

计算机系统结构课程实验

总结报告

实验题目：动态流水线设计与性能定量分析

学号：2152809

姓名：曾崇然

指导教师：陆有军老师

日期：2023-12-30

1. 实验环境部署与硬件配置说明

1. 实验环境部署：操作系统：windows11，开发

工具：vivado，仿真工具：vivado自带的仿真工具

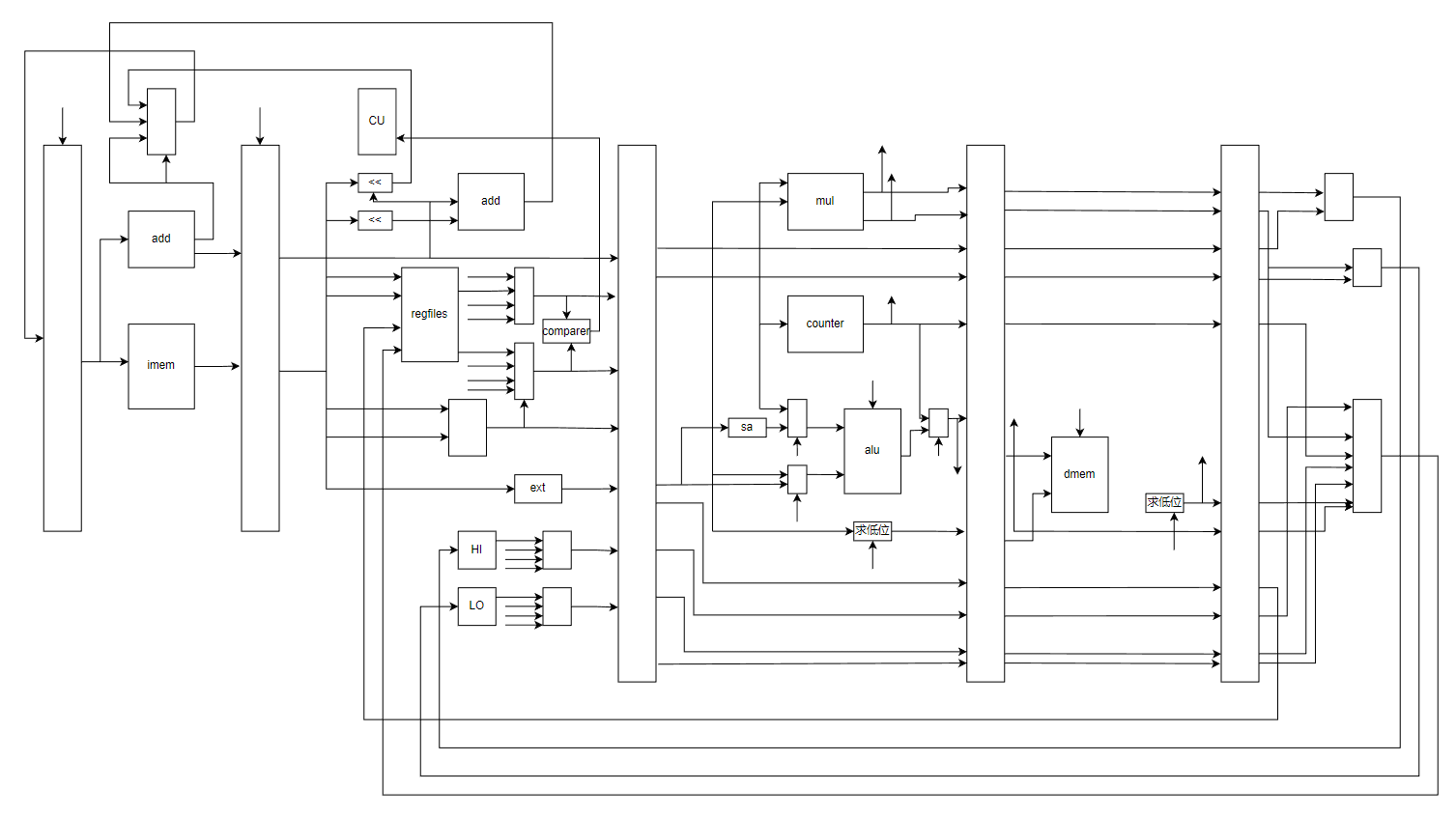
2. 硬件配置说明：Xilinx FPGA器件：Nexys DDR

开发板

1. 实验的总体结构

实验要求完成至少31条MIPS指令的动态流水线CPU设计，并支持中断。在CPU运行验证程序的过程中，由按键或拨动开关产生一个暂停的中断，再次按键或拨动开关结束中断，继续运行后续的运算，并在数码管上动态显示运算值。

1. 动态流水线的总体结构



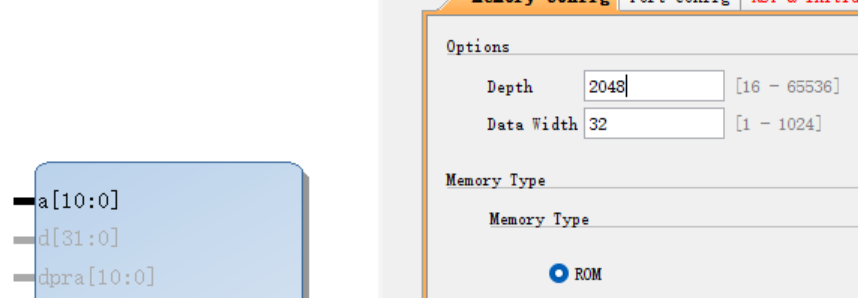
2、指令选取

共选取32条MIPS指令，具体如下：

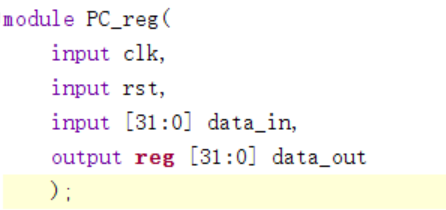
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 助记符 | 指令格式 | | | | | |
| Bit | 31-26 | 25-21 | 20-16 | 15-11 | 10-6 | 5-0 |
| add | 000000 | rs | rt | rd | 00000 | 100000 |
| addu | 000000 | rs | rt | rd | 00000 | 100001 |
| sub | 000000 | rs | rt | rd | 00000 | 100010 |
| subu | 000000 | rs | rt | rd | 00000 | 100011 |
| and | 000000 | rs | rt | rd | 00000 | 100100 |
| or | 000000 | rs | rt | rd | 00000 | 100101 |
| xor | 000000 | rs | rt | rd | 00000 | 100110 |
| nor | 000000 | rs | rt | rd | 00000 | 100111 |
| slt | 000000 | rs | rt | rd | 00000 | 101010 |
| sltu | 000000 | rs | rt | rd | 00000 | 101011 |
| sll | 000000 | 00000 | rt | rd | shamt | 000000 |
| srl | 000000 | 00000 | rt | rd | shamt | 000010 |
| sra | 000000 | 00000 | rt | rd | shamt | 000011 |
| sllv | 000000 | rs | rt | rd | 00000 | 000100 |
| srlv | 000000 | rs | rt | rd | 00000 | 000110 |
| srav | 000000 | rs | rt | rd | 00000 | 000111 |
| jr | 000000 | rs | 00000 | 00000 | 00000 | 001000 |
| addi | 001000 | rs | rt | immediate | | |
| addiu | 001001 | rs | rt | immediate | | |
| andi | 001100 | rs | rt | immediate | | |
| ori | 001101 | rs | rt | immediate | | |
| xori | 001110 | rs | rt | immediate | | |
| lui | 001111 | rs | rt | immediate | | |
| lw | 100011 | rs | rt | immediate | | |
| sw | 101011 | rs | rt | immediate | | |
| beq | 000100 | rs | rt | immediate | | |
| bne | 000101 | rs | rt | immediate | | |
| slti | 001010 | rs | rt | immediate | | |
| sltiu | 001011 | rs | rt | immediate | | |
| j | 000010 | address | | | | |
| jal | 000011 | address | | | | |
| mul | 011100 | rs | rt | rd | 00000 | 000010 |

