计组第七章作业

孙启翔 241220098

April 2025

2(4).

1. 寄存器中：寄存器寻址
2. 存储器中：立即寻址、直接寻址、间接寻址、寄存器间接寻址、变址寻址、相对寻址、基址寻址

2(9).

1. 区别：执行调用指令时必须保存下条指令的地址
2. 转移跳转不需要，调用指令需要

3.

由于执行到转移指令时 PC 的内容为 258，并且该指令由两个字节构成，所以执行该指令时 PC 的值变为 260，要求转移到 220，偏移量为-40，转换成二进制补码为 11011000

6.

由于二地址指令共有 k_2 条，而指令字长为 16 位，操作数的地址码长度为 6 位，所以还剩下 4 位可以作为操作码。所以前四位供单地址和零地址使用的指令数量共有 $16-k_2$ 种。此时共能满足的单地址指令前 10 位二进制数的数量最大为 $(16-k_2)\cdot 2^6$ 种，其中排除掉零地址指令所占用的数量共有 $\frac{k_0}{2^6}$。综上，单地址指令最多有 $(16-k_2)\cdot 2^6-\frac{k_0}{2^6}$ 条

7.

由该指令系统至多支持 64 种不同操作可知操作码的长度为 6 位，同时由于 CPU 中有 8 个通用寄存器，所以寄存器编码的长度为 3 位。下面来具体分析每种格式的指令长度、各字段所占位数以及含义、需要几次存储器访问。

	长度	位数及含义	访问次数
RR	16	6 位操作码 $+2\times3$ 位寄存器 +4 位空闲	0
RI	32	6 位操作码 +3 位寄存器 +16 位立即数 +7 位空闲	0
RS	16	6 位操作码 $+2\times3$ 位寄存器 +4 位空闲	1
RX	32	6 位操作码 $+2\times3$ 位寄存器 +16 位存储器 +4 位空闲	1
XI	48	6 位操作码 +3 位寄存器 +16 位立即数 +16 位存储器 +7 位空闲	1
SI	32	6 位操作码 +3 位寄存器 +16 位立即数 +7 位空闲	1
SS	16	6 位操作码 $+2\times3$ 位寄存器 +4 位空闲	2

8.

(1) 操作码共有 4 位，所以该指令系统最多有 $2^4=16$ 条指令表示寄存器的编号共有 3 位，所以最多可有 $2^3=8$ 个通用寄存器主存地址空间大小为 128KB，计算机字长为 16 位，即 2 字节，共有 $128KB/2=64KB=2^{16}$，所以 MAR 至少需要 16 位 MDR 的位数应该与计算机字长相同，即 16 位

(2) 主存中共有 2^{16} 个存储单元，所以转移指令的目标地址范围为 $0\sim2^{16}-1$

(3) 对应的机器码为：0010 001100 010101B，即 2315H R_4 储存的地址的内容对应为 5678H，R_5 储存的地址的内容对应为 1234H，二者加和得到 68ACH，并将结果存回 R_5 的地址对应的内容，即地址 5678H 对应的内容修改为 68ACH，同时 R_5 存储的地址自加，变为 5679H

10.

偏移量是带符号的，即如果低 12 位的符号位为 1，是一个负数，那么在高 20 位中就需要“补偿”这个负号带来的影响。

11.

```
addi t1 t0 0
sll t2 t0 1
add t1 t1 t2
sll t3 t0 2
add t1 t1 t3
```

13.

```
add t0, zero, zero          //将t0赋值为0
loop:                       //循环开始标签
    beq a1, zero, finish    //如果a1等于0，则跳转finish
    add t0, t0, a0          //t0 += a0
    addi a1, a1, -1         //a1 -= 1
    j loop                  //无条件跳转到loop处继续执行
finish:                     //循环结束标签
addi t0, t0, 100            //t0 += 100
add a0, t0, zero            //a0 = t0
```

这段代码用于计算 $100 + a \times b$ 的值，将 a1 的初始值赋为 b，将 a0 的初始值赋为 a，每次循环 t0+=a，共循环 b 次

15.

b = 31&a :

```
andi t1, t0, 31
```

b = 65535&a :

```
lui t1, 16
addi t1, t1, -1
and t1, t0, t1
```

17.

可能 here 和 there 语句间隔过大，超过了 16 位带符号整数表示的范围
修改方式：

```
here: beq t0, t2, tmp
tmp: j there
there: addi t1, t0, 4
```