

# 《计算机组成原理》

## （第五讲习题答案）

厦门大学信息学院软件工程系 曾文华

2022年5月4日

# 第5章 指令系统

- 5.1 指令系统概述
- 5.2 指令格式
- 5.3 寻址方式
- 5.4 指令类型
- 5.5 指令格式设计
- 5.6 CISC和RISC
- 5.7 指令系统举例

# 习题 (P182-185)

- 5.2
- 5.3
- 5.4
- 5.5
- 5.7
- 5.8
- 5.10
- 5.11
- 5.12

# 习题答案（P182-185）

- 5.1 解释下列名称

- **指令**：指令是控制计算机执行某种操作（如加、减、传送、转移等操作）的命令，它是CPU能直接识别并执行的基本功能单位。
- **指令系统**：一台计算机中所有指令的集合称为该计算机的指令系统（也称指令集）。
- **操作码**：指令包括操作码OP和地址码A，操作码OP用于解决进行何种操作的问题。
- **扩展操作码**：可以采用扩展操作码技术实现变长操作码，即操作码的长度随地址码数目的增加而减少。
- **地址码**：指令包括操作码OP和地址码A，地址码A用于解决处理什么操作数的问题，地址码A可以包括多个操作数。
- **寻址方式**：寻址方式就是寻找指令或操作数有效地址的方法。
- **程序计数器PC**：程序计数器PC用于保存将要执行的指令的字节地址（主存地址）。

- 5.1 解释下列名称（续）

- **有效地址**：指令和操作数在主存的地址，称为有效地址EA，Effective Address。
- **存储器堆栈**：堆栈（Stack）以先进后出（First In Last Out）的方式存储数据，堆栈通常有存储器堆栈和寄存器堆栈两种，设置在存储器中的堆栈，称为存储器堆栈。
- **寄存器堆栈**：堆栈（Stack）以先进后出（First In Last Out）的方式存储数据，堆栈通常有存储器堆栈和寄存器堆栈两种，设置在寄存器中的堆栈，称为寄存器堆栈。
- **基址寄存器**：在基址寻址方式下，指定一个寄存器用来存放基地址，这个寄存器就称为基址寄存器。
- **变址寄存器**：在变址寻址方式下，指定一个寄存器用来存放变化的地址，这个寄存器就称为变址寄存器。
- **转子指令**：子程序调用指令又称为转子指令或过程调用指令，转子指令中必须明确给出子程序的入口地址。
- **CISC**：Complex Instruction Set Computer，复杂指令系统计算机；计算机系统设计者在设计指令系统时，增加了越来越多的功能强大的复杂指令，以及更多的寻址方式，以满足来自不同方面的需求；因此，计算机的指令系统越来越庞大、复杂，称为复杂指令系统计算机（CISC）。
- **RISC**：Reduced Instruction Set Computer，精简指令系统计算机；精简指令系统计算机（RISC）体系结构的基本思想：针对CISC指令系统指令种类太多、指令格式不规范、寻址方式太多的缺点，通过减少指令种类、规范指令格式和简化寻址方式，来方便处理器内部的并行处理，从而大幅度提高处理器的性能。

## • 5.2 选择题

### – (1) A

- 三地址指令：29条，操作码OP需要5位，OP (xxxxx) A1 (6位) A2 (6位) A3 (6位)；用掉29条，剩余3种情况 ( $2^5=32$ ,  $32-29=3$ )
- 二地址指令：OP (xxxxx xxxxxx) A1 (6位) A2 (6位) 可以有 $3 \times 2^6=192$ 种情况，满足107条的要求
- 因此，指令长度=23位，因为计算机按字节编址，指令长度必须是字节的倍数，因此指令长度=24位

### – (2) A

- 指令长度=32位 OP=8位
- Store指令：sw rt,imm(rs) OP (8位) rt (4位) rs (4位) Imm (? 位) ? 位=32-4-4-8=16位
- 16位补码的表示范围：-32768 ~ +32767