1. 总体架构部件的解释说明
   1. 动态流水线总体结构部件的解释说明

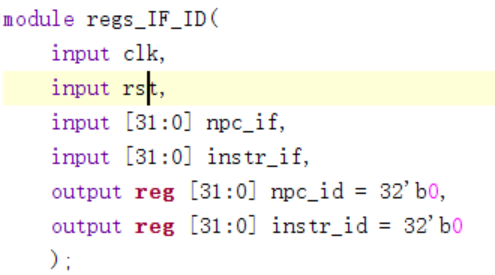
a. 指令存储器imem：用于存储将要执行的指令



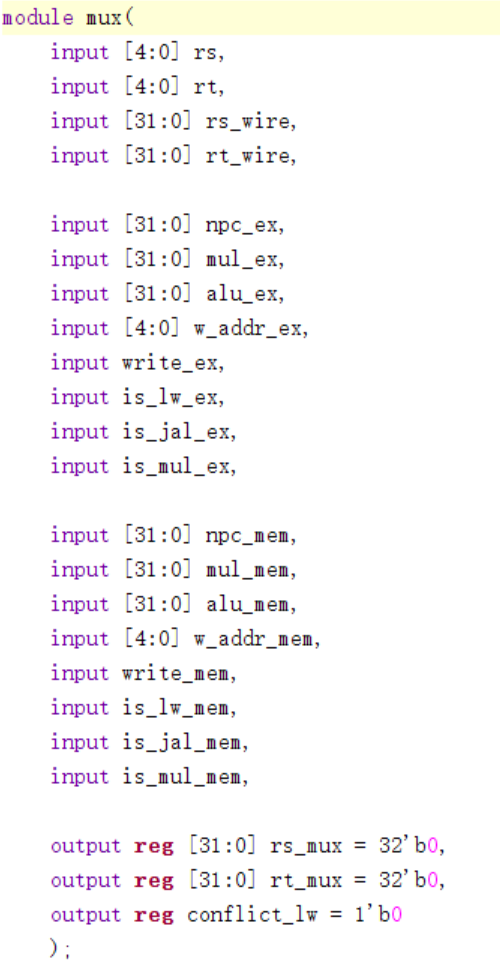
b. PC寄存器PC\_reg

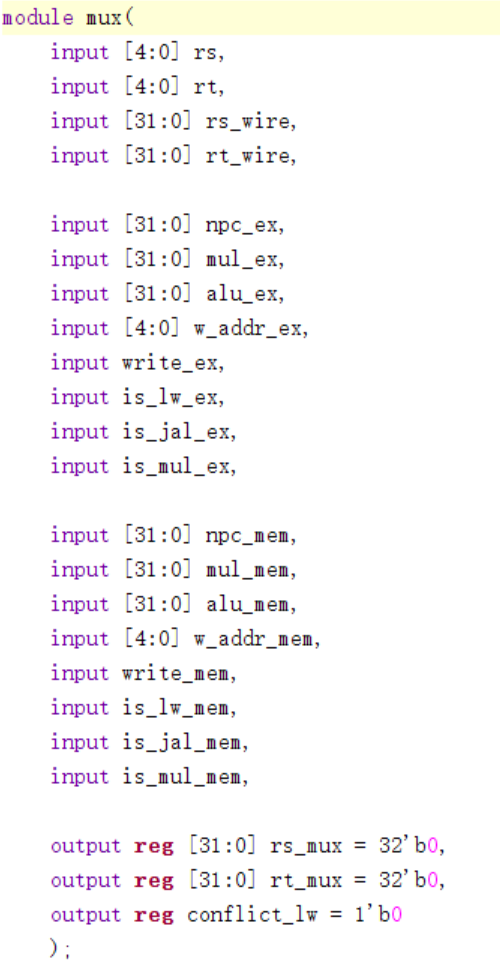


c. IF和ID之间的流水寄存器regs\_IF\_ID：用于将IF输出结果传递给ID

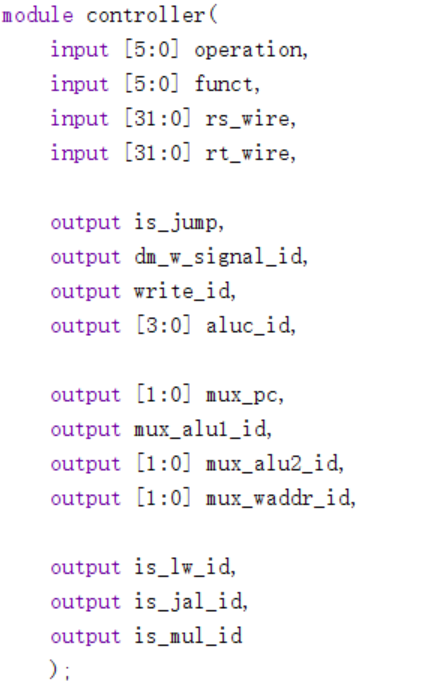


d. 专用路径模块mux：将EX阶段或者MEM阶段的输出结果定向前推至ID阶段的多路选择器中

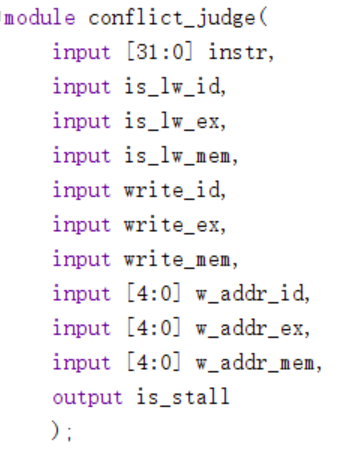




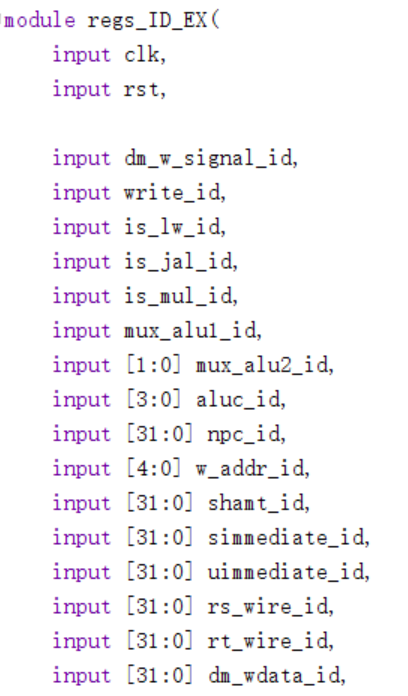
e. ID阶段的控制模块controller：用于进行指令的解码和产生控制信号

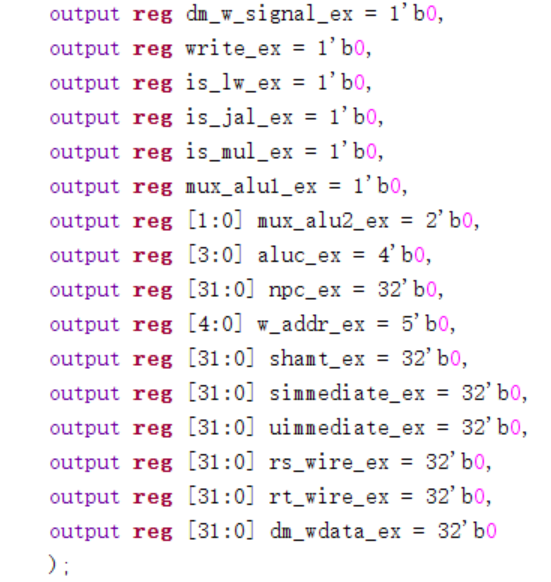


f. 冲突检测模块conflict\_judge：用于判断指令是否存在冲突，若存在，则暂停流水

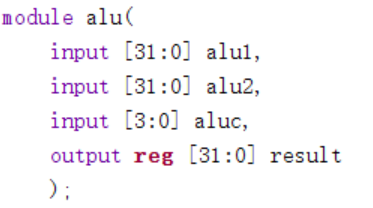


