实验4 简单计算机系统-系统设计B-实验报告

电 25 吴晨聪 2022010311

# 4.1 cpuD模块设计

上次任务的基础上增加能够执行三条立即数运算代码的能力，相应的，只需要加入一个数据选择器并在控制器做些调整即可。数据选择器放在第二个运算数srcB之前，对寄存器的第二个输出q2和imm之间做选择，如果是I型指令则选择imm，如果是R型指令则选择q2。控制器输出的其余控制信号和R指令一致即可。

此外，立即数运算的加入使得寄存器的值可以通过指令写入，先对某个寄存器单元与0做与运算，即可清空原有值，再对一个需要输入的数做加运算，即可存入该值。仿真结果如下图1所示：

一張含有 螢幕擷取畫面, 電子產品, 電子工程, 電路 的圖片

AI 產生的內容可能不正確。

图1 cpuD仿真运行结果

由图1可见，前4条指令实现了将R0和R1先清零，再分别输入11和22，随后的运算结果和实验3的最后任务的运行结果一致，说明存入的数据可以正常运算，该代码运行正常。

rom存入的机器码与汇编语言格式指令之间的关系如下表所示。

|  |  |  |
| --- | --- | --- |
| 行号 | 指令代码 | 机器码 |
| 0 | ANDI R0,R0,0 | 0x8000 |
| 1 | ANDI R1,R1,0 | 0x8500 |
| 2 | ADDI R0,R0,11 | 0xa00b |
| 3 | ADDI R1,R1,22 | 0xa516 |
| 4 | AND R2,R0,R1 | 0x0180 |
| 5 | OR R3,R0,R1 | 0x11c0 |
| 6 | ADD R2,R2,R3 | 0x2b80 |
| 7 | SUB R3,R3,R2 | 0x3ec0 |
| 8 | ADD R3,R3,R2 | 0x2ec0 |
| 9 | ADDC R2,R0,R1 | 0x6180 |
| 10 | SUB R1,R3,R2 | 0x3e40 |
| 11 | SUBC R2,R3,R2 | 0x5e80 |
| 12 | SLT R2,R1,R0 | 0x4480 |
| 13 | SLT R3,R0,R1 | 0x41c0 |
| 14 | SUB R2,R0,R1 | 0x3180 |
| 15 | SUBC R3,R0,R1 | 0x51c0 |

# 4.2 cpuE设计

该任务的内容是增加lw和sw两条指令，可以写入和读出RAM的值。在sw时，控制器需要把写入RAM的标志信号memwrite置为1，寄存器的写入信号regwrite置为0，ALU的第二个运算数仍然取imm即可；在lw时，则相反，把menwrite置为1，把regwrite置为0，此外还要将存入寄存器的数据选为RAM的值。

实际运行时发现了一个问题，由于RAM的输入端设置了寄存器，所以在读取数据时会比理论上的速度要慢一个周期。可以采取的方法有：RAM的clk设置为一个为主clk数倍的时钟；将lw指令的执行时间延长。本任务中采取了后者，将一个lw的指令延长到2个周期来执行，使得RAM中的数据能够正常存入到regfile内。运行仿真的结果如下图2所示：

一張含有 螢幕擷取畫面, 文字, 電子產品, 電路 的圖片

AI 產生的內容可能不正確。

图2 cpuE仿真运行结果

由图所见，（在左半段代码执行结束以后）R2和R3寄存器中原本存储的是14和13，（在右半段代码执行结束以后）R2和R3存储的值变为了13和14，说明该程序成功运行。

rom存入的机器码与汇编语言格式指令之间的关系如下表所示。

|  |  |  |
| --- | --- | --- |
| 行号 | 指令代码 | 机器码 |
| 0 | ANDI R0,R0,0 | 0x8000 |
| 1 | ANDI R1,R1,0 | 0x8500 |
| 2 | ADDI R0,R0,11 | 0xa00b |
| 3 | ADDI R1,R1,22 | 0xa516 |
| 4 | AND R2,R0,R1 | 0x0180 |
| 5 | OR R3,R0,R1 | 0x11c0 |
| 6 | ADD R2,R2,R3 | 0x2b80 |
| 7 | SUB R3,R3,R2 | 0x3ec0 |
| 8 | ADD R3,R3,R2 | 0x2ec0 |
| 9 | ADDC R2,R0,R1 | 0x6180 |
| 10 | SUB R1,R3,R2 | 0x3e40 |
| 11 | SUBC R2,R3,R2 | 0x5e80 |
| 12 | SLT R2,R1,R0 | 0x4480 |
| 13 | SLT R3,R0,R1 | 0x41c0 |
| 14 | SUB R2,R0,R1 | 0x3180 |
| 15 | SUBC R3,R0,R1 | 0x51c0 |
| 16 | ANDI R0,R0,0 | 0x8000 |
| 17 | SW R2, R0, 11 | 0xc20b |
| 18 | SW R3, R0, 12 | 0xc30c |
| 19 | LW R3, R0, 11 | 0xb30b |
| 20 | LW R3, R0, 11 | 0xb30b |
| 19 | LW R2, R0, 12 | 0xb20c |
| 20 | LW R2, R0, 12 | 0xb20c |
| 21 | ADD R2,R2,R0 | 0x2880 |
| 22 | ADD R3,R3,R0 | 0x2cc0 |