实验 11-多周期处理器综合性开放实验

# 一、 实验准备

前面我们已经完成的了 MIPS ISA 实验，并学会了使用 MARS 仿真器来编译代码，同时也能通过 MARS 仿真器将汇编代码编译成二进制机器代码。同时，在前面的实验中，我已经搭建了多周期处理器系统，并通过了 modelsim 的仿真测试。这节课我们将在 FPGA 板上实现多周期处理器，并运行一段实际的程序，并将最后的结果显示在七段数码管上；

在进行本次实验前，你需要具备以下基础能力：

1. 熟悉 MARS 仿真器，理解汇编代码转换成二进制机器代码的过程；
2. 理解存储器原理，将编译二进制代码存进存储器；
3. 理解多周期处理器的运行流程，实现多周期处理器的上板运行；

二、 实验目的

1. 掌握多周期CPU 数据通路及其设计方法；
2. 理解 MIPS 汇编代码转换成二进制机器代码的过程；
3. 掌握多周期CPU 在 FPGA 上的物理实现方法；
4. 掌握测试多周期 CPU 上板测试的方法；

三、 实验设备

PC 机一台，Basys3 开发板，Xilinx Vivado 开发套件。

# 四、 实验任务

## 实验要求

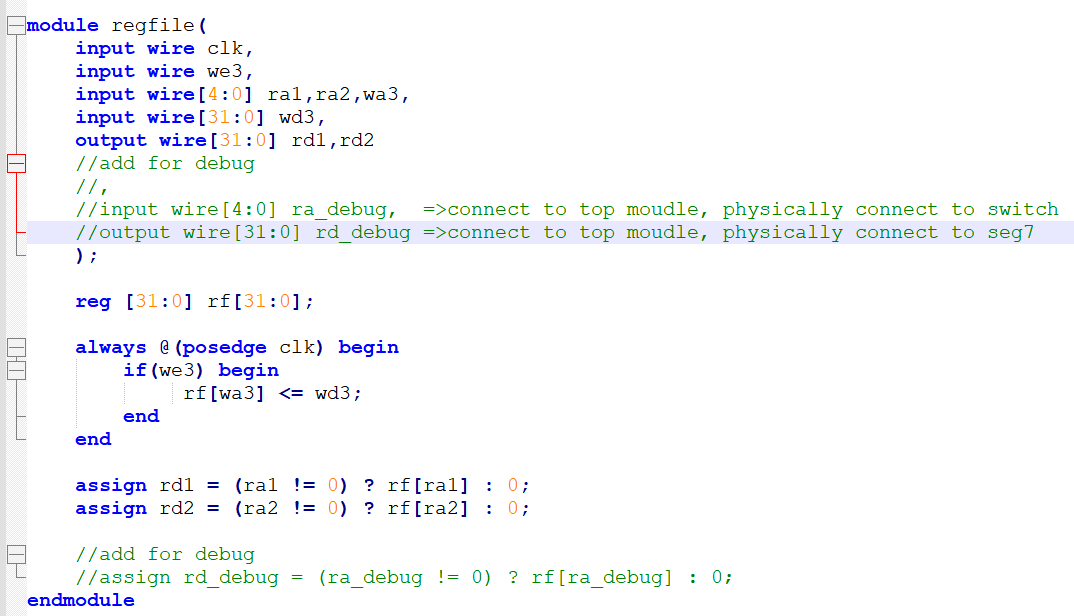
1. 自己编写一个特定功能的汇编代码（比如，等差数列求和，冒泡排序，求平均值），代码要求具有实际意义，实现 C 语言代码，并将代码按照 ISA 规则翻译成机器代码；
2. 程序的输入输出变量需要制定特定的寄存器或者物理地址，举个例子，如果要实现等差数列求和的代码，计算 M=1+2+3+…+N, 这里的 N 是一个输入变量，M 是输出变量；需要在代码中指明输入 N 是放在那个寄存器中（如果是数组，就需要指明数组的首地址）；M需要指明放在哪一个寄存器里面，可以通过七段数码管显示出来，验证计算结果的正确性；
3. 将机器代码转成 COE 格式，并存入到存储器的指令地址空间（注意指令和数据的空间的分割，指令空间要足够大）；
4. 改造仿真多周期实验代码，使得能运行在 FPGA 开发板上；具体改造见后面的实验步骤；
5. 给出实际程序在所搭建的多周期处理器上的实际运行结果，并测试不同测试输入条件下，计算结果的正确性，并完成实验报告；

## 实验实现过程中会遇到的问题

**问题 1：汇编代码到机器代码的转换**

这个问题可以有 2 种方式实现。第一种方法是对照 MIPS ISA 指令编码规则（参照指令及其对应机器码pdf 文件），自己翻译；第二种方法是借助 MARS 仿真器，将你自己编写的代码在 MARS 进行编译调试，同时编译成二进制机器代码；推荐使用第二种方式；

## 问题 2：程序运行结果显示问题

最后的结果可以存到一个寄存器里面，然后通过实验 9 中的方法，改造寄存器堆模块，增加一路读口，然后把新增加的读口地址连接到拨码开关，把新增加的读口接到七段数码管上显示（类似于前面的寄存器堆实验里面的做法，同学们可以参考一下）；验证程序运行后，通过拨码开关选择不同的寄存器，通过七段数码管上显示查看每个寄存器的值是否符合预期。修改之后的代码片段：