### – (3) A

- 指令长度=16位 48条指令，因此OP=6位 4种寻址方式，因此寻址特征位I=2位
- 单地址指令：OP (6位) I (4位) A (? 位) A=16-6-2=8位
- 8位地址的范围是：0 ~ 255

### – (4) C

- 先变址后间址的寻址方式：EA=((I)+D)

### – (5) C

- 因为指令字长=16位，主存按字节编址，因此转移指令取出后，PC=2000H+2=2002H
- 目标地址=2002H+06H=2008H

### – (6) A

- 间接寻址：不属于偏移寻址
- 基址寻址、相对寻址、变址寻址，其有效地址都是：寄存器内容+偏移量（形式地址），都属于偏移寻址

## • 5.2 选择题（续）

- (7) **D**
  - 变址寻址方式的 $EA=(R)+形式地址=1000H+2000H=3000H$
  - 操作数 $=(EA)=(3000H)=4000H$
- (8) **D**
  - 变址寻址方式的 $EA=X$ （变址寄存器） $+D$ （形式地址），其中 $X$ 变化、 $D$ 不变，变址寻址方式主要用于对数组元素进行重复的访问。
- (9) **D**
  - 因为形式地址是用补码表示， $FF12H$ 扩展到32位为： $FFFF\ FF12$
  - 基址寻址方式的 $EA=基址寄存器内容+形式地址=F000\ 0000H + FFFF\ FF12H= EFFF\ FF12H$
  - 因为是大端方式，按字节编址， $EA$ 对应的是操作数的最高有效字节，即 $12H$ ，操作数的LBS（最低有效字节）的地址 $=EA+3=EFFF\ FF12H + 3=EFFF\ FF15H$
- (10) **B**
  - 变址寻址方式的 $EA=sizeof(double)*X$ （变址寄存器） $+D$ （形式地址） $=8*X+D$
  - 这里， $D=首地址=2000H$ ， $EA=2100H$ ，因此变址寄存器 $X$ 的内容 $=(2100H-2000H)/8=100H/8=256/8=32$
- (11) **C**
  - 无符号数大于： $A-B$ ；因为 $A>B$ ，因此无进位/借位，且结果 $\neq 0$ ；因此： $CF=0$ ， $ZF=0$ ， $/(CF+ZF)=1$
- (12) **A**
  - $(R1)=FFFFFFFFH=-1$      $(R2)=FFFFFFF0H=-16$      $(R1)-(R2)=15$  无溢出、无进位/借位；因此： $OF=0$ ， $CF=0$
- (13) **A**
  - **A错：**RISC计算机普遍采用硬布线控制器
  - B、C、D：对

## • 5.3 简答题

— (1) 什么叫指令？什么叫指令系统？

— 答：

- 指令是控制计算机执行某种操作（如加、减、传送、转移等操作）的命令，它是CPU能直接识别并执行的基本功能单位。
- 一台计算机中所有指令的集合称为该计算机的指令系统（也称指令集）。

— (2) 计算机中为什么要设置多种操作数寻址方式？

— 答：

- 能给用户提供更丰富的程序设计手段，有利于编译器实现高级语言向汇编语言的转换，方便在效率和方便性以及寻址空间大小方面进行折中平衡。
- 立即数寻址和寄存器寻址速度最快，但是寄存器数据有限、立即数范围也非常有限；间接寻址、寄存器间接寻址、基址寻址可以扩大寻址范围；变址寻址、相对寻址、直接寻址可以提高程序设计的灵活性。

— (3) 操作数寻址方式在指令中如何表示？

— 答：

- 有的计算机在指令字中用明确的字段（寻址方式字段）表示操作数的寻址方式，如PDP-11、x86指令。
- 有的计算机将操作数寻址方式隐含在操作码中，如MIPS、RISC-V指令集。

