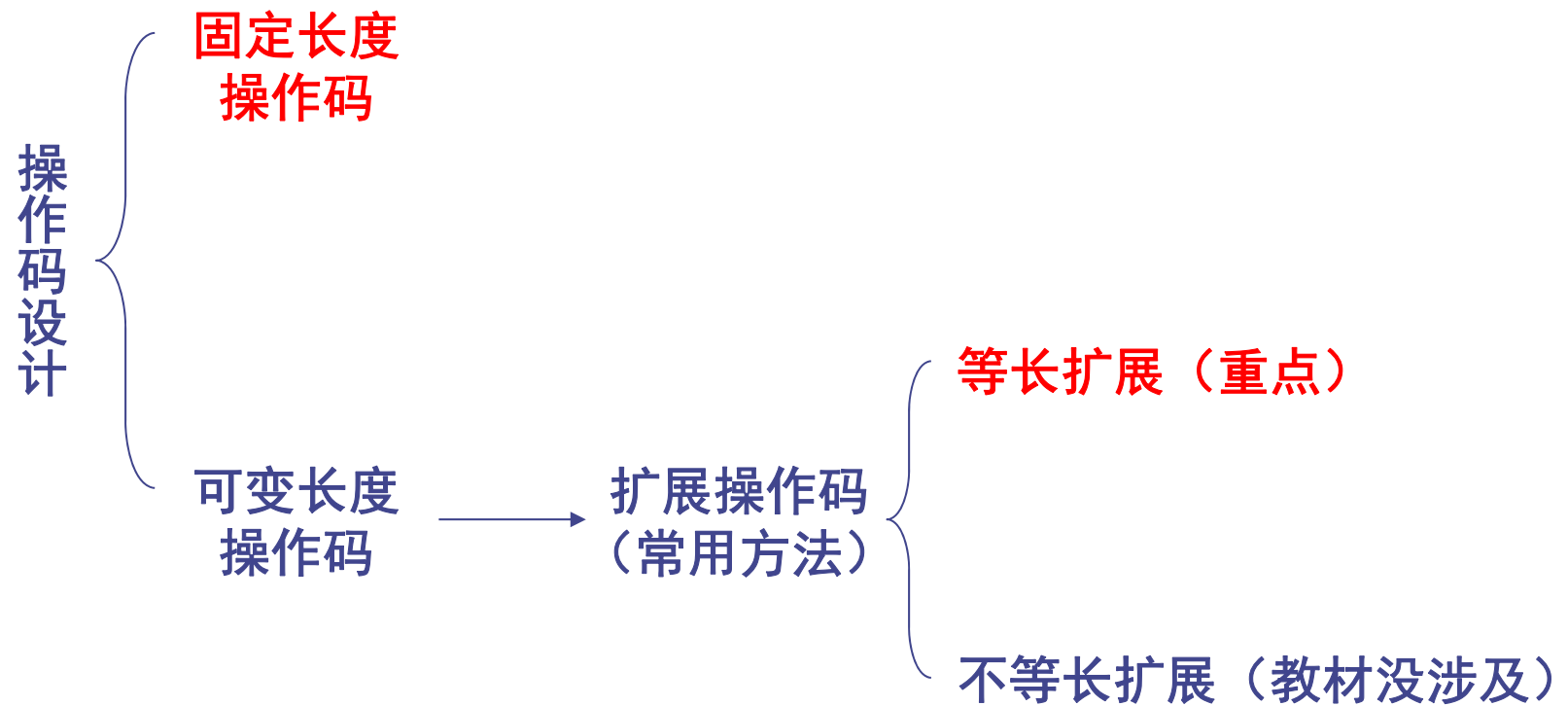


第7章 指令系统

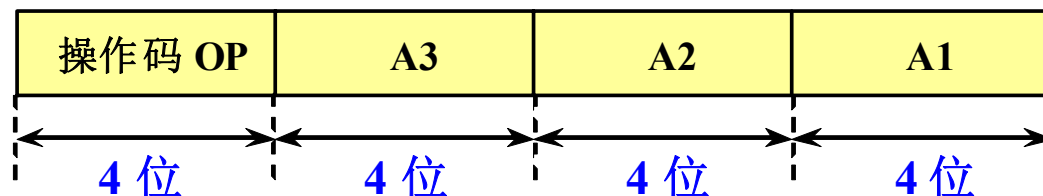
--习题与作业

王冬青
济事楼410
同济大学软件学院



举一个例子：

- 某指令系统中，指令长度为16位，基本操作码为4位，具有3个4位的地址字段。
- 采用固定长度操作码，该指令系统最多只能有16条指令。
- 指令格式



➤ 在指令字长不变的情况下，希望有61条指令。

其中：

15条三地址指令

15条二地址指令

15条一地址指令

16条零地址指令

- 采用等长扩展，操作码分别为4位、8位、12位、16位。

0000	A3	A2	A1	} 15 条三地址指令，操作码为 4 位
0001	A3	A2	A1	
⋮		⋮		
1110	A3	A2	A1	
<hr/>				
1111	0000	A2	A1	} 15 条二地址指令，操作码为 8 位
1111	0001	A2	A1	
⋮		⋮		
1111	1110	A2	A1	
<hr/>				
1111	1111	0000	A1	} 15 条一地址指令，操作码为 12 位
1111	1111	0001	A1	
⋮		⋮		
1111	1111	1110	A1	
<hr/>				
1111	1111	1111	0000	} 16 条零地址指令，操作码为 16 位
1111	1111	1111	0001	
⋮		⋮		
1111	1111	1111	1111	

➤ 采用另一种扩展方法，共76条。

其中：

15条三地址指令

14条二地址指令

31条一地址指令

16条零地址指令

- 采用新的扩展方法，操作码分别为4位、8位、12位、16位。

0000	A3	A2	A1	} 15 条三地址指令，操作码为 4 位
0001	A3	A2	A1	
⋮		⋮		
1110	A3	A2	A1	
<hr/>				
1111	0000	A2	A1	} 14 条二地址指令，操作码为 8 位
1111	0001	A2	A1	
⋮		⋮		
1111	1101	A2	A1	
<hr/>				
1111	1110	0000	A1	} 31 条一地址指令，操作码为 12 位
⋮		⋮		
1111	1110	1111	A1	
1111	1111	0000	A1	
⋮		⋮		
1111	1111	1110	A1	
<hr/>				
