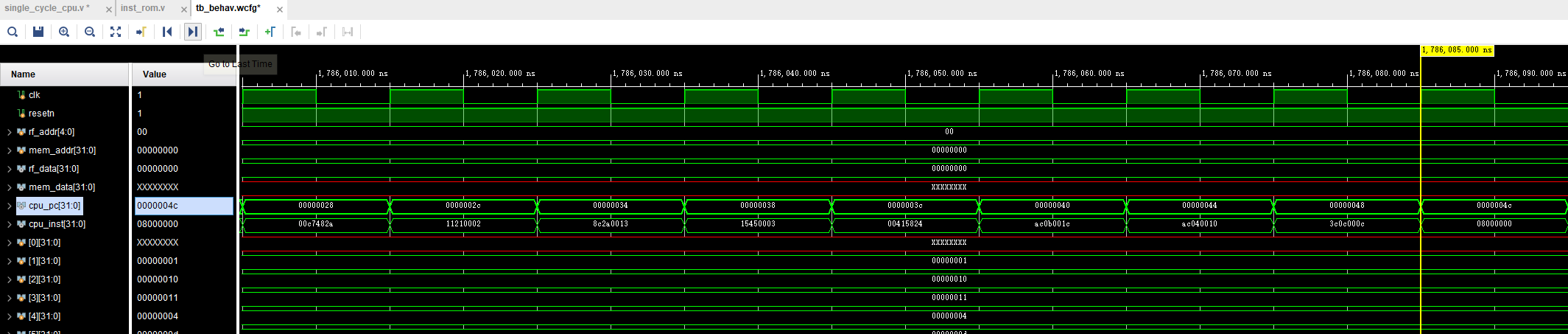
**48H**





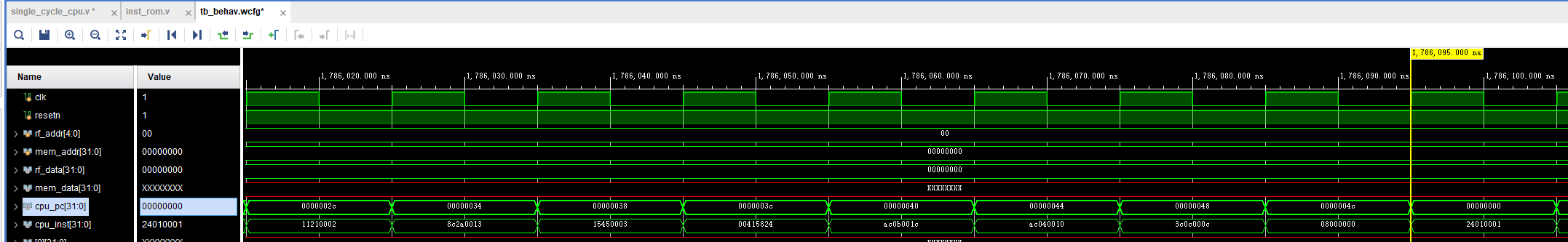
指令编码为3C0C000CH,对应汇编指令为 lui $12,#12 ,指令功能为高位加载，将立即数12加载进12号寄存器的高位，地位填充0，故12号寄存器结果应为000C 0000H，观察仿真图像，结果正确

**4CH**





指令编码为0800 0000H，对应汇编指令为 j 00H, 指令结果为将00H送入PC，无条件跳转到00H处。



点击下一个时钟周期，发现来到了00H处，实验结果正确。

**实验总结**

通过本次实验，我更加清楚的了解了单周期cpu运作的原理。

首先对指令的来源，有三种方式，分别是pc+4,分支和跳转，要予以区分。顺序执行pc = pc+4，分支要判断是beq还是bne，跳转为PC=PC+offset<<2，无条件跳转为PC={PC[31:28],target<<2}

根据指令不同位置取出不同的数据，操作码是31到26，rs是25到21，rt是20到16，rd是15到11，特殊域10到6，功能码5到0，立即数15到0，地址偏移量15到0，j指令目标地址25到0.

要根据op和funct等信息确定指令是执行的哪种操作，并根据操作类型生成alu控制信号。对立即数进行扩展，进行alu运算。

访存阶段要确定内存写使能，非resetn状态下有效。内存写的地址为alu运算结果，数据为rt寄存器值。写回阶段要判断是写入rt还是rd，是写入alu运算结果还是load结果。

设计单周期的cpu不仅要对指令熟悉掌握，更要清清楚楚的理解好单周期cpu的运作流程，以及每种操作对应的控制信号序列。