— (4) 基址寻址和变址寻址的作用是什么？分析它们的异同点。

— 答：

- 基址寻址是面向系统的，用于程序的重定位和扩展寻址空间，解决程序逻辑空间和存储器物理空间的无关性。
- 变址寻址是面向用户的，主要解决程序循环问题，方便用户编写出高效访问存储空间的程序。
- 相同的：二者在形式上及计算操作数有效地址的方法上相似，都是将寄存器的值加上形式地址，形成操作数的有效地址。
- 不同点：基址寄存器的值通常是不变的，程序中所有地址都是相对于基址来变化的，形式地址表示的偏移量位数较短，偏移范围较小。变址寻址则相反，指令中形式地址给出的是一个存储器地址基准，变址寄存器中存放的是偏移量，不同的变址寄存器给出不同的单元，偏移量位数足以表示整个存储空间。



## • 5.3 简答题（续）

### — （5）RISC处理器有何特点？

#### — 答：

- ① 优先选择使用频率最高的一些简单指令，以及一些很有用但不复杂的指令，避免使用复杂指令
- ② 大多数指令在一个时钟周期内执行完成
- ③ 采用LOAD/STORE结构，只允许LOAD/STORE指令访问主存，其余指令只能对寄存器操作数进行处理
- ④ 采用简单的指令格式和寻址方式，指令长度固定
- ⑤ 固定的指令格式：指令长度、格式固定，可简化指令的译码逻辑，有利于提高流水线的执行效率；为了便于编译的优化，常采用三地址指令格式
- ⑥ 面向寄存器的结构；为减少访问主存，CPU内应设大量的通用寄存器
- ⑦ 采用硬布线控制逻辑；由于指令系统的精简，控制部件可由组合逻辑实现，不用或少用微程序控制，这样可使控制部件的速度大大提高
- ⑧ 注重编译的优化，力求有效地支持高级语言程序

### — （6）比较定长指令和变长指令的优缺点。

#### — 答：

- 定长指令的优点：结构规整，有利于简化硬件，尤其是指令译码部件的设计；缺点：指令字长平均长度较长，指令不易扩展。
- 变长指令的优点：结构灵活，能充分利用指令中的每一位，指令码点冗余少，指令字长平均长度较短，指令易于扩展；缺点：格式不规整，不同指令取指时间可能不同，扩展复杂。

### — （7）指令的地址码与指令中的操作码含义有何不同？

#### — 答：

- 指令的地址码通常指定操作数的地址，地址码字段的作用随指令类型和寻址方式的不同而不同，它可能作为一个操作数、操作数的地址（包括操作数所在的主存地址、寄存器的编号或外设的端口地址），也可能是一个用于计算地址的偏移量。
- 指令中的操作码则表示指令的功能。

- **5.4** 根据操作数所在的位置，在空格处填写其寻址方式：

- (1) 操作数在指令中为\_\_\_\_\_寻址方式。
- (2) 操作数地址（主存）在指令中为\_\_\_\_\_寻址方式。
- (3) 操作数在寄存器中为\_\_\_\_\_寻址方式。
- (4) 操作数地址在寄存器中为\_\_\_\_\_寻址方式。

- 答：

- (1) 立即数寻址
- (2) 直接寻址
- (3) 寄存器寻址
- (4) 寄存器间接寻址

- 5.5 某计算机字长为16位，运算器为16位，有16个通用寄存器，8种寻址方式，主存为128KW（注：W为字长，这里W=16位），指令中操作数地址码由寻址方式字段和寄存器号字段组成。请回答下列问题：

