



计算机组成与系统结构

第四章 指令系统 (2)

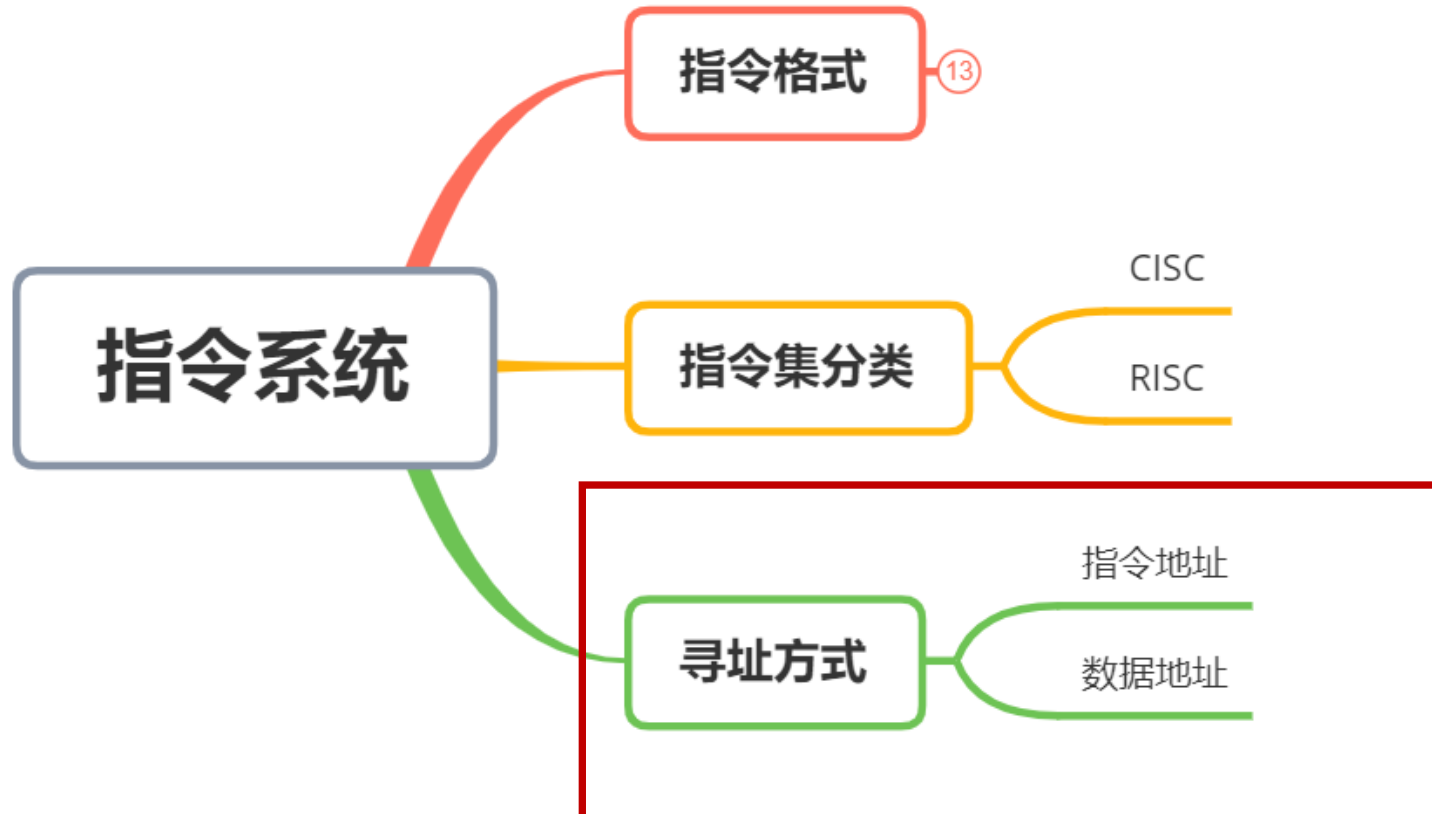
吕昕晨

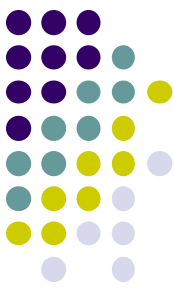
lvxinchen@bupt.edu.cn

网络空间安全学院



第四章——指令系统





指令系统设计

- 指令系统要求

- 完备性
- 有效性
- 规整性
- 兼容性

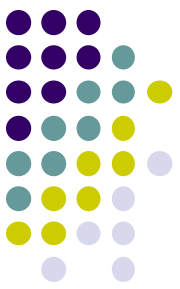
- 问题1：规定指令格式，支持更多指令功能

- 操作码、操作数、寄存器号、存储单元地址.....

- 问题2：寻址方式

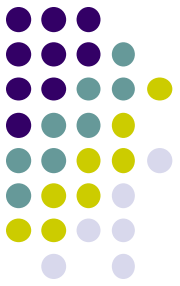
- 如何给出下一条指令地址（指令寻址）
- 数据地址范围受地址码限制（8位地址码→256个字节）
- 增加地址码，影响指令系统效率
- 设计寻址方式，支持更大范围寻址（数据寻址）





指令系统设计——寻址方式（问题2）

- 问题2.1：指令寻址方式设计
 - 指令执行方式：顺序、循环、跳转.....
 - 指令寻址需满足程序执行需求
- 问题2.2：数据寻址方式设计
 - 寻址要求分析：明确寻址需支持的方式情况，明确具体需求
 - 如何设计满足要求的寻址方式：根据需求，设计多种寻址方式，多种寻址方式综合设计

