

朱志儒-16337341-计算机组成原理（四）

一、

1. ADD \$s1, \$s2, \$s3

机器代码：00000010010100111000100000100000

执行步骤：从指令存储器中取出指令，指令[31-26]传入控制单元，产生的信号有

RegDst: 1, Jump: 1, Branch: 0, MemRead: 0, MentoReg: 0, ALUOp: 10, MemWrite: 0, ALUSrc: 0, RegWrite: 1。由于 Branch: 0, Jump: 1 则 PC 自增为 PC + 4。指令[25-21]，指令[20-16]分别传入寄存器单元，读取寄存器 \$s2, \$s3 的数据，数据 1 传入 ALU，由于 ALUSrc: 0，则数据 2 传入 ALU。由于 ALUOp: 10，ALU 控制单元通过读取指令[5-0]产生 ALU 控制信号 0010，信号传给 ALU，ALU 执行加操作。由于 MentoReg: 0，运算结果传到寄存器单元，由于 RegDst: 1，根据指令[15-11]选择目标寄存器 \$s1，由于 RegWrite: 1，最后结果写入寄存器 \$s1。指令执行完毕。

RegDst	Jump	Branch	MemRead	MentoReg	ALUOp	MemWrite	ALUSrc	RegWrite
1	1	0	0	0	10	0	0	1

2. SUB \$s4, \$t5, \$t9

机器代码：00000001101110011010000000100010

执行步骤：从指令存储器中取出指令，指令[31-26]传入控制单元，产生的信号有

RegDst: 1, Jump: 1, Branch: 0, MemRead: 0, MentoReg: 0, ALUOp: 10, MemWrite: 0, ALUSrc: 0, RegWrite: 1。由于 Branch: 0, Jump: 1 则 PC 自增为 PC + 4。指令[25-21]，指令[20-16]分别传入寄存器单元，读取寄存器 \$t5, \$t9 的数据，数据 1 传入 ALU，由于 ALUSrc: 0，则数据 2 传入 ALU。由于 ALUOp: 10，ALU 控制单元通过读取指令[5-0]产生 ALU 控制信号 0110，信号传给 ALU，ALU 执行减操作。由于 MentoReg: 0，运算结果传到寄存器单元，由于 RegDst: 1，根据指令[15-11]选择目标寄存器 \$s4，由于 RegWrite: 1，最后结果写入寄存器 \$s4。指令执行完毕。

RegDst	Jump	Branch	MemRead	MentoReg	ALUOp	MemWrite	ALUSrc	RegWrite
1	1	0	0	0	10	0	0	1

3. AND \$t1, \$t2, \$s3

机器代码：00000001101110011010000000100010

执行步骤：从指令存储器中取出指令，指令[31-26]传入控制单元，产生的信号有

RegDst: 1, Jump: 1, Branch: 0, MemRead: 0, MentoReg: 0, ALUOp: 10, MemWrite: 0, ALUSrc: 0, RegWrite: 1。由于 Branch: 0, Jump: 1 则 PC 自增为 PC + 4。指令[25-21]，指令[20-16]分别传入寄存器单元，读取寄存器 \$t2, \$s3 的数据，数据 1 传入 ALU，由于 ALUSrc: 0，则数据 2 传入 ALU。由于 ALUOp: 10，ALU 控制单元通过读取指令[5-0]产生 ALU 控制信号 0000，信号传给 ALU，ALU 执行与操作。由于 MentoReg: 0，运算结果传到寄存器单元，由于 RegDst: 1，根据指令[15-11]选择目标寄存器 \$t4，由于 RegWrite: 1，最后结果写入寄存器 \$t4。指令执行完毕。

RegDst	Jump	Branch	MemRead	MentoReg	ALUOp	MemWrite	ALUSrc	RegWrite
1	1	0	0	0	10	0	0	1

4. OR \$t6, \$t7, \$s3

机器代码: 00000001111100110111000000100101

执行步骤: 从指令存储器中取出指令, 指令[31-26]传入控制单元, 产生的信号有

RegDst: 1, Jump: 1, Branch: 0, MemRead: 0, MentoReg: 0, ALUOp: 10, MemWrite: 0, ALUSrc: 0, RegWrite: 1。由于 Branch: 0, Jump: 1 则 PC 自增为 PC + 4。指令[25-21], 指令[20-16]分别传入寄存器单元, 读取寄存器 \$t7, \$s3 的数据, 数据 1 传入 ALU, 由于 ALUSrc: 0, 则数据 2 传入 ALU。由于 ALUOp: 10, ALU 控制单元通过读取指令[5-0]产生 ALU 控制信号 0001, 信号传给 ALU, ALU 执行或操作。由于 MentoReg: 0, 运算结果传到寄存器单元, 由于 RegDst: 1, 根据指令[15-11]选择目标寄存器 \$t6, 由于 RegWrite: 1, 最后结果写入寄存器 \$t6。指令执行完毕。

RegDst	Jump	Branch	MemRead	MentoReg	ALUOp	MemWrite	ALUSrc	RegWrite
1	1	0	0	0	10	0	0	1