- (1) 单操作数指令最多有多少条？
- (2) 双操作数指令最多有多少条？
- (3) 直接寻址的范围多大？
- (4) 变址寻址的范围多大？

- 答：

- (1) 16个通用寄存器需要4位，8种寻址方式需要3位；单操作数指令：OP（? 位） I（3位） R（4位）， $OP=16-4-3=9$ 位，最多 $2^9=512$ 条
- (2) 双操作数指令：OP（? 位） I1（3位） R1（4位） I2（3位） R2（4位）， $OP=16-4-3-4-3=2$ 位，最多 $2^2=4$ 条
- (3) 直接寻址方式：OP（2位） I1（3位） R1（4位） R2（4位） A2（? 位）  $A2=16-2-4-3-4=3$ 位， $2^3=8$ ，直接寻址的范围=0~7
- (4) 变址寄存器的位数=运算器的位数=16位， $2^{16}=65536$ ，变址寻址的范围=0~65535

- 5.6 假设某计算机的指令长度固定为16位，具有双操作数、单操作数和无操作数等3类指令，每个操作数的地址规定用6位表示。
  - （1）若操作码字段不固定，现已设计出m条双操作数指令、n条无操作数指令，在此情况下，这台计算机最多可以设计出多少条单操作数指令？
  - （2）若操作码字段不固定，当双操作数指令取最大数时，且在此基础上，单操作数指令条数也取最大数，试计算这3类指令可拥有多少条指令？

• 答：

— （1）

- 双操作数指令：xxxx（4位） A1（6位） A2（6位），有 $2^4=16$ 种情况，现已有m条，剩余 $16-m$ 种情况用于单操作数指令和无操作数指令
- 单操作数指令：xxxx（4位） xxxxxx（6位） A（6位），有 $(16-m) \times 2^6$ 种情况，假设有z条单操作数指令，剩余 $(16-m) \times 2^6 - z$ 种情况用于无操作数指令
- 无操作数指令：xxxx（4位） xxxxxx（6位） xxxxxx（6位），有 $((16-m) \times 2^6 - z) \times 2^6 = n$ 种情况
- $z = (16-m) \times 2^6 - n \times 2^{-6} = (16-m) \times 64 - n/64$

— （2）双操作数指令最多可以是 $2^4-1=15$ 条，单操作数指令最多可以是 $1 \times 2^6-1=63$ 条，此时无操作数指令为 $1 \times 2^6=64$ 条

- **5.7** 设相对寻址的转移指令占3个字节，第一个字节是操作码，第二个字节是相对位移量（补码表示）的低8位，第三个字节是相对位移量（补码表示）的高8位，每当CPU从存储器取一个字节时，便自动完成 $(PC)+1 \rightarrow PC$ 。请回答下列问题：
  - （1）若PC当前值为256（十进制），要求转移到290（十进制），则转移指令第二、三字节的机器码是什么（十六进制）？
  - （2）若PC当前值为128（十进制），要求转移到110（十进制），则转移指令第二、三字节的机器码又是什么（十六进制）？

• 答：

- （1）PC当前值为256（十进制），转移指令取出后， $PC=256+3=259$ ，位移量 $=290-259=31=1F=001F$ ；转移指令第二个字节是1F，第三个字节是00。
- （2）PC当前值为128（十进制），转移指令取出后， $PC=128+3=131$ ，位移量 $=110-131=-21=EB=FFEB$ ；转移指令第二个字节是EB，第三个字节是FF。

- 5.8 计算机的指令格式包括操作码OP、寻址方式特征位I和形式地址D等3个字段，其中OP字段是6位，寻址方式特征位字段I为2位，形式地址字段D为8位。I的取值与寻址方式的对应关系如下：