## 问题 3：如何让 CPU 停止在某一个地方，可以显示 CPU 运行的结果

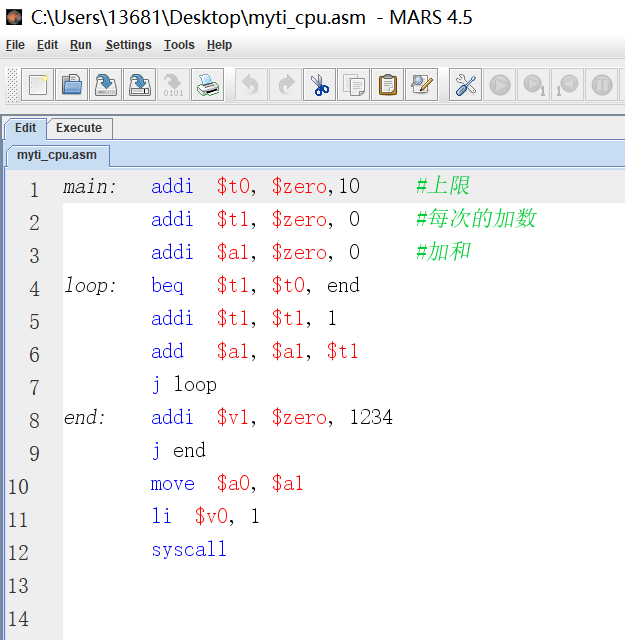
这个问题可以通过在程序的末尾（也就是输出结果的地方）加入 while(1), 可以让 CPU 停止在所要求的哪一行，然后可以通过拨码开关查看寄存器的值，查看运行结果。这种方法也可以用于，在线调试每一行代码的情况；注意，while(1)代码也要加入 MARS 仿真器里面生成相应的二进制机器代码；

# 五、 实验报告要求

1. 按照实验要求搭建可以在 FPGA 开发板上运行的多周期处理器系统；
2. 自己选定一个具有特定实际意义的程序，编写其汇编代码，转换成二进制机器代码，并生成相应的 COE 文件，并验证代码的正确性；
3. 将生成的COE 文件导入多周期处理器系统中，验证实际原型结果，展示其实验结果的正确性；
4. 在实验报告中，详细阐述上面 1-3 条实验设计过程，验证每个实验步骤的正确性，并记录验证过程；根据实际运行结果说明你设计的处理器为什么是正确的，能运行的，并分析实验结果。
5. 提交运行结果视频，并语言解释实验结果；

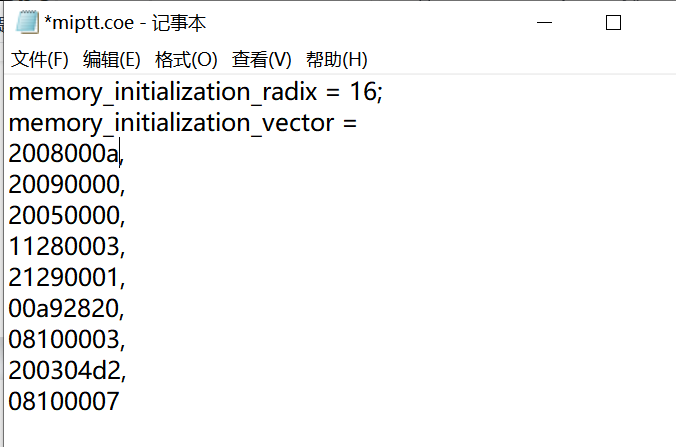
我选择写进去的程序是从1开始到指定自然数的加和，在上限的选择上有两个，一个是10，一个是12，并且两种情况都进行了上板，验证了准确性。