g. ID和EX阶段的流水寄存器regs\_ID\_EX：用于将ID阶段的输出传递给EX

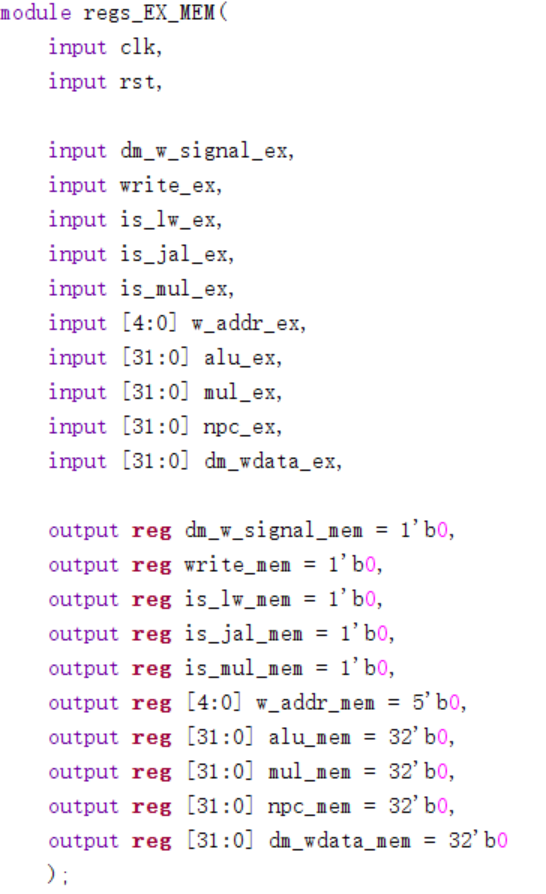




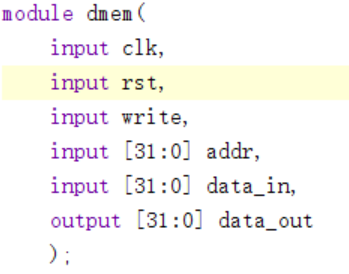
h. 运算模块alu：



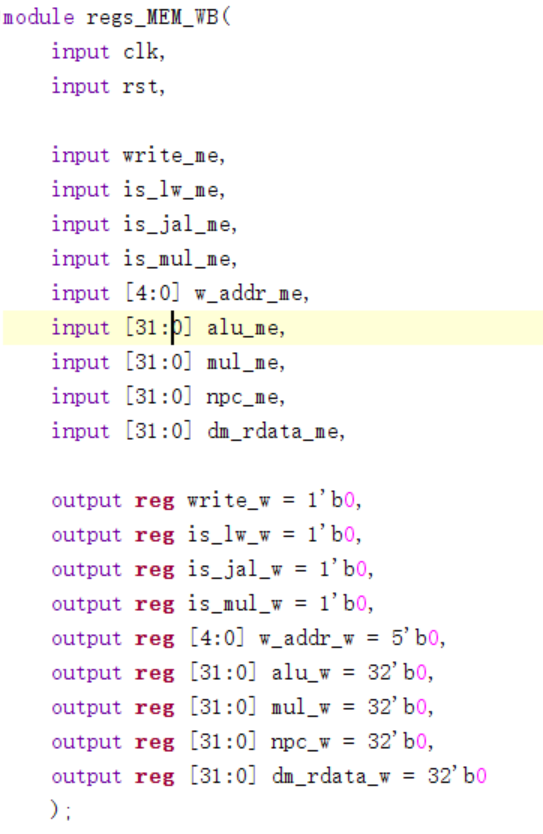
i. EX阶段和MEM阶段的流水寄存器regs\_EX\_MEM：用于将EX阶段的输出传递给MEM阶段



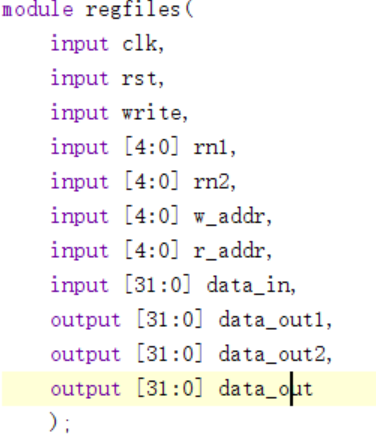
j. 数据存储模块dmem：



k. MEM阶段和WB阶段的流水寄存器regs\_MEM\_WB，用于将MEM阶段的输出传递给WB阶段



l. 寄存器堆模块regfiles：



* 1. 整体结构解释

a. 总体采用哈佛结构，指令和数据分开存储，通过数据总线和指令总线和CPU进行数据交换。

b. 一条指令的执行分为IF、ID、EX、MEM、WB五个阶段，分别为取址、取值、计算、访存和写回，根据指令类型的不同在不同的阶段完成不同的操作。

c. 在相邻的两个阶段设置流水寄存器来实现指令的流水执行

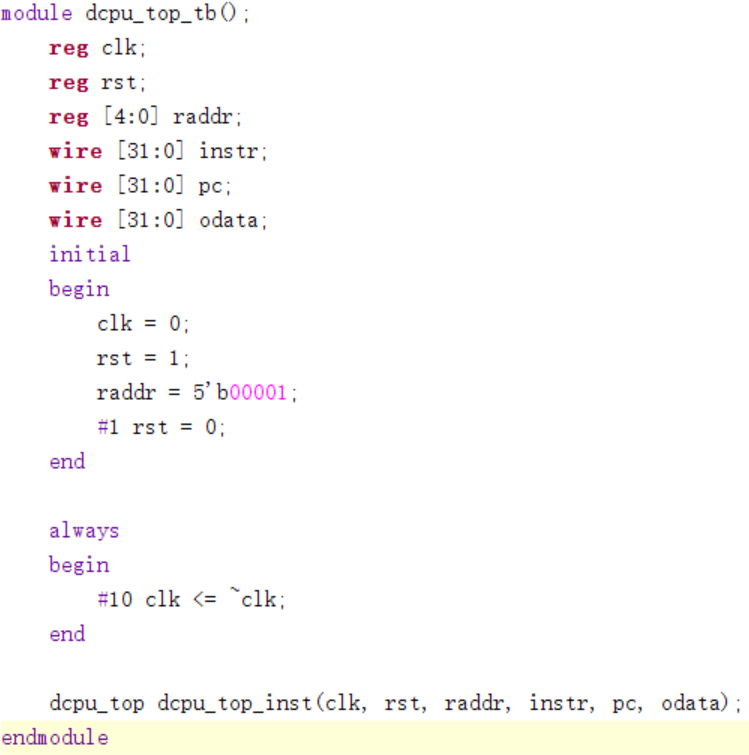
d. 设置专用路径，将之前周期EX阶段或者MEM阶段输出的结果定向前推至ID段向EX阶段输出源操作数时，使用控制信号进行多路进行选择。