I=00: 直接寻址（书上的题目有误）

I=01: 用变址寄存器X1进行寻址

I=10: 用变址寄存器X2进行寻址

I=11: 相对寻址

设(PC)=1234H, (X1)=0037H, (X2)=1122H, 以下4条指令均采用上述格式，请确定这些指令的有效地址：

(1) 4420H

(2) 2244H

(3) 1322H

(4) 3521H

- 答：

- (1) 4420H=0100 0100 0010 0000 =0100 01 00 0010 0000, 直接寻址, 有效地址=0010 0000=0020H
- (2) 2244H=0010 0010 0100 0100 =0010 00 10 0100 0100, 变址寻址X2, 有效地址=(X2)+0100 0100=1122H+44H=1166H
- (3) 1322H=0001 0011 0010 0010 =0001 00 11 0010 0010, 相对寻址, 有效地址=(PC)+2+0010 0010=1234+2+22H=1258H（注意：这里要用取出指令后的PC值=1234H+2）
- (4) 3521H=0011 0101 0010 0001 =0011 01 01 0010 0001, 变址寻址X1, 有效地址=(X1)+0010 0001=0037H+21H=0058H

- 5.9 设某计算机A有60条指令，指令的操作码字段固定为6位，从000000 ~ 111011，该计算机的后续机型B中需要增加20条指令，并与A保持兼容。

(1) 试采用扩展操作码为计算机B设计指令操作码。

(2) 求出计算机B中操作码的平均长度。

- 答：

— (1)

- 计算机A的操作码只剩下4种情况：111100、111101、111110、111111，可以将计算机B的操作码扩展到9位，增加的3位有8种组合，最多可以表示 $4 \times 8 = 32$ 条指令，满足增加20条指令的要求，计算机B的操作码为：

000000 ~ 111011	60条
111100 xxx	8条
111101 xxx	8条
111110 xxx	8条
111111 xxx	8条

- 如果是扩展到8位（增加2位，有4种组合），则最多可以表达 $4 \times 4 = 16$ 条指令，不够！

— (2)

- 计算机B的操作码平均长度 =  $(60 \times 6 \text{位} + 32 \times 9 \text{位}) / (60 + 32) = (360 + 288) / 92 = 7.04 \text{位}$

- 5.10 以下MIPS指令代表什么操作？写出它的MIPS汇编指令格式。

0000 0000 1010 1111 1000 0000 0010 0000

- 答：

— 因为前6位=000000，因此是R型指令：000000 rs (5位) rt (5位) rd (5位) shamt (5位) func (6位)

— 0000 0000 1010 1111 1000 0000 0010 0000 = 000000 00101 01111 10000 00000 100000

— rs=00101=\$a1

— rt=01111=\$t7

— rd=10000=\$s0

— shamt=00000

— func=100000=32，代表加法操作



— 汇编指令格式：add rd,rs,rt

— 汇编指令格式：add \$s0,\$a1,\$t7

表5.8 常用R型指令及其func编码

func (6位，十进制)	指令助记符	指令功能描述	备注
00	sll rd,rt,shamt	R[rd]=R[rt]<<shamt	逻辑左移指令，注意rs字段未使用
02	srl rd,rt,shamt	R[rd]=R[rt]>>shamt	逻辑右移指令，注意rs字段未使用
03	sra rd,rt,shamt	R[rd]=R[rt]>>shamt	算术右移指令，注意rs字段未使用
04	sliv rd,rt,rs	R[rd]=R[rt]<<R[rs]	可变左移指令
08	jr rs	PC=R[rs]	无条件跳转指令，R[rs]值应是4的倍数，字对齐
09	jalr rs	R[31]=PC+8 PC=R[rs]	子程序调用指令，R[31]保存程序的断点
12	syscall	系统调用指令	无操作数
16	mflr rd	R[rd]=HI	取HI寄存器的值指令，mflr指令取LO
17	mthi rs	HI=R[rs]	存HI寄存器的值指令，mtlo指令存LO
24	mult rs,rt	{HI,LO}=R[rs]*R[rt]	有符号乘指令，64位结果送入HI、LO寄存器
32	add rd,rs,rt	R[rd]=R[rs]+R[rt]	加法指令，溢出时发生异常，且不修改R[rd]
34	sub rd,rs,rt	R[rd]=R[rs]-R[rt]	减法指令，溢出时发生异常，且不修改R[rd]
36	and rd,rs,rt	R[rd]=R[rs]&R[rt]	逻辑与指令
37	or rd,rs,rt	R[rd]=R[rs] R[rt]	逻辑或指令
42	slt rd,rs,rt	R[rd]=(R[rs]<R[rt])?1:0	小于置位指令，有符号比较