1. 上限为10的代码：



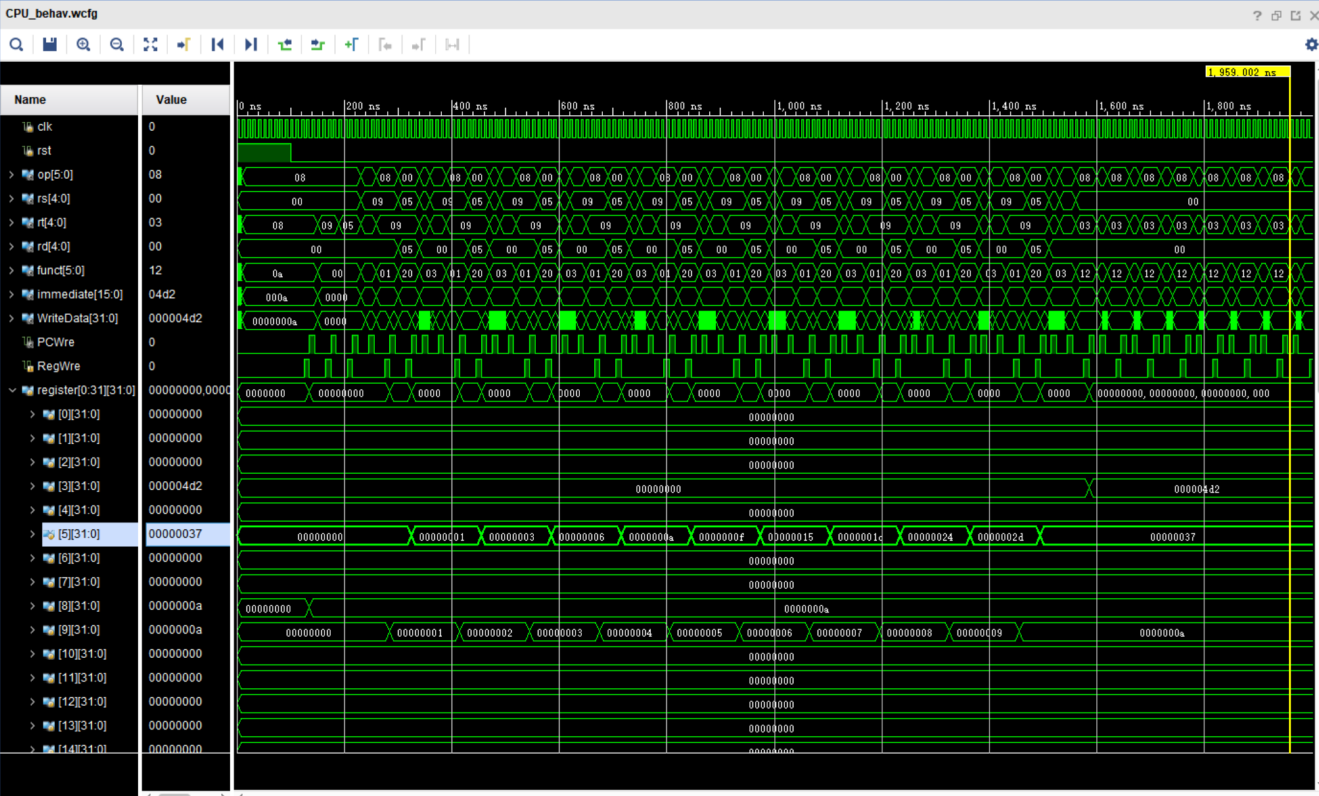
这是在mars上写的汇编代码，main模块进行一切寄存器的初始化赋值。Loop模块主要进行计算加和。End模块用1234来确保跳出loop，并设计了一个死循环来方便上板的寄存器查看。