e. 采用延迟操的思想解决控制相关的冲突

f. 增加冲突判断模块，当检测到冲突时使用暂停流水的方式来消除数据相关。

1. 实验仿真过程
   1. 动态流水线的仿真过程

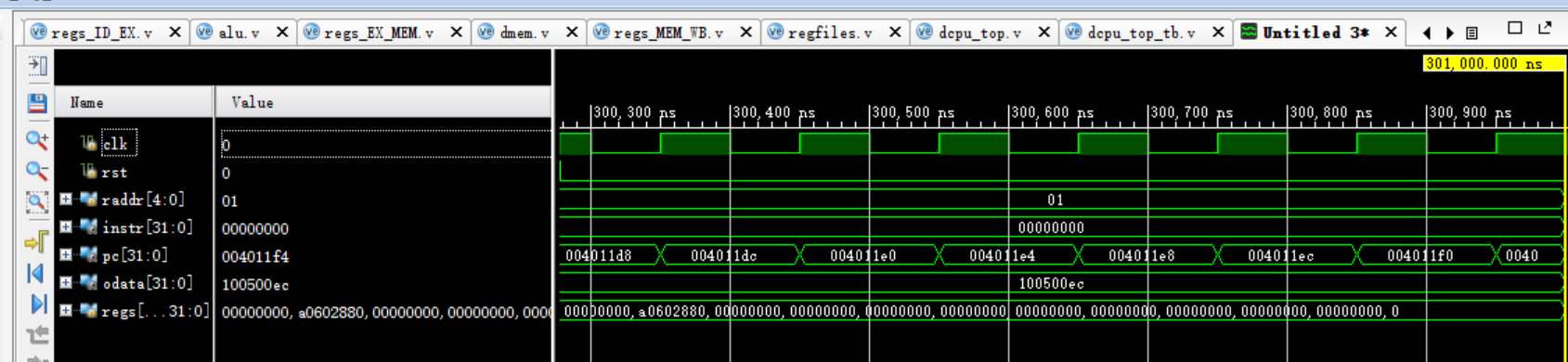
a. 测试文件的编写



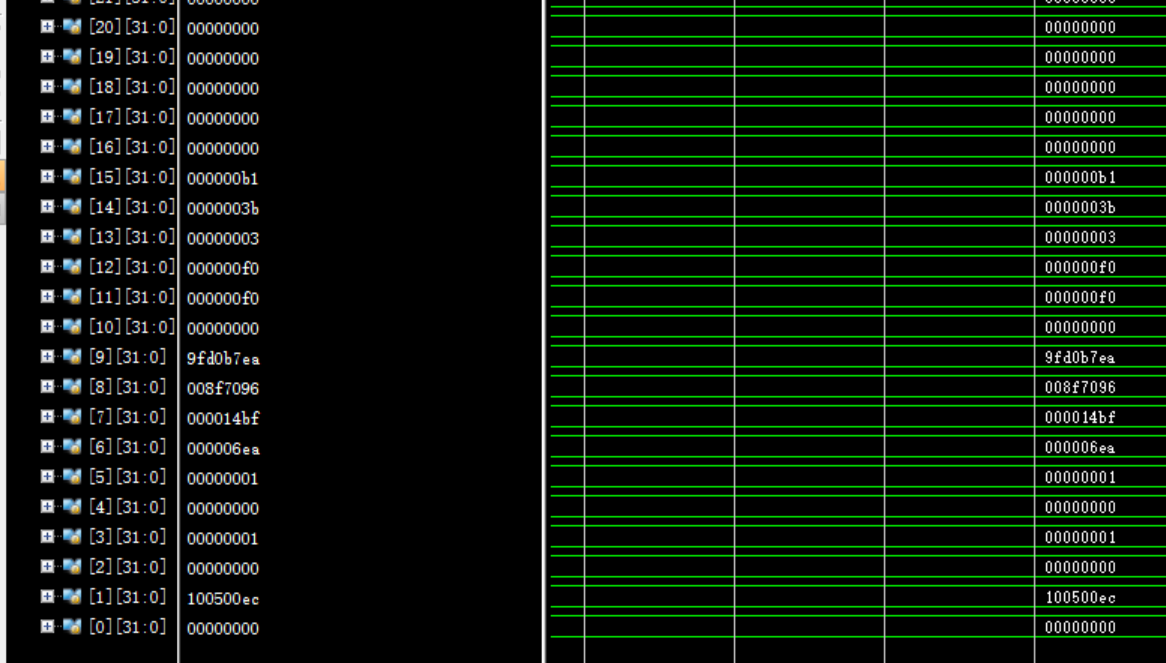
b. 进行仿真



1. 实验仿真的波形图及某时刻寄存器值的物理意义
   1. 动态流水线的波形图



* 1. 结束时刻寄存器值的物理意义



a. 寄存器6中存储了a[59]的值，十六进制的6ea，即1770

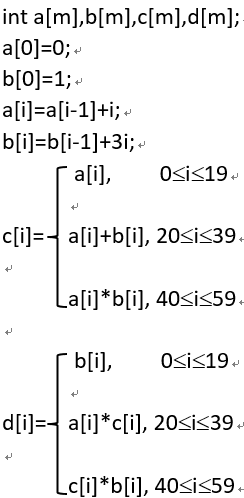
b. 寄存器7中存储了b[59]的值，十六进制的14bf，即5311

c. 寄存器8中存储了c[59]的值，十六进制的8f7096，即9400470

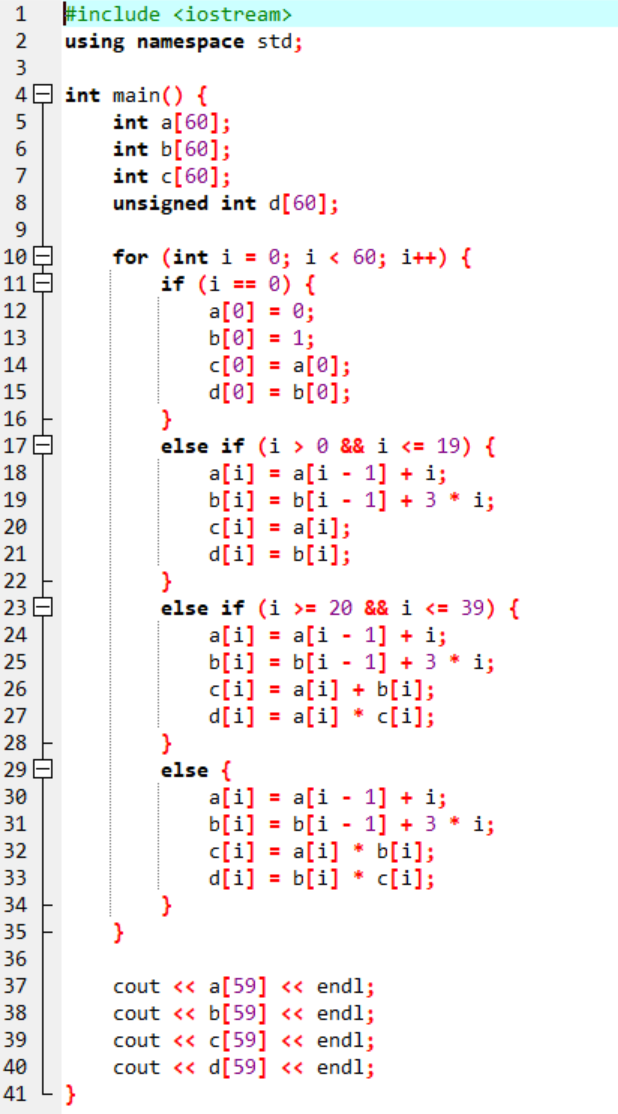
d. 寄存器9中存储了d[59]的值，十六进制的9fd0b7ea，即2681255914