- 5.11 假设以下C语言语句中包含的变量f、g、h、i、j分别存放在寄存器\$11~\$15中，请写出实现C语言语句f=(g+h)\*i/j功能的MIPS汇编指令序列，并写出每条MIPS指令的十六进制数。

答：

- f、g、h、i、j存放在寄存器\$11~\$15中，即：f=\$t3、g=\$t4、h=\$t5、i=\$t6、j=\$t7
- \$t3编号=01011、\$t4编号=01100、\$t5编号=01101、\$t6编号=01110、\$t7编号=01111

- g+h -> f: **add \$t3,\$t4,\$t5**
- f\*i -> \$lo: **mult \$t3,\$t6**
- \$lo -> f: **mflo \$t3**
- f/j -> \$lo: **div \$t3,\$t7**
- \$lo -> f: **mflo \$t3**

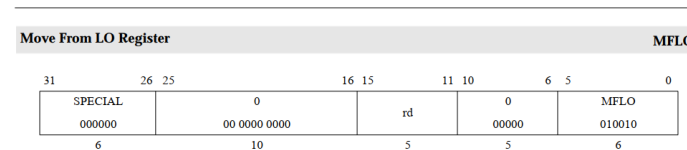
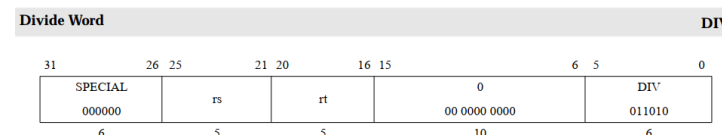
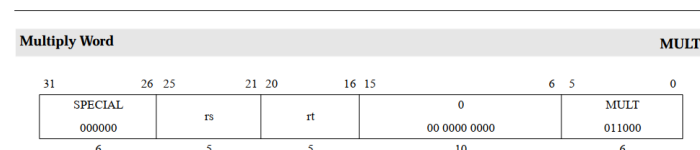
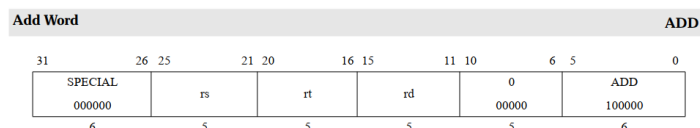
- 加法指令: add rd,rs,rt                      000000   Rs   Rt   Rd   00000 100000(32)
- 乘法指令: mult rs,rt                      000000   Rs   Rt   00000 00000 011000(24)
- 除法指令: div rs,rt                      000000   Rs   Rt   00000 00000 011010(26)
- mflo指令: mflo rd                      000000   00000 00000 Rd   00000 010010(18)

- add \$t3,\$t4,\$t5      机器码=000000 01100 01101 01011 00000 100000 = 0000 0001 1000 1101 0101 1000 0010 0000=**018D5820H**
- mult \$t3,\$t6      机器码=000000 01011 01110 00000 00000 011000 = 0000 0001 0110 1110 0000 0000 0001 1000=**016E0018H**
- mflo \$t3      机器码=000000 00000 00000 01011 00000 010010 = 0000 0000 0000 0000 0101 1000 0001 0010=**00005812H**
- div \$t3,\$t7      机器码=000000 01011 01111 00000 00000 011010 = 0000 0001 0110 1111 0000 0000 0001 1010=**016F001AH**
- mflo \$t3      机器码=000000 00000 00000 01011 00000 010010 = 0000 0000 0000 0000 0101 1000 0001 0010=**00005812H**