将以上代码翻译成机器码



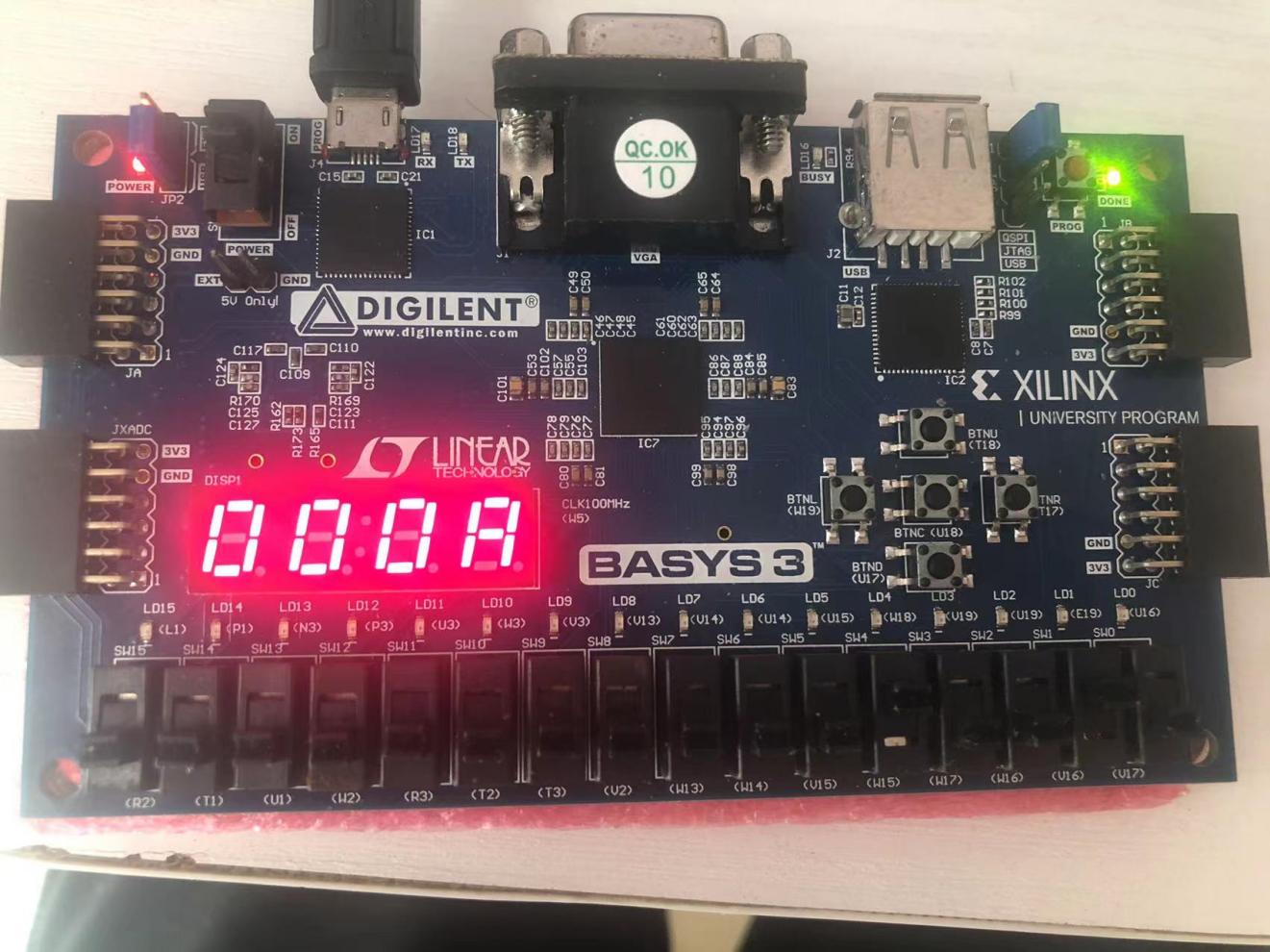
并将机器码导入到多周期cpu里的存储器当中

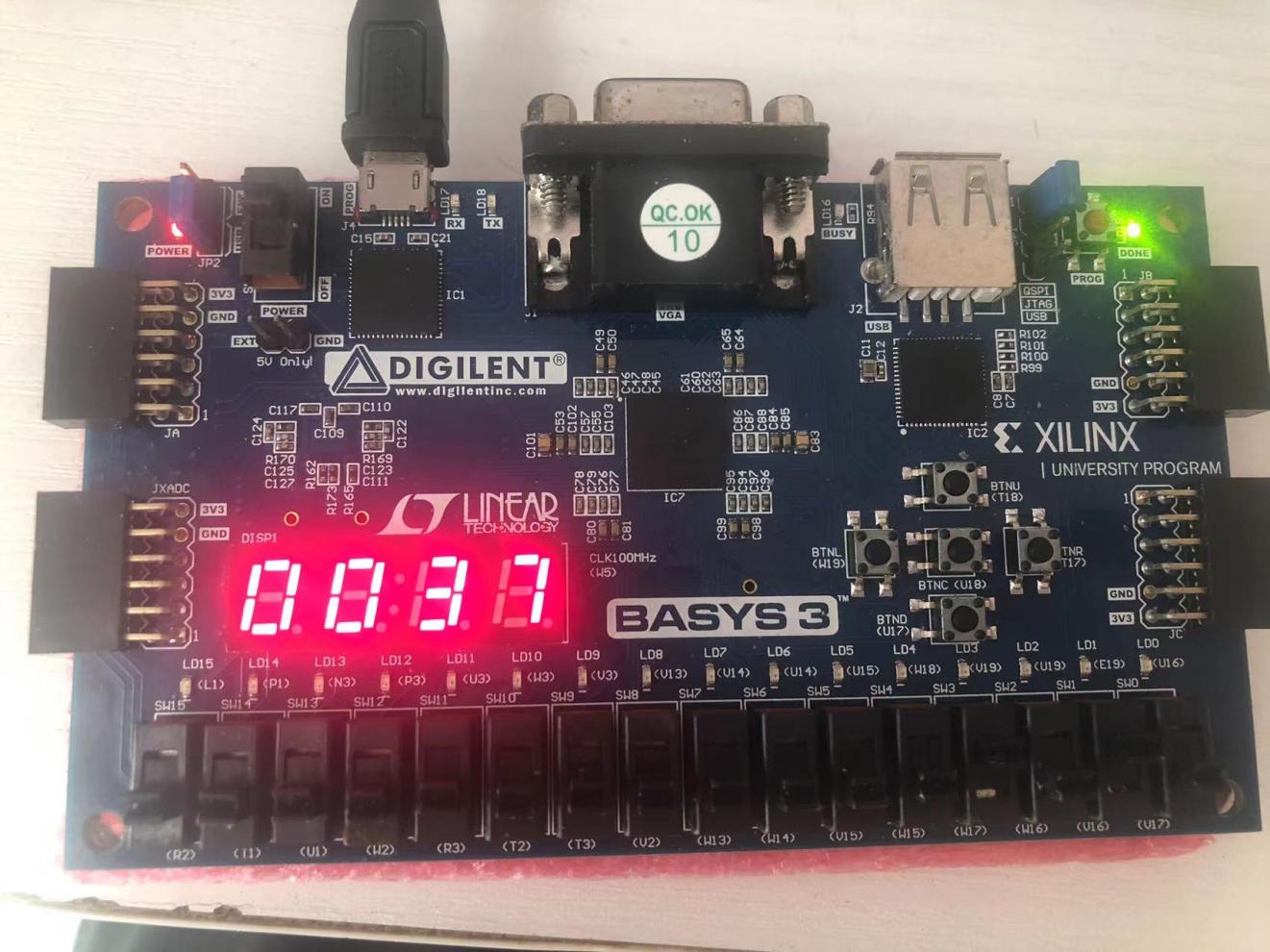
在vivado里进行仿真