e. 其他寄存器存储了一些计算的所需的中间值

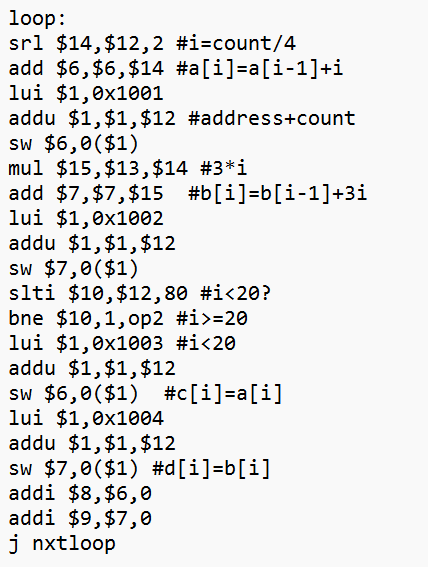
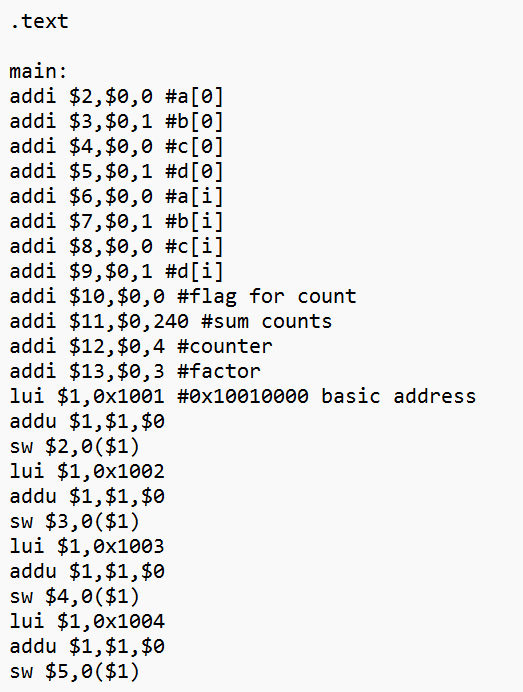
1. 实验验算数学模型及算法程序

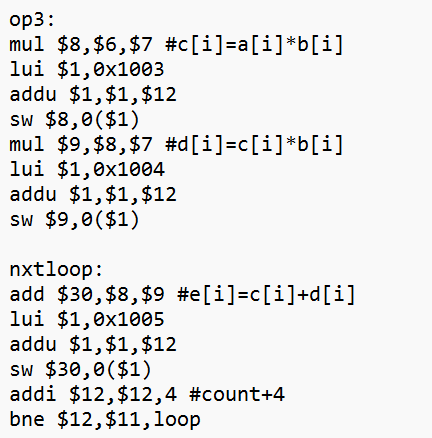
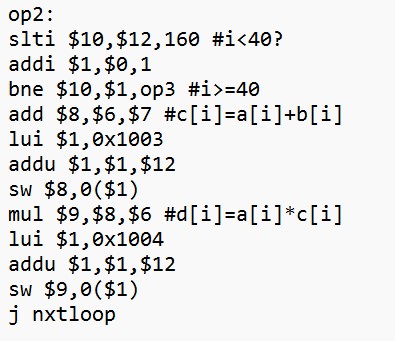