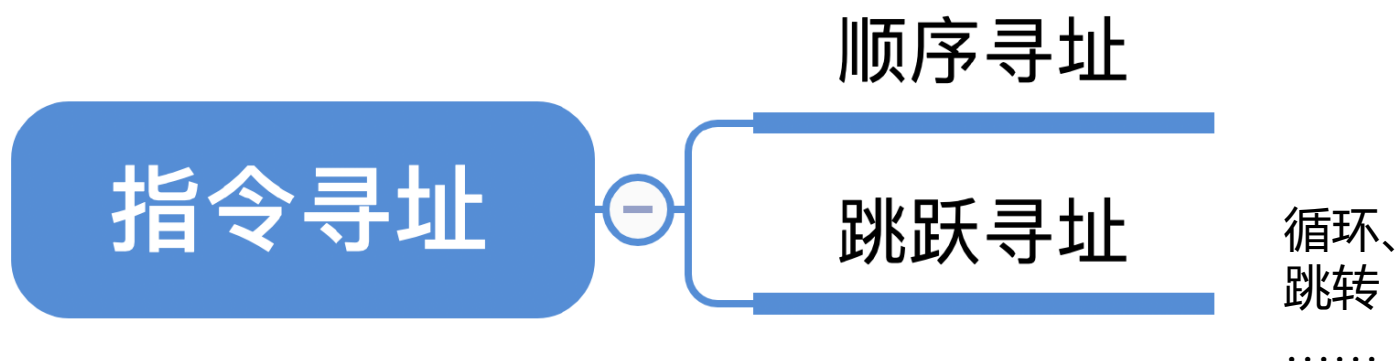


第四章 指令系统

- 指令寻址
- 数据寻址
 - 操作数类型与分类
 - 常见数据寻址方式



指令寻址方式



指令寻址：如何确定下一条指令的存放地址？

