**实验报告**

2019 年 5 月 14 日 成绩：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名 | 於文卓 | 学号 | 17061833 | 班级 | 17052317 |
| 专业 | 计算机科学与技术 | | 课程名称 | 计算机组成原理课程设计 | |
| 任课老师 | 冯建文 | 指导老师 | 冯建文 | 机位号 |  |
| 实验序号 | 8 | 实验名称 | 实现R型指令的CPU设计实验 | | |
| 实验时间 | 5月14日 | 实验地点 | 1-225 | 实验设备号 | #9 |
| **一、实验目的与要求** | | | | | |
| 1. 实验目的： 2. 掌握MIPS R型指令的数据通路设计，掌握指令流和数据流的控制方法 3. 掌握完整的单周期CPU顶层模块的设计方法 4. 实现MIPS R型指令的功能 5. 实验要求： 6. 在实验三、实验四、实验五和实验六的基础上，编写一个CPU模块，能够实现8条指定的R型指令 7. 编写一段测试8条指令的汇编程序，使用实验六的汇编器，将其翻译成二进制机器码，并通过关联文件初始化指令存储器。 8. 编写一个实验验证的顶层模块，用于验证实验 | | | | | |
| **二、实验设计与程序代码** | | | | | |
| 1. 模块设计说明   （1）在CPU的顶层模块中调用之前设计好的取指令模块、寄存器模块、ALU模块；  （2）在取指令模块中重新生成指令寄存器ROM  （3）在CPU的顶层中还要对指令进行译码，设置set\_of和set\_zf  （4）再设计一个顶层模块，调用消抖模块、数码管模块和cpu的顶层模块，用于板级验证   1. 实验程序源代码及注释等 2. CPU顶层模块     取指令    寄存器堆    ALU     1. 板级验证顶层模块     消抖模块    数码管模块 | | | | | |
| **三、实验仿真** | | | | | |
| 1. 仿真代码      1. 仿真波形      1. 仿真结果分析   通过仿真波形可以看到，开始时经过rst\_初始化后，每来一个时钟信号clk，f值就改变，为ALU运算结果。在220ns的时候，rst\_为0，此时f清零。当f为全零时，zf信号为1.  仿真结果符合预期，仿真正确。 | | | | | |
| **四、电路图** | | | | | |
|  | | | | | |
| **五、引脚配置（约束文件）** | | | | | |
| NET "scan\_clk" LOC = E3 |IOSTANDARD = LVCMOS18;  NET "clk" LOC = P18 |IOSTANDARD = LVCMOS18 |CLOCK\_DEDICATED\_ROUTE = FALSE;  NET "rst\_" LOC = N17 |IOSTANDARD = LVCMOS18 |CLOCK\_DEDICATED\_ROUTE = FALSE;  NET "of" LOC = U6 |IOSTANDARD = LVCMOS18;  NET "zf" LOC = R5 |IOSTANDARD = LVCMOS18;  NET "wei[7]" LOC = C9 |IOSTANDARD = LVCMOS18;  NET "wei[6]" LOC = C10 |IOSTANDARD = LVCMOS18;  NET "wei[5]" LOC = D10 |IOSTANDARD = LVCMOS18;  NET "wei[4]" LOC = C11 |IOSTANDARD = LVCMOS18;  NET "wei[3]" LOC = M17 |IOSTANDARD = LVCMOS18;  NET "wei[2]" LOC = J14 |IOSTANDARD = LVCMOS18;  NET "wei[1]" LOC = K13 |IOSTANDARD = LVCMOS18;  NET "wei[0]" LOC = P14 |IOSTANDARD = LVCMOS18;  NET "duan[7]" LOC = F14 |IOSTANDARD = LVCMOS18;  NET "duan[6]" LOC = N14 |IOSTANDARD = LVCMOS18;  NET "duan[5]" LOC = J13 |IOSTANDARD = LVCMOS18;  NET "duan[4]" LOC = G13 |IOSTANDARD = LVCMOS18;  NET "duan[3]" LOC = F13 |IOSTANDARD = LVCMOS18;  NET "duan[2]" LOC = G14 |IOSTANDARD = LVCMOS18;  NET "duan[1]" LOC = M13 |IOSTANDARD = LVCMOS18;  NET "duan[0]" LOC = H14 |IOSTANDARD = LVCMOS18; | | | | | |
