

中山大学数据科学与计算机学院本科生实验报告

(2020 学年春季学期)

课程名称：计算机组成原理实验

任课教师：郭雪梅

助教：汪庭葳、丁文

年级&班级	19 级行政 1 班	专业(方向)	计算机科学与技术 (人工智能与大数据方向)
学号	19335015	姓名	陈恩婷
电话	+86 173 2420 9242	Email	chenent@mail2.sysu.edu.cn
开始日期	2020 年 12 月 25 日	完成日期	2021 年 1 月 11 日

一、实验题目

流水线 CPU 设计实验

二、实验目的

1. 了解流水线 CPU 基本功能部件的设计与实现方法。
2. 了解提高 CPU 性能的方法。
3. 掌握流水线 MIPS 微处理器的工作原理。
4. 理解数据冒险、控制冒险的概念以及流水线冲突的解决方法。
5. 掌握流水线 MIPS 微处理器的测试方法。

三、实验设备

1. 硬件平台

基于 Xilinx FPGA basys3 实验平台，在开发系统中实现 MIPS 微处理器。

2. 软件平台

操作系统：Win 7。

开发平台：Vivado 2017.4。

编程语言：VerilogHDL 硬件描述语言。

四、实验内容

利用HDL语言，基于Xilinx FPGA basys3实验平台，用Verilog HDL语言或VHDL语言来编写，实现流水线CPU的设计，这个流水线CPU至少支持add、sub、and、or、addi、andi、

ori、lw、sw、beq、bne十一指令。

采用5级流水线技术（可选:具有数据前推机制），寄存器堆的写操作提前半个时钟周期，下降沿读取，上升沿写入数据。

1. 实验内容

实验中设计的各大模块如下：

- a. 取指令部分 IF
- b. 指令译码部分 ID
- c. 执行部分 EX
- d. 访存部分 MEM

同时该流水线 CPU 还包括了各级间寄存器 flipflop。

四、实验结果

实验结果对应的初始值如下：

- a. The content of RAM

memory_initialization_radix=16;

memory_initialization_vector=

00000000, 00000000, 00000000, 00000000, 00000000, 00000002,
00000002, 00000000, 00000000, 00000000, 00000000, 00000000,
00000000, 00000000, 00000000, 00000000, 00000000, 00000000,
00000000, 00000000, 00000001, 00000004, 00000000, 00000000;

- b. Instructions in ROM

memory_initialization_radix=16;

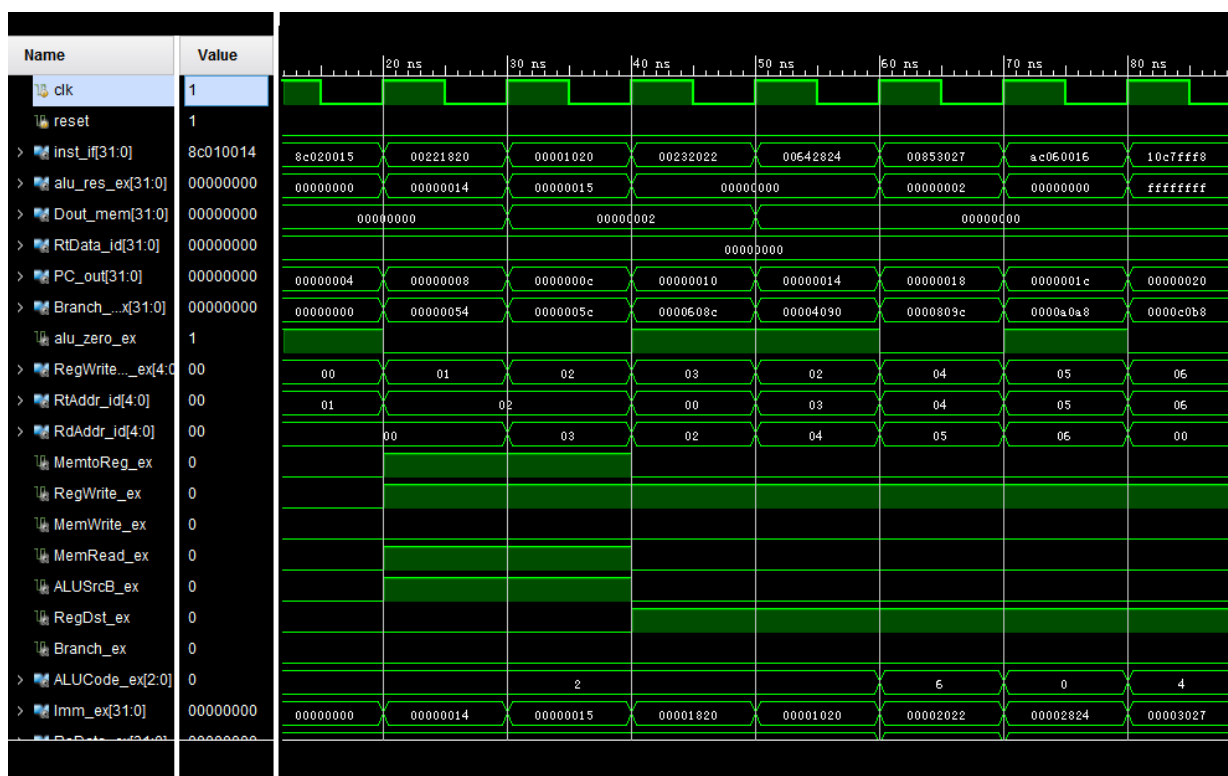
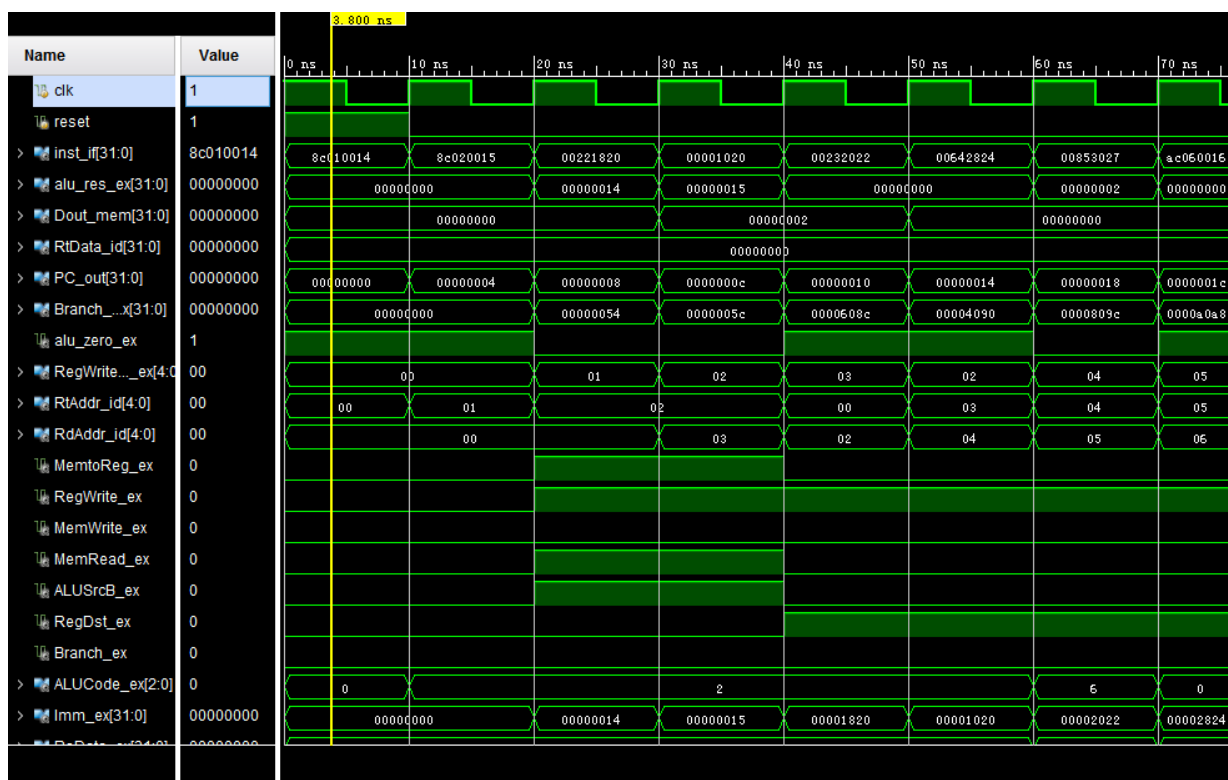
memory_initialization_vector=