1111	1111	1111	0000	} 16 条零地址指令，操作码为 16 位
1111	1111	1111	0001	
⋮		⋮		
1111	1111	1111	1111	

例1： 某计算机指令字长16位,地址码6位,指令有一地址和二地址两种格式,设共有N条($N < 16$)二地址指令, 试问一地址指令最多可以有多少条?

解： 二地址指令的结构是 (4位操作码OP) (6位地址码A1) (6位地址码A2)
一地址指令的结构是 (10位操作码OP) (6位地址码A)
如果全做二地址指令 (一地址指令为0条), 共16条二地址指令。
每少一条二地址指令, 则多 2^6 条一地址指令, 所以一地址指令最多有
 $(16-N) * 2^6$ 条。

例2： 设机器字长12位，操作码字段3位，地址码字段每段3位，给出一种设计方案，使得三地址指令4条，二地址指令8条，一地址指令8条。零地址指令最多能够有多少条？

解： 三地址 $N_3 = 4$ 条， 二地址 $N_2=8$ 条， 一地址 $N_1=8$ 条
零地址 $N_0 \leq ((2^3-N_3)*2^3-N_2)*2^3-N_1)*2^3 = 1472$

计算机组成原理

例3：某计算机按字节编址，指令字长固定且只有两种指令格式，其中三地址指令**29**条，二地址指令**107**条，每个地址字段为**6**位，则指令字长至少应该是 _____ 位。

(操作码, 地址码, 地址个数)

{ 定长操作码: $29 + 107 = 136 < 2^8 \Rightarrow 8$ 位
 $8 + 6 + 6 + 6 = 26$ 位 $\Rightarrow 32$ bit

扩展操作码: $29 < 2^5$ 5 位, $32 - 29 = 3$

$2^6 = 64 \times 3 \Rightarrow 107$ 5 6 6 6 = 23 \Rightarrow 24 bit

按字节编址

作业:

7.11、7.12、7.14

7.15、7.16

某计算机字长为16位，运算器为16位，有16个16位通用寄存器，8种寻址方式，主存容量为64K字。指令中地址码由寻址方式字段和寄存器字段组成，采用单字长指令，则该计算机最多可构成_(82)_条单操作数指令；寄存器间接寻址的范围为_(83)_K字。

答案解析

[解析] 单操作数的指令长度为16位，其中，寻址方式为3位，16个通用寄存器占4位，其余9位用来构成指令操作码，共有 $2^9=512$ 种。所以，最多可构成512条指令。