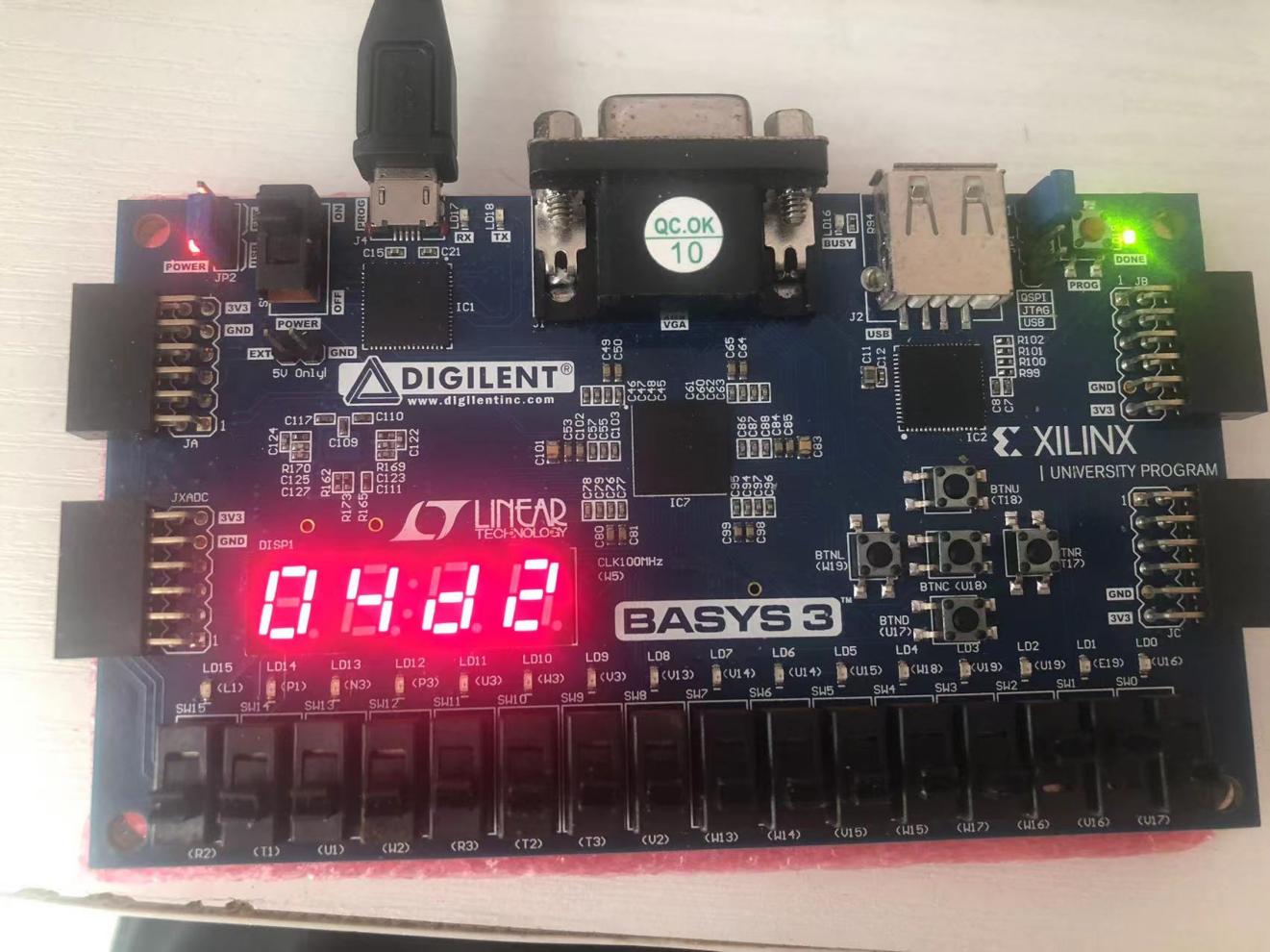


可以看到五号寄存器里的值为16进制的37，确实是十进制的55。九号寄存器表示每次的加数，也确实加到了10。三号寄存器里也存进了1234（十六进制4d2）.