| **六、思考与探索** | | | | | |
| 1. 实验结果记录：  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 序号 | 指令 | 执行结果 | 标志 | | 结论 | | ZF | OF | | 1 | 00000827 | FFFF\_FFFF | 0 | 0 | 正确 | | 2 | 0001102b | 0000\_0001 | 0 | 0 | 正确 | | 3 | 00421820 | 0000\_0002 | 0 | 0 | 正确 | | 4 | 00622020 | 0000\_0003 | 0 | 0 | 正确 | | 5 | 00832820 | 0000\_0005 | 0 | 0 | 正确 | | 6 | 00a33020 | 0000\_0007 | 0 | 0 | 正确 | | 7 | 00463804 | 0000\_000E | 0 | 0 | 正确 | | 8 | 00a64820 | 0000\_000C | 0 | 0 | 正确 | | 9 | 01264004 | 0000\_7000 | 0 | 0 | 正确 | | 10 | 00284826 | FFFF\_8FFF | 0 | 0 | 正确 | | 11 | 01215020 | FFFF\_8FFE | 0 | 0 | 正确 | | 12 | 01075822 | 0000\_6FF2 | 0 | 0 | 正确 | | 13 | 00e86022 | FFFF\_900E | 0 | 0 | 正确 | | 14 | 012c6824 | FFFF\_800E | 0 | 0 | 正确 | | 15 | 012c7025 | FFFF\_9FFF | 0 | 0 | 正确 | | 16 | 00c77825 | 0000\_000F | 0 | 0 | 正确 | | 17 | 00c78027 | FFFF\_FFF0 | 0 | 0 | 正确 | | 18 | 00e38820 | 0000\_0010 | 0 | 0 | 正确 | | 19 | 02289004 | 7000\_0000 | 0 | 0 | 正确 | | 20 | 02239804 | 0002\_0000 | 0 | 0 | 正确 | | 21 | 00f3a004 | 8000\_0000 | 0 | 0 | 正确 | | 22 | 0281a820 | 7FFF\_FFFF | 0 | 1 | 正确 | | 23 | 0255b025 | 7FFF\_FFFF | 0 | 1 | 正确 | | 24 | 0296b820 | FFFF\_FFFF | 0 | 0 | 正确 | | 25 | 0296c022 | 0000\_0001 | 0 | 1 | 正确 | | 26 | 02d4c822 | FFFF\_FFFF | 0 | 1 | 正确 | | 27 | 0241d026 | 8FFF\_FFFF | 0 | 1 | 正确 | | 28 | 02d4d82b | 0000\_0001 | 0 | 1 | 正确 | | 29 | 0354e02b | 0000\_0000 | 1 | 1 | 正确 | | 30 | 02c2e820 | 8000\_0000 | 0 | 1 | 正确 | | 31 | 0282f022 | 7FFF\_FFFF | 0 | 1 | 正确 | | 32 | 017af820 | 9000\_6FF1 | 0 | 0 | 正确 |  1. 实验结论：   每次按下P18按钮，都能正确显示ALU的F值，并且表示LED灯亮起正确。  实验正确   1. 问题与解决方案： 2. 拷贝指令寄存器的时候，无法读出指令——需要重新生成IP核 3. 用按钮作为时钟信号的时候，按下一次读出多条指令——需要对按钮进行防抖 4. 通过设置set\_of 和 set\_zf 来输出正确的of 和 zf 5. 思考题：      |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |   经过测试，ISE中的<符号是对无符号数的判断，因此要实现有符号数的比较，可以对符号位做判断，在符号位不同的情况下正数大于负数，在符号位相同时，用和sltu一样的方法比较大小 | | | | | |