表5.7 MIPS的通用寄存器

寄存器#	助记符	释义
\$0	\$zero	固定值为0 硬件置位
\$1	\$at	汇编器保留，临时变量
\$2~\$3	\$v0~\$v1	函数调用返回值
\$4~\$7	\$a0~\$a3	4个函数调用参数
\$8~\$15	\$t0~\$t7	暂存寄存器，被调用者按需保存
\$16~\$23	\$s0~\$s7	save寄存器，调用者按需保存
\$24~\$25	\$t8~\$t9	暂存寄存器，同上
\$26~\$27	\$k0~\$k1	操作系统保留，中断异常处理
\$28	\$gp	全局指针 (Global Pointer)
\$29	\$sp	堆栈指针 (Stack Pointer)
\$30	\$fp	帧指针 (Frame Pointer)
\$31	\$ra	函数返回地址 (Return Address)

请参考MIPS32指令手册.pdf



- 5.12 某计算机字长为16位，主存地址空间大小为128KB，按字编址。采用单字长指令格式，指令各字段单元如图5.34所示。



图5.34 单字长指令各字段定义

转移指令采用相对寻址方式，相对偏移量用补码表示，寻址方式定义如表5.20所示。

表5.20 转移指令寻址方式

Ms/Md	寻址方式	助记符	定义
000B	寄存器直接寻址	Rn	操作数=(Rn)
001B	寄存器间接寻址	(Rn)	操作数=((Rn))
010B	寄存器间接+自增寻址	(Rn)+	操作数=((Rn))      (Rn)+1 -> (Rn)
011B	相对寻址	D(Rn)	转移目标地址=(PC)+(Rn)

注：(x)表示存储器地址x或寄存器x的内容。

请回答下列问题：

- (1) 该指令系统最多可有多少条指令？该计算机最多有多少个通用寄存器？
- (2) 存储器地址寄存器MAR和存储器数据寄存器MDR至少需要多少位？
- (3) 转移指令的目标地址范围是多少？
- (4) 若操作码0010B表示加法操作（助记符为add），寄存器R4和R5的编号分别为100B和101B，R4的内容为1234H，R5的内容为5678H，地址1234H中的内容为5678H，地址5678H中的内容为1234H，则汇编语言为“add (R4),(R5)+”（逗号前为源操作数，逗号后为目的操作数）对应的机器码是什么（十六进制）？该指令执行后，哪些寄存器和存储单元的内容会改变？改变后的内容是什么？

• 答：

- (1) 因为OP=4位，最多可有16条指令；因为Rs、Rd都是3位，最多有8个通用寄存器。
- (2) 因为主存为128KB，字长为16位；因此，存储器地址寄存器MAR需要16位（ $2^{16} \times 16\text{位} = 128\text{KB}$ ，红色的16为地址）；因为字长为16位，因此存储器数据寄存器MDR需要16位。
- (3) 转移指令采用相对寻址方式，计算机字长为16位，转移指令的目标地址范围是 $0 \sim 2^{16}-1$ （0 ~ 65535）
- (4)
  - “add (R4),(R5)+”指令：OP=0010；源操作数=(R4)=001 100（001表示寄存器间接寻址，100表示R4）；目的操作数=(R5)+=010 101（010表示寄存器间接+自增寻址，101表示R5）；对应的机器码=0010 001 100 010 101= 2315H
  - 该指令为自增寻址，执行后，R5寄存器加1，(R5)=5678H+1=5679H；该指令为加法指令，即：存储单元(1234H)加存储单元(5678)，结果送到存储单元(5678)，即5678H+1234H=68ACH，送(5678H)；因此存储单元(5678)的内容变为68ACH

- 5.13 某计算机采用16位定长指令格式，其CPU中有一个标志寄存器，其中包含进位/借位标志CF、零标志ZF和符号标志NF。假定为该计算机设计了条件转移指令，其格式如图5.35所示。