第1空的正确答案为选项B。

因为通用寄存器是16位的，所以，寄存器间接寻址的范围为 $2^{16}/1024=64\text{K}$ 。第2空的正确答案为选项C。

2.11 某计算机字长**16**位，主存按字编址，采用单字长单地址指令格式，其格式如下所示：

6-bit OP-Code	2-bit X	8-bit D
------------------	------------	------------

OP-Code：操作码。**D**：形式地址。

X：寻址方式码，**X=00**：直接寻址；

X=01：用变址寄存器**X1**变址；

X=10：用变址寄存器**X2**变址；

X=11：相对寻址；

若执行指令时，机器状态如下：

(PC) = 1548H, (X1) = 036AH, (X2) = 46B2H

请分别确定下列指令的有效地址**EA**。

① **3056H** ② **42A0H** ③ **1347H** ④ **4598H** ⑤ **67CEH**

□ 题解：

- X**
- 也应该进行相应的符号位扩展
这里其实是做了零扩展
- ①指令码=0011 00 0 0 0101 0110, 直接寻址
EA=D=0101 0110B = 0056H
- ②指令码=0100 00 1 0 1010 0000, 用变址寄存器X2变址
EA= (X2) +D = 46B2H + A0H = 4752H
46B2H+FFA0H=4652H
46B2H- 0060H=4652H
- ③指令码=0001 00 1 1 0100 0111, 相对寻址
EA= (PC) +D = 1548H + 47H=158FH
1548H+0047H=158FH
- ④指令码=0100 01 0 1 1001 1000, 用变址寄存器X1变址
EA= (X1) +D = 036AH + 98H=0402H
036AH+FF98H=0302H
036AH- 0068H=0302H
- ⑤指令码=0110 01 1 1 1100 1110, 相对寻址
EA= (PC) +D = 1548H + FFCEH=1516H
1548H- 0032H=1516H
- ↑
这里做了符号位扩展

- 2.13 某计算机字长16位，主存按字编址，采用单字长单地址指令格式，指令各字段定义如下：

15	12	11	9	8	6	5	0
4-bit				3-bit		6-bit	
OP-Code				M		Rn	
						A	

其中，**OP-Code**为操作码，**M**为寻址方式码，**Rn**为通用寄存器编号，**A**为形式地址。寻址方式码定义如下：

M	寻址方式	有效地址表达式
000B	一次间接	$EA = (A)$
001B	寄存器间接	$EA = (Rn)$
010B	变址	$EA = (Rn) + A, Rn \leftarrow (Rn) + I$
011B	相对	$EA = (PC) + A$

8种组合中只用了4种

注：有效地址表达式中 **(X)** 表示存储器地址**X**或寄存器**X**的内容；指令中**Rn**字段和**A**字段是否使用视寻址方式而定；位移量用补码表示。

- ❑ 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)上表中各种寻址方式的寻址范围多大（不包括相对寻址）？相对寻址的浮动范围多大？
- ❑ (3)设开始取指令时，对应寄存器和主存相关单元的内容如下图，图中的数字均为十六进制表示，请写出指令**0627H**和**3559H**的操作数各为多少？分别单独执行这两条指令后相关寄存器的内容各是多少？**设PC更新是在取指后做。**

	地址	主 存
PC		
	19H	0100H
R0		
	27H	4000H
	400H	1000H
R5		
	401H	3559H
	419H	0123H
R7		
	41AH	0627H
	1FE7H	1234H
	1FE8H	5678H

□ 题解:

- 1、该指令系统最多可有 $2^4=16$ 条指令，该计算机最多有 $2^3=8$ 个通用寄存器
- 2、一次间接寻址范围= $2^{16}=64K$ 字、
寄存器间接寻址范围= $2^{16}=64K$ 字
变址寻址范围= $2^{16}=64K$ 字
相对寻址的浮动范围= $-32\sim+31$

15	12	11	9	8	6	5	0
4-bit				3-bit		3-bit	
OP-Code				M		Rn	
						6-bit	
						A	

- 3、 a、 指令0627H展开: 0000 011 0 00 10 0111B
 OP=0000B, M=011B=相对寻址,
 Rn=000B (无用), A=10 0111B (负数补码)
 EA=(PC)+A=2001H+FFE7H=1FE8H
 (取指后(PC)+1, 且A符号扩展)
 操作数a= (EA) = (1FE8H) =5678H
 指令执行后: (PC) = 1FE9H

011B	相对	EA= (PC) + A
------	----	--------------

15	12	11	9	8	6	5	0
4-bit				3-bit		3-bit	
OP-Code				M		Rn	
				6-bit			
				A			

○b、指令3559H展开：0011 010 1 01 01 1001B

OP=0011B, M=010B=变址寻址,

010B	变址	$EA = (Rn) + A, Rn \leftarrow (Rn) + 1$
------	----	---

Rn=101B (R5), A=01 1001B (正数补码)

EA=(R5)+A=0400H+0019H=0419H (A符号扩展)

操作数b= (EA) = (0419H) =0123H

(R5) = (R5) +1=0401H

指令执行后：(PC) =2001H, (R5) =0401H

□ □

作业：

7.10、7.13、7.17

load-store结构:

例如 $c = a + b;$

a	10
b	20
c	

内存

两种处理方式的对比:

RISC

```
Load  R1,[a]
Load  R2,[b]
Add   R3,R1,R2 ;只支持寄存器间的运算
Store R3,[c]
```

CISC

```
MOV  EAX,[a]
ADD  EAX,[b] ;支持寄存器和内存直接运算
MOV  [c],EAX
```