5. SLT \$t0, \$s3, \$s4

机器代码: 00000010011101000100000000101010

执行步骤: 从指令存储器中取出指令, 指令[31-26]传入控制单元, 产生的信号有

RegDst: 1, Jump: 1, Branch: 0, MemRead: 0, MentoReg: 0, ALUOp: 10, MemWrite: 0, ALUSrc: 0, RegWrite: 1。由于 Branch: 0, Jump: 1 则 PC 自增为 PC + 4。指令[25-21], 指令[20-16]分别传入寄存器单元, 读取寄存器 \$s3, \$s4 的数据, 数据 1 传入 ALU, 由于 ALUSrc: 0, 则数据 2 传入 ALU。由于 ALUOp: 10, ALU 控制单元通过读取指令[5-0]产生 ALU 控制信号 0111, 信号传给 ALU, ALU 执行小于则置位操作。由于 MentoReg: 0, 运算结果传到寄存器单元, 由于 RegDst: 1, 根据指令[15-11]选择目标寄存器 \$t0, 由于 RegWrite: 1, 最后结果写入寄存器 \$t0。指令执行完毕。

RegDst	Jump	Branch	MemRead	MentoReg	ALUOp	MemWrite	ALUSrc	RegWrite
1	1	0	0	0	10	0	0	1

6. LW \$ra, 4(\$sp)

机器代码: 10001111101111110000000000000100

执行步骤: 从指令存储器中取出指令, 指令[31-26]传入控制单元, 产生的信号有

RegDst: 0, Jump: 1, Branch: 0, MemRead: 1, MentoReg: 1, ALUOp: 00, MemWrite: 0, ALUSrc: 1, RegWrite: 1。由于 Branch: 0, Jump: 1, 则 PC 自增为 PC + 4。指令[25-21]传入寄存器单元, 读取寄存器 \$sp 的数据, 数据 1 传入 ALU, 由于 ALUSrc: 1, 指令[15-0]经过符号扩展传入 ALU。由于 ALUOp: 00, ALU 控制单元产生 ALU 控制信号 0010, 信号传给 ALU, ALU 执行加操作。运算结果传到数据存储器以确定操作数据的地址。由于 MemRead: 1, 读取数据存储器中的数据。由于 MentoReg: 1, 读取的数据传入寄存器单元。由于 RegDst: 0, 根据指令[20-16]选择目标寄存器 \$ra, 由于 RegWrite: 1, 最后结果写入寄存器 \$ra。指令执行完毕。

RegDst	Jump	Branch	MemRead	MentoReg	ALUOp	MemWrite	ALUSrc	RegWrite
0	1	0	1	1	00	0	1	1

7. SW \$a0, 8(\$sp)

机器代码: 10101111101001000000000000001000

执行步骤: 从指令存储器中取出指令, 指令[31-26]传入控制单元, 产生的信号有

RegDst: X, Jump: 1, Branch: 0, MemRead: 0, MentoReg: X, ALUOp: 00, MemWrite: 1, ALUSrc: 1, RegWrite: 0。由于 Branch: 0, Jump: 1, 则 PC 自增为 PC + 4。指令[25-21], 指令[20-16]分别传入寄存器单元, 读取寄存器 \$sp, \$a0 的数据, 数据 1 传入 ALU, 由于 ALUSrc: 1, 指令[15-0]经过符号扩展传入 ALU。由于 ALUOp: 00, ALU 控制单元产生 ALU 控制信号 0010, 信号传给 ALU, ALU 执行加操作。运算结果传到数据存储器以确定操作数据的地址。由于 MemWrite: 1, 将数据 2 写入数据存储器。指令执行完毕。

RegDst	Jump	Branch	MemRead	MentoReg	ALUOp	MemWrite	ALUSrc	RegWrite
X	1	0	0	X	00	1	1	0

8. J 20000

机器代码: 00001000000000100000000000000000

执行步骤: 从指令存储器中取出指令, 指令[31-26]传入控制单元, 产生的信号有

RegDst: X, Jump: 0, Branch: X, MemRead: X, MentoReg: X, ALUOp: XX, MemWrite: X, ALUSrc: X, RegWrite: X。由于 Jump: 0, 指令[25-0]左移 2 位, 再将 PC+4 的高 4 位与其连接, 最后的结果传给 PC。指令执行完毕。

RegDst	Jump	Branch	MemRead	MentoReg	ALUOp	MemWrite	ALUSrc	RegWrite
X	0	X	X	X	XX	X	X	X

9. 20000H BEQ \$s3, \$s4, EXIT

.....

40000H EXIT

机器代码: 00010010011101000111111111111111

执行步骤: 从指令存储器中取出指令, 指令[31-26]传入控制单元, 产生的信号有

RegDst: X, Jump: 1, Branch: 1, MemRead: 0, MentoReg: X, ALUOp: 01, MemWrite: 0, ALUSrc: 0, RegWrite: 0。由于 Branch: 1, 指令[25-21], 指令[20-16]分别传入寄存器单元, 读取寄存器 \$s3, \$s4 的数据, 数据 1 传入 ALU, 由于 ALUSrc: 0, 数据 2 传入 ALU。由于 ALUOp: 01, ALU 控制单元产生 ALU 控制信号 0110, 信号传给 ALU, ALU 执行减操作。若 ALU 的零标志为 0, 则 PC 自增为 PC+4; 若 ALU 的零标志为 1, 则指令[15-0]符号扩展再左移 2 位, 接着与 PC+4 的值相加, 最后的结果传入 PC。指令执行完毕。

RegDst	Jump	Branch	MemRead	MentoReg	ALUOp	MemWrite	ALUSrc	RegWrite
X	1	1	0	X	01	0	0	0

二、类型: R 型

汇编指令: SUB \$v1, \$v1, \$v0

二进制表示: 00000000011000100001100000100010

三、类型: L 型

汇编指令: LW \$v0, 4(\$at)

二进制表示: 100011000010001000000000000000100

四、流水线处理器的单周期为: 350ps

非流水线处理器的单周期为: 1250ps