8c010014, 8c020015, 00221820, 00001020,
00232022, 00642824, 00853027, ac060016,
10c7fff8;

The corresponding instructions are:

Instruction	Bin Code	Address	Inst. Type
lw r1, \$20(r0)	0x8c01_0014	0	6
lw r6, \$21(r0)	0x8c06_0015	1	6
add r3,r0,r0	0x0000_1820	2	1
add r2,r0,r0	0x0000_1020	3	1
sub r4,r1,r3	0x0023_2022	4	2
and r5,r4,r5	0x0085_3027	5	3
nor r6, r4, r5	0x0085_3027	6	5
sw r6, \$22(r0)	0xac06_0016	7	7
beq r6, r7, -8	0x10c7_fff8	8	8

实验的仿真波形图如下：



从图中可见：

a. lw r1, 20(\$r0)

地址 0x14 所存的值为 0, 该指令给 r1 赋值为 2。从仿真结果可见 ALU 的输出为 0x14。

b. lw r2, 21(\$r0)

注意到地址 0x15 处存储的值为 00000002，所以该指令应当给 r2 赋值为 2，但是这在 WB 阶段前不会发生。所以如果有指令在 WB 阶段前访问 r2 的值，就会发生数据冒险。

c. add r3, r1, r2

这里就出现了数据冒险，因为 r2 的值在 WB 之前都没有改变，r2 目前的值还是 0，这样会导致 r1+r2 的结果为 0。从仿真结果可见 ALU 的运算结果确实为 0 而不是 2。

d. add r2, r0, r0

ALU 的运算结果为 0。

e. sub r4, r1, r3

这里第一条指令的 WB 阶段已结束，r1 的值为 2，r3 为 0，r4 = 2。

f. and r5, r3, r4

ALU 的运算结果为 0。

g. nor r5, r4, r5

这里因为 r4 的值还没有更新，r4、r5 都为零，r6 应为 ffffffff，如图所示。

h. sw r6, 22(\$r0)

将数值存入 r6。

i. beq r6, r7, -8

最后一条指令，最后从 0x24 跳转到 0x08。

五、实验感想

1. 实验中遇到的问题及解决方案

a. 仿真运行时有些信号出现了红色

解决方案：返回检查前面的步骤，发现ALUCode[2:0]的输入值不在代码定义的范围，修改后运行成功了。

b. 输出不符合期望

解决方案：仔细检查各模块输入输出，发现有的模块忘记添加进工程文件中，

完善工程后输出正确了。

2. 总结流水线CPU设计体会

a. 做实验时自己排查和解决问题，并且分析实验结果的能力很重要，尤其是小的

细节，不能忽略。

b. 设计流水线CPU，关键是处理好各级之间数据传递的关系，搞清楚哪些数据需要

传递，如何传递（使用级间寄存器）等等。