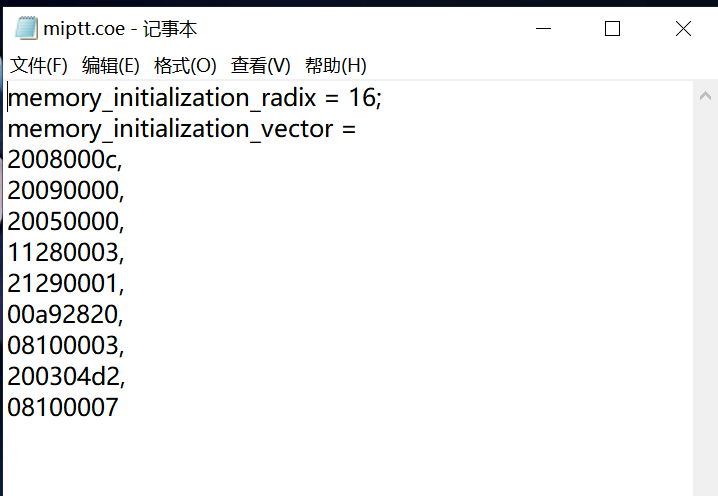
以下是上板照片



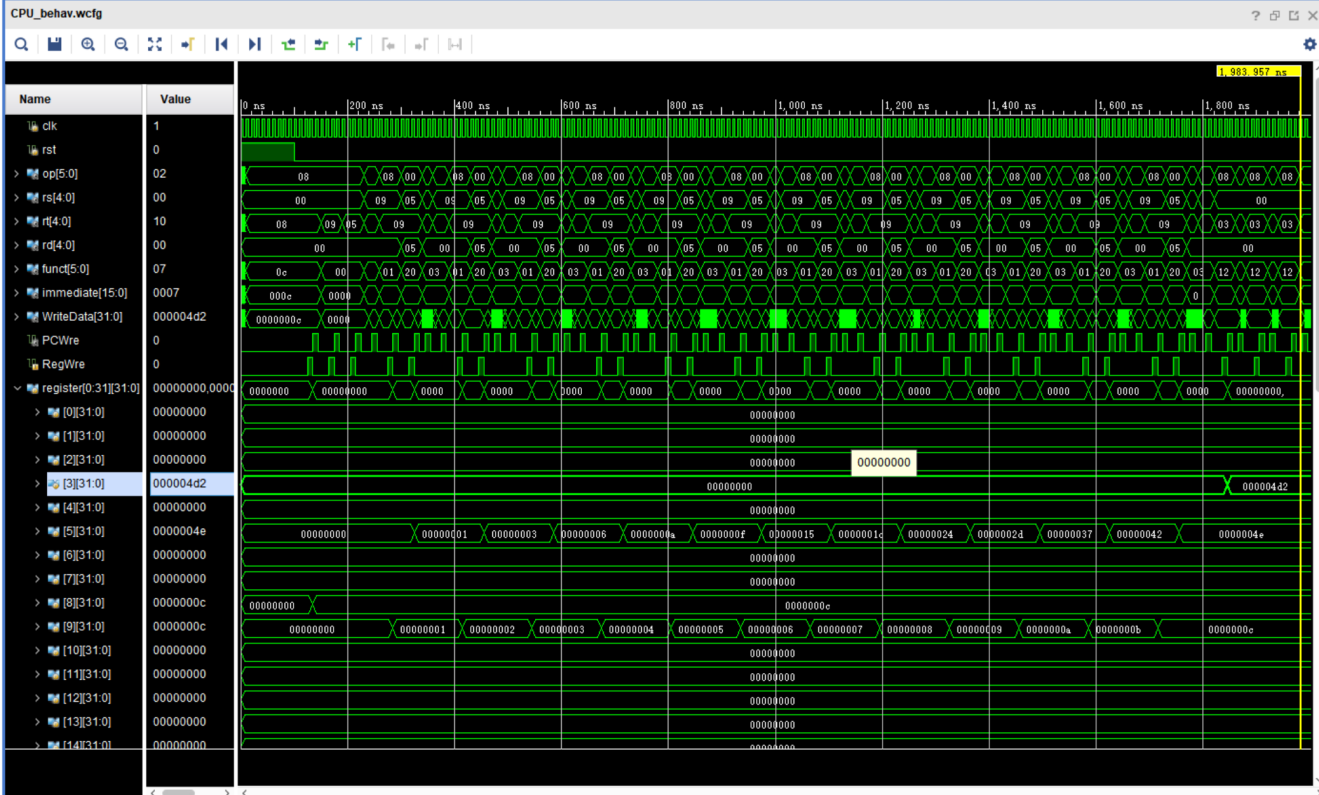
****

****

接下来是上限为12的程序