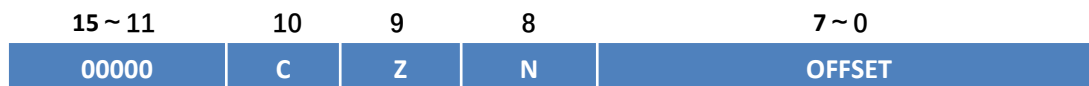


图5.35 条件转移指令格式

其中，00000为操作码OP。C、Z、N分别为CF、ZF、NF的对应检测位；某检测位为1时表示需检测对应标志，需检测的标志位中只要有一个为1就转移，否则不转移；例如，若C=1、Z=0、N=1，则需检测CF和NF的值，当CF=1或NF=1时发生转移。OFFSET是相对偏移量，用补码表示。转移执行时，转移目标地址为(PC)+2+OFFSETx2；顺序执行时，下一条指令的地址为(PC)+2。请回答以下问题：

- (1) 该计算机存储器按字节编址还是按字编址？该条件转移指令向后（反向）最多可跳转多少条指令？
- (2) 某条件转移指令的地址为200CH，指令内容如图5.36所示，若该指令执行时CF=0、ZF=0、NF=1，则该指令执行后PC的值是多少？若该指令执行时CF=1、ZF=0、NF=0，则该指令执行后PC的值又是多少？请给出计算过程。
- (3) 实现“无符号数比较小于等于时转移”功能的指令中，C、Z、N应各是多少？



图5.36 某条件转移指令

• 答：

— (1) 该计算机存储器按字节编址还是按字编址？该条件转移指令向后（反向）最多可跳转多少条指令？

— 答：

- 因为：转移执行时，转移目标地址为 $(PC)+2+OFFSET \times 2$ ；顺序执行时，下一条指令的地址为 $(PC)+2$ ；并且该计算机采用16位定长指令格式；因此，该计算机存储器是按字节编址（因为是按字节编址，因此转移目标地址中有红色的2）。
- 因为：转移目标地址为 $(PC)+2+OFFSET \times 2$ ， $OFFSET=8$ 位补码，因此转移指令的范围为： $-128 \times 2 \sim +127 \times 2$ ，即向后跳转128条指令，向前跳转127条指令。

— (2) 某条件转移指令的地址为200CH，指令内容如图5.36所示，若该指令执行时 $CF=0$ 、 $ZF=0$ 、 $NF=1$ ，则该指令执行后PC的值是多少？若该指令执行时 $CF=1$ 、 $ZF=0$ 、 $NF=0$ ，则该指令执行后PC的值又是多少？请给出计算过程。

— 答：

- 图5.36的指令中： $Z=1$ 、 $N=1$ ，表示要检测ZF位和NF位。
- 该指令执行时 $CF=0$ 、 $ZF=0$ 、 $NF=1$ ，即ZF和NF中有1个检测位为1（ $NF=1$ ），要转移，因此 $PC=(PC)+2+OFFSET \times 2$ ； $OFFSET=1110011=E3$ ，符号扩展后=FFE3， $PC=200CH+2+FFE3 \times 2=200EH+FFC6H$ ；即该指令执行后PC的值是1FD4H
- 该指令执行时 $CF=1$ 、 $ZF=0$ 、 $NF=0$ ，ZF和NF中没有1个检测位为1，不转移，因此 $PC=(PC)+2=200CH+2=200EH$ ；即该指令执行后PC的值是200EH

— (3) 实现“无符号数比较小于等于时转移”功能的指令中，C、Z、N应各是多少？

— 答：

- $A < B$ ，则A-B有借位，即 $CF=1$ ； $A=B$ ，则 $A-B=0$ ，即 $ZF=1$ ；因此，指令中的 $C=1$ 、 $Z=1$ 、 $N=0$ 。
- 无符号数，说明 $NF=0$ ，所以， $C=1$ ， $Z=0$ ， $NF=0$

**Thanks**