1. 编写C++程序如下：



2. 汇编程序如下：





3. coe文件：

20020000

20030001

20040000

20050001

20060000

20070001

20080000

20090001

200a0000

200b00f0

200c0004

200d0003

3c011001

00200821

ac220000

3c011002

00200821

ac230000

3c011003

00200821

ac240000

3c011004

00200821

ac250000

000c7082

00ce3020

3c011001

002c0821

ac260000

71ae7802

00ef3820

3c011002

002c0821

ac270000

298a0050

20010001

142a0009

3c011003

002c0821

ac260000

3c011004

002c0821

ac270000

20c80000

20e90000

08100042

298a00a0

20010001

15410009

00c74020

3c011003

002c0821

ac280000

71064802

3c011004

002c0821

ac290000

08100042

70c74002

3c011003

002c0821

ac280000

71074802

3c011004

002c0821

ac290000

0109f020

3c011005

002c0821

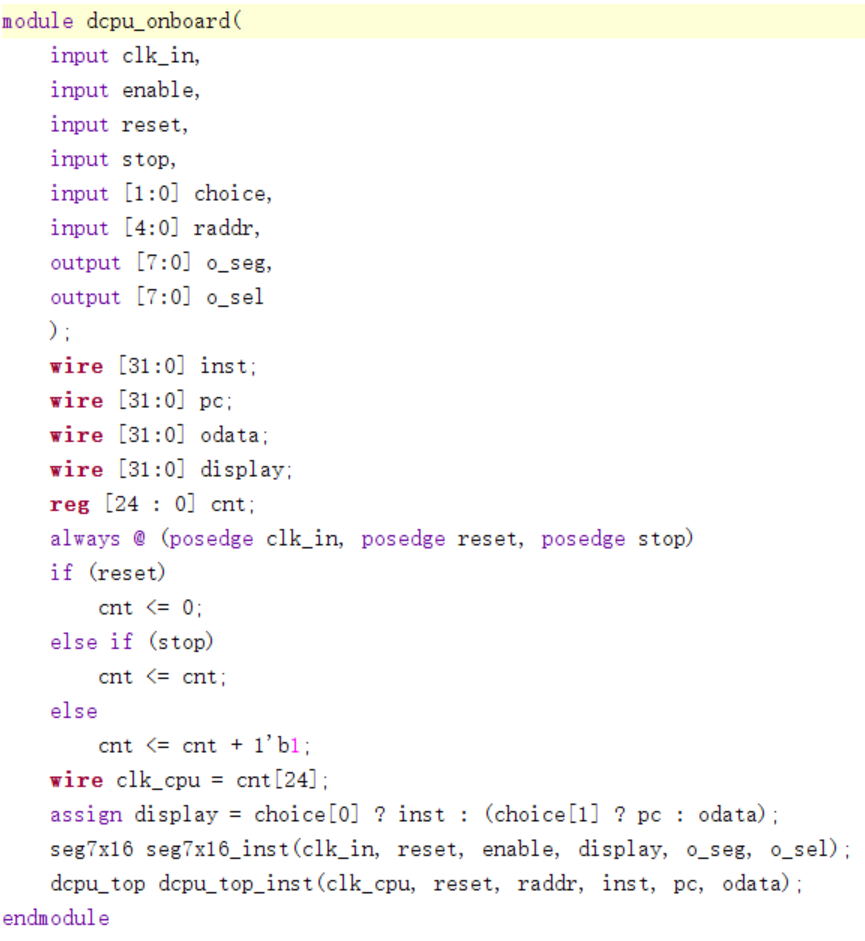
ac3e0000

218c0004

158bffd0

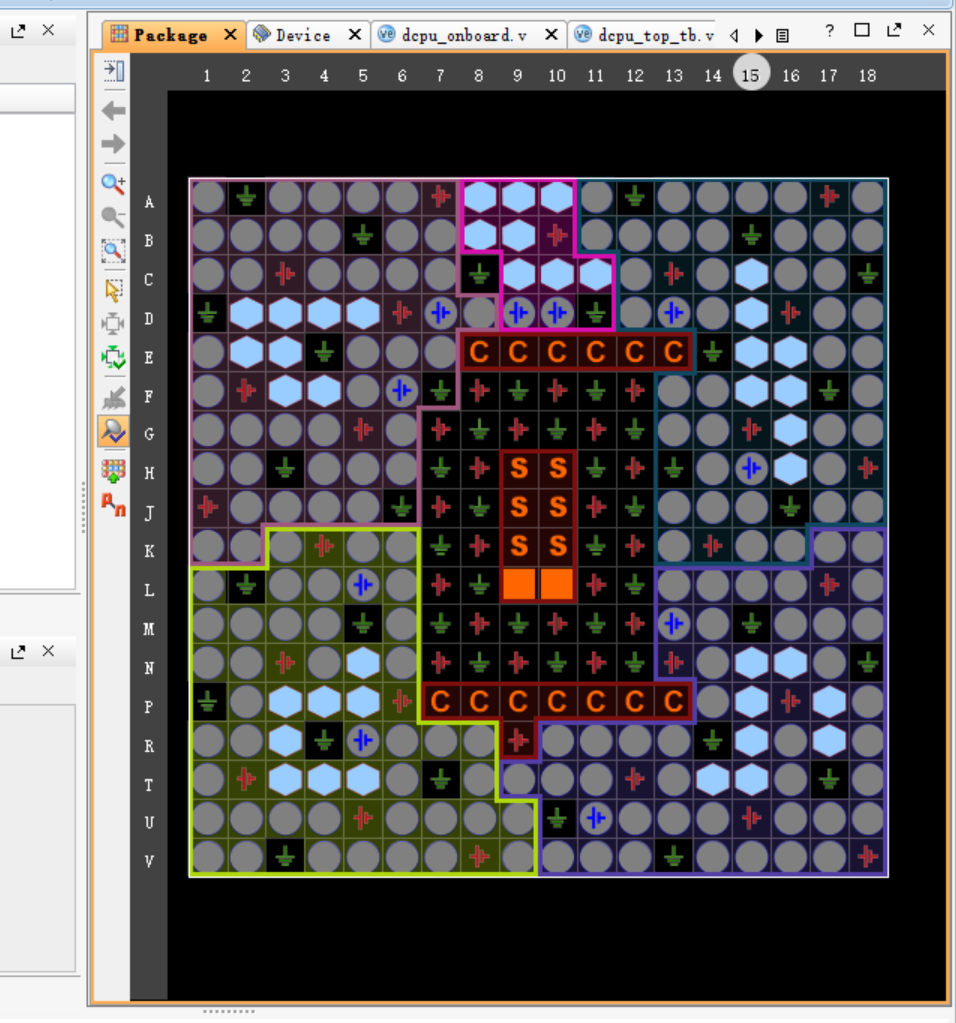
1. 实验验算程序下板测试过程与实现

1. 编写下板顶层文件（使用七段数码管显示结果）：

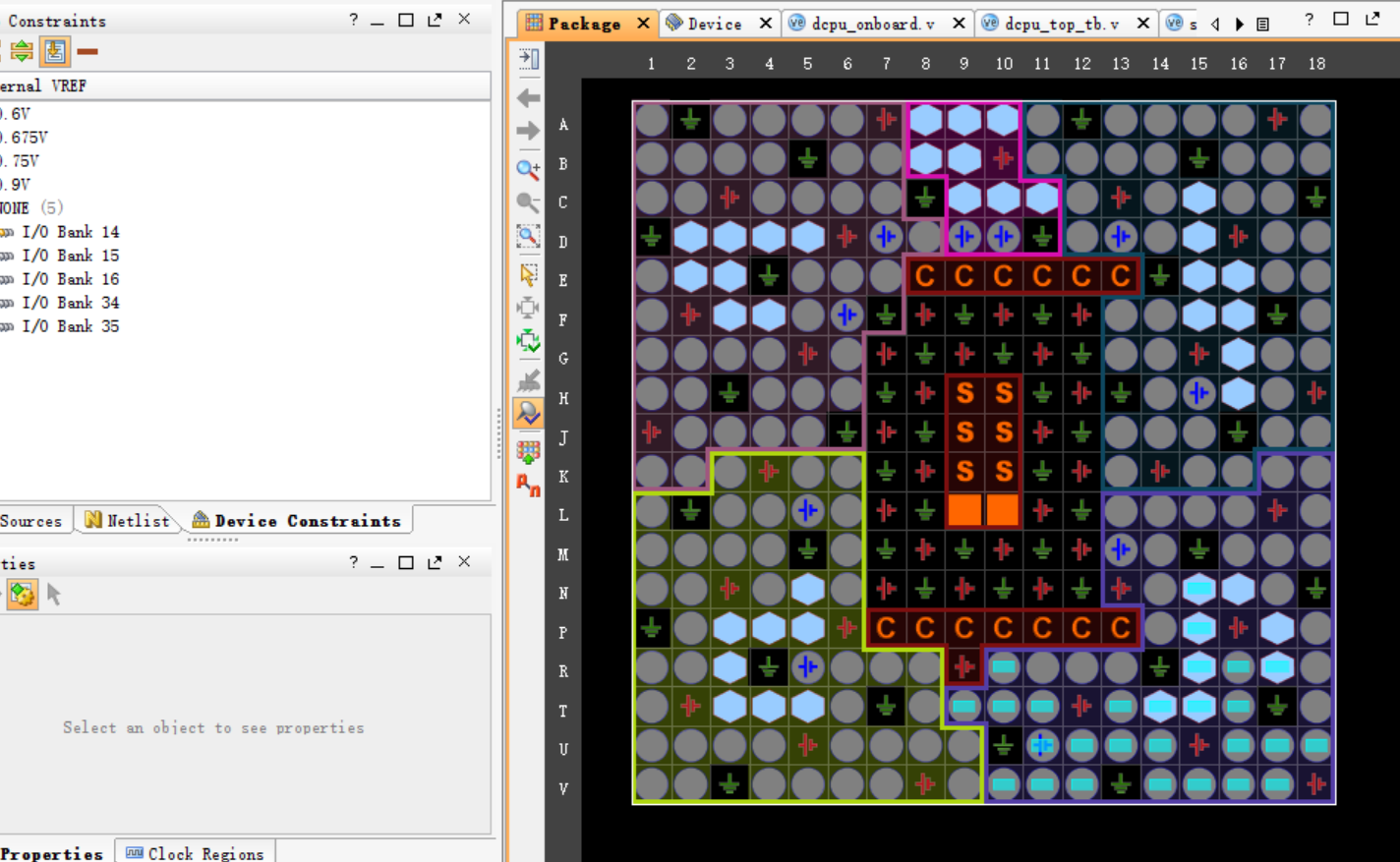


2. 综合、布线、下板

a. 综合



b. 布线

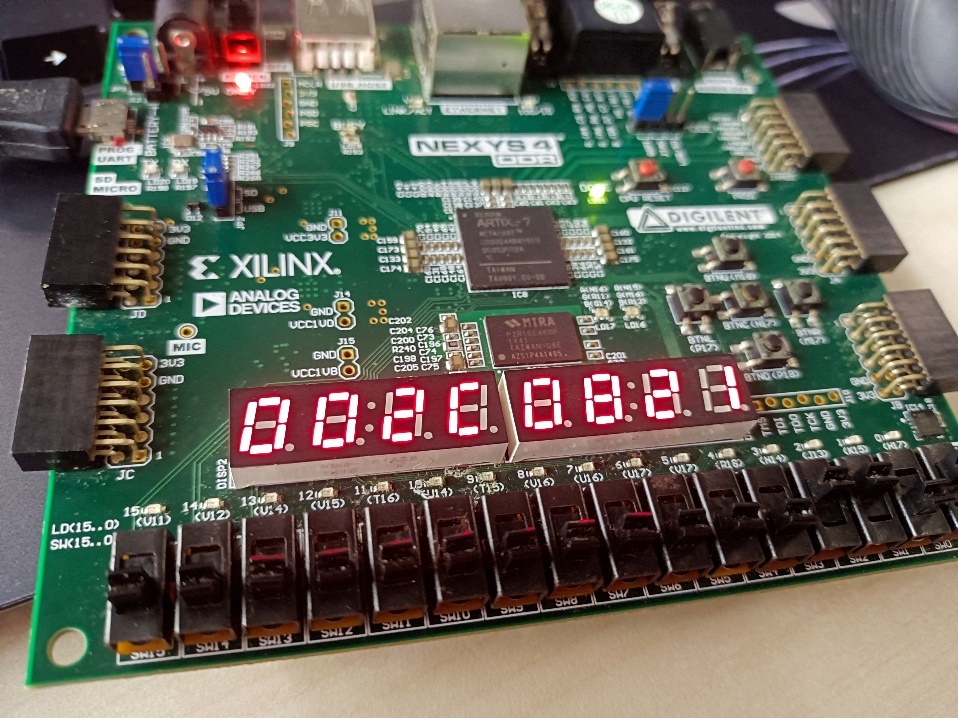


c. 下板

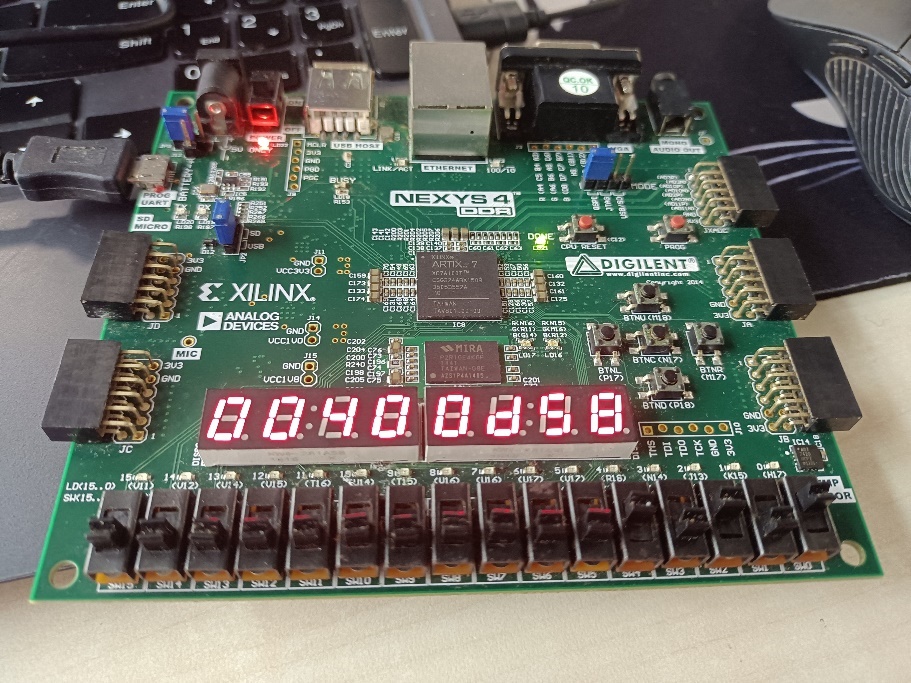


3. 观察现象（通过拨码开关来查看值）

a. 程序执行中间某时刻的指令值

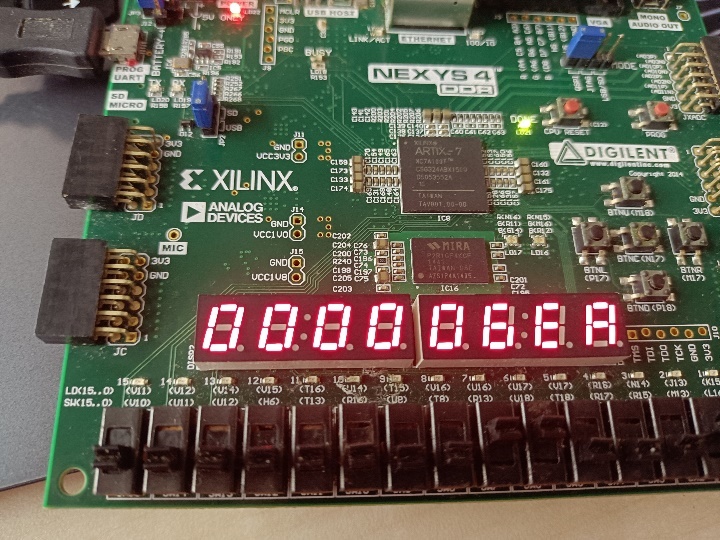


b. 结束时PC的值

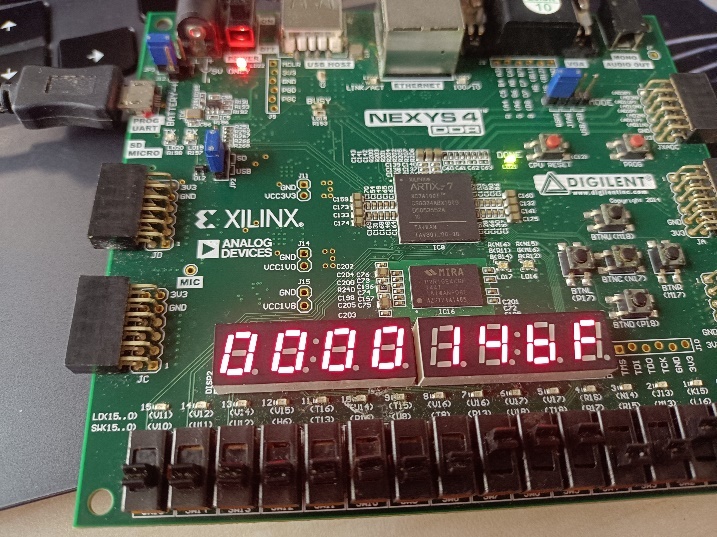


c. 结束时寄存器的值

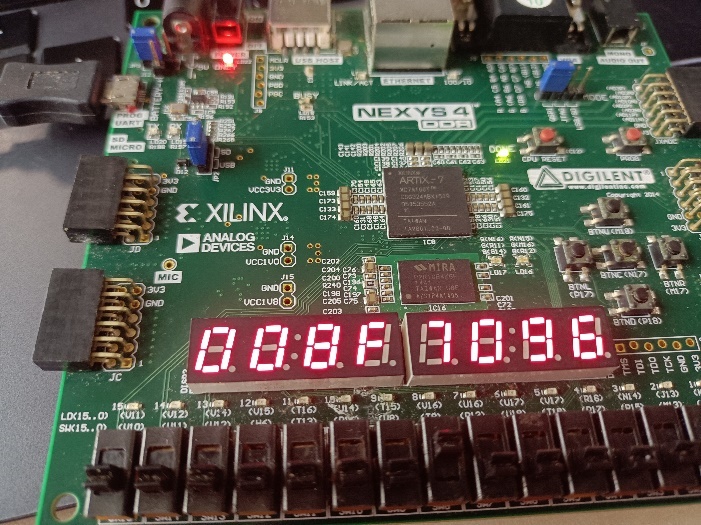
寄存器6（a[59]）：



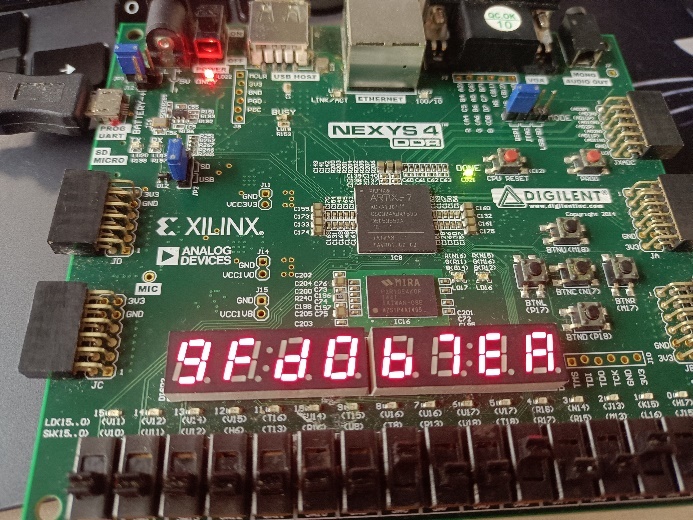
寄存器7（b[59]）：



寄存器8（c[59]）：

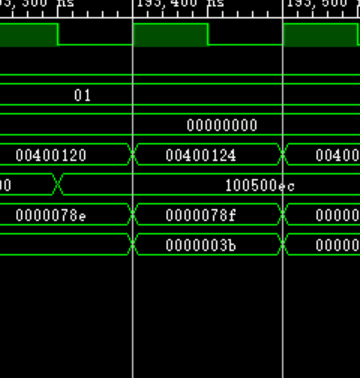


寄存器9（d[59]）：