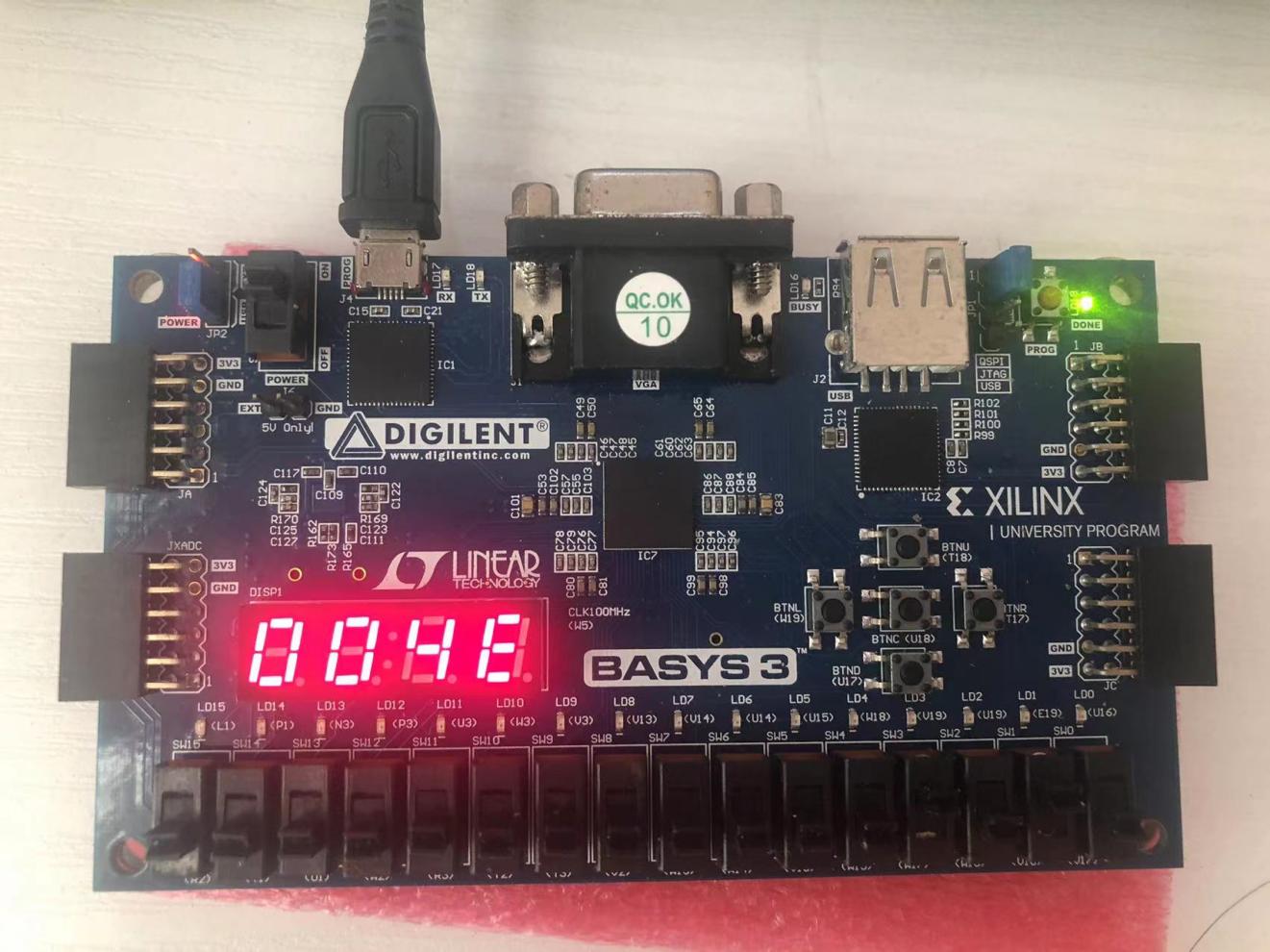


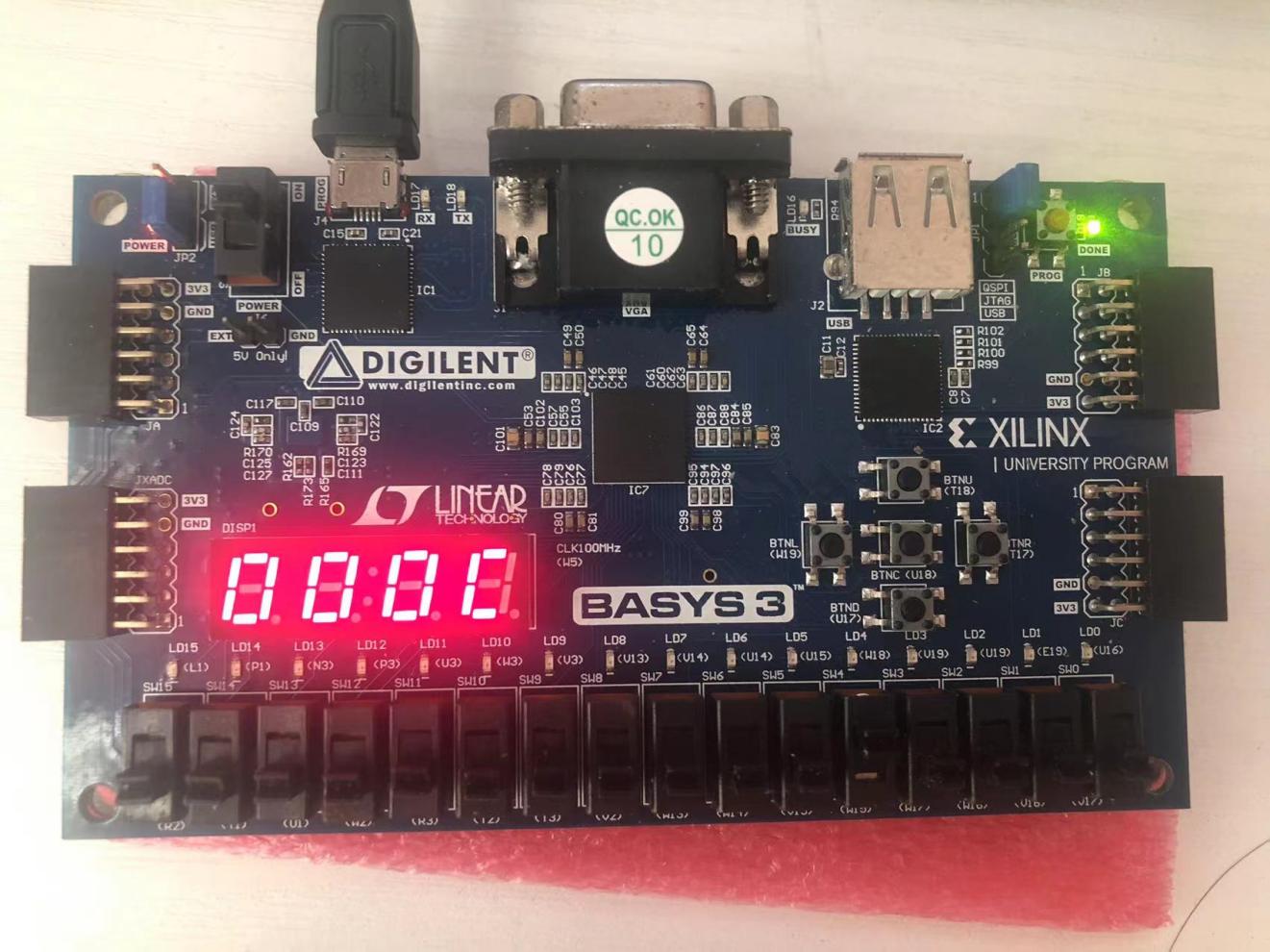
导入到存储器，进行仿真

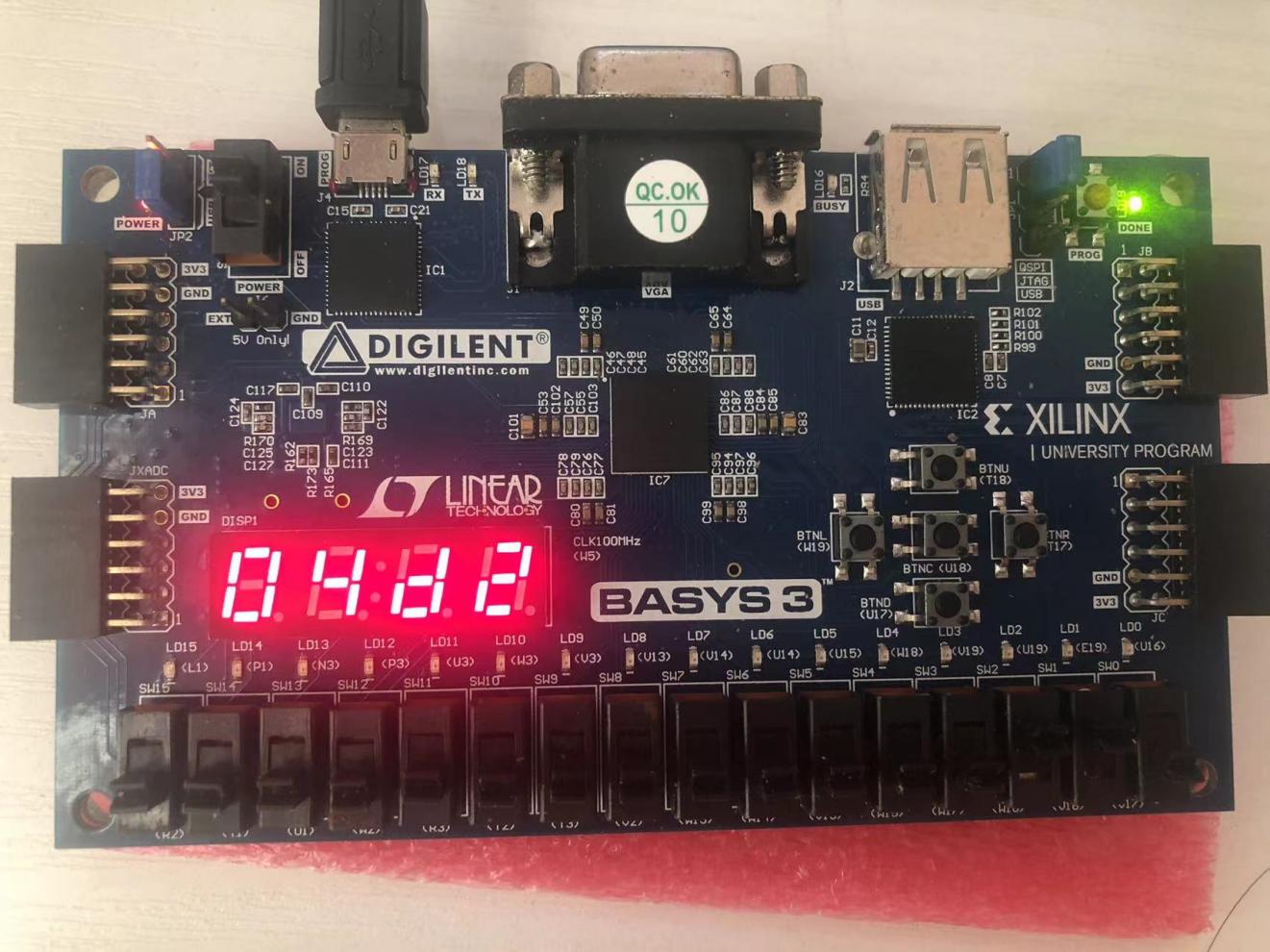


可以看到结果为4e，换算成十进制为78

以下是上版图片







具体演示过程可以查看验收视频

实验总结：

在上板的过程中也遇到了一些小问题，比如没有写死循环会导致板内在某一时刻计算停止且无法正常显示当时的加和，但是在这种情况下系统的仿真却并没有出现问题，所以实践和理论检验还是有不小差距的。