1. 流水线的性能指标定性分析（包括：吞吐率、加速比、效率及相关与冲突分析）
   1. 动态流水线的性能指标定性分析

a. 吞吐率



观察波形，运行完程序一共执行了78f（16进制）条指令，空指令的数量为3b（16进制），所以吞吐率为(1935-59)/1939个时钟周期，结果为0.968/时钟周期

b. 加速比

非流水线所使用的时钟周期数量：(1935-59)\*5

流水线所使用的时钟周期数量：(1935+5-1)

所以加速比为4.838

c. 效率

总时空区：1939\*5

有效的时空区：1876\*5

所以效率为：0.968

2、冲突分析

在本次动态流水线CPU的设计中，相对于静态流水线的CPU，添加了专用路径，延迟槽等机制，解决了部分的先写后读冲突和控制冲突，大大减少了流水线CPU流水停止的概率，提高了流水线CPU的执行效率。

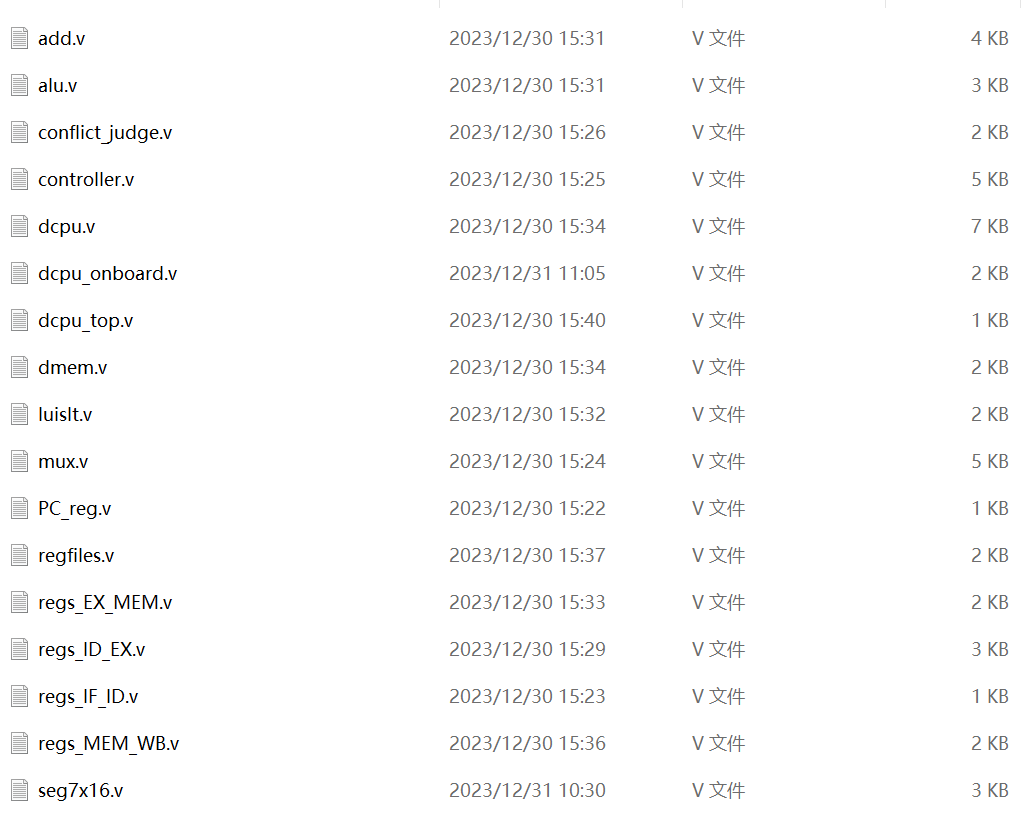
1. 总结与体会

本次实验设计了32条指令的动态流水线，并编写

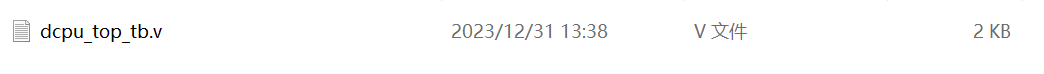
了相关的测试程序进行仿真和下板测试，分析了动态流水线CPU的相关性能，掌握了解决指令冲突的几种方法对于CPU性能的显著提升效果。对提升CPU性能的具体方法有了更加深入的了解。

1. 附件（所有程序）
   1. 动态流水线的设计程序

.v文件：



测试文件：

2、测试程序文件：

C++：

test.cpp

汇编语言：

test.asm

下板文件：

test.coe