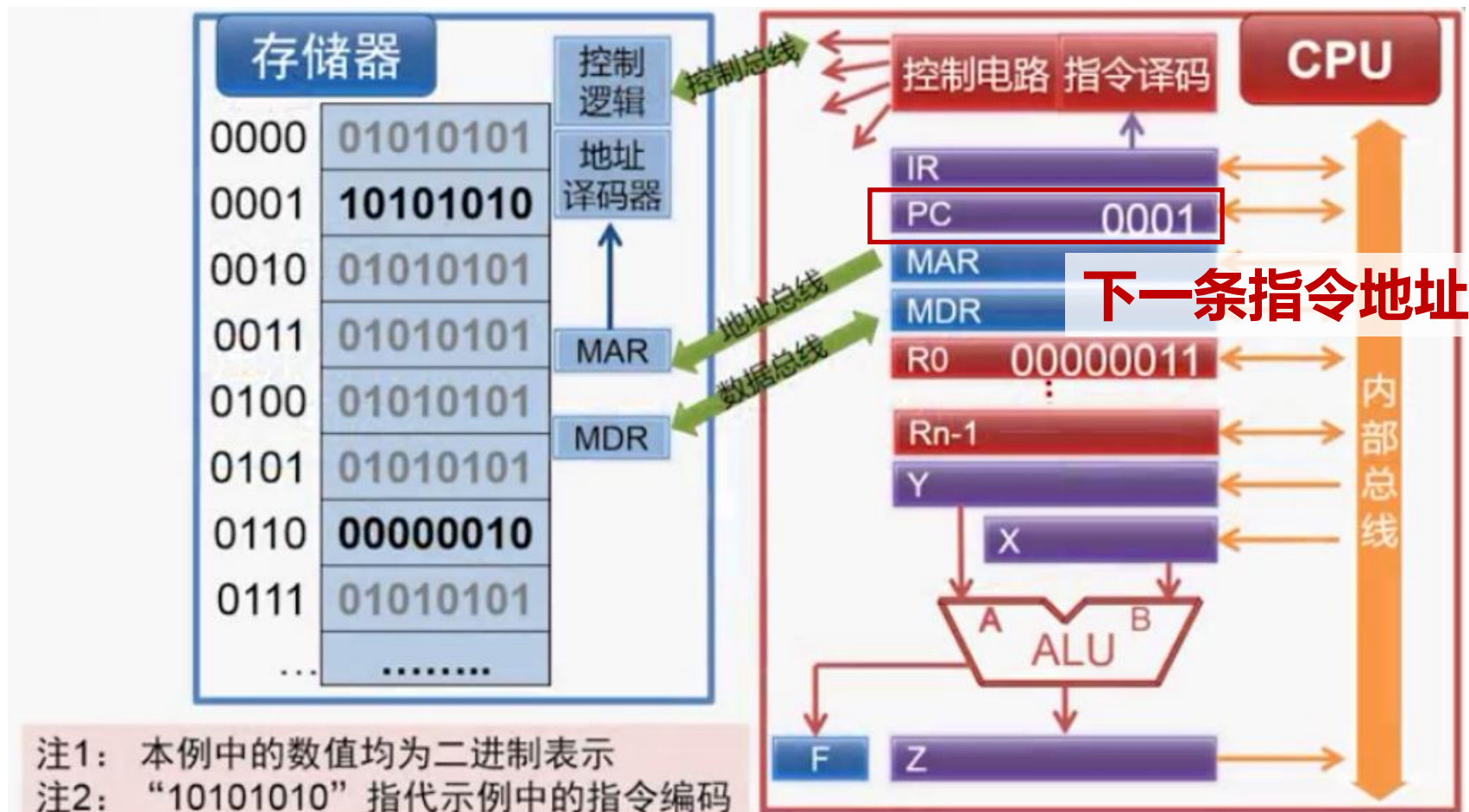
一条指令的结构

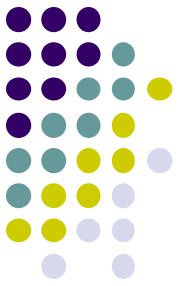
操作码 (OP)

地址码 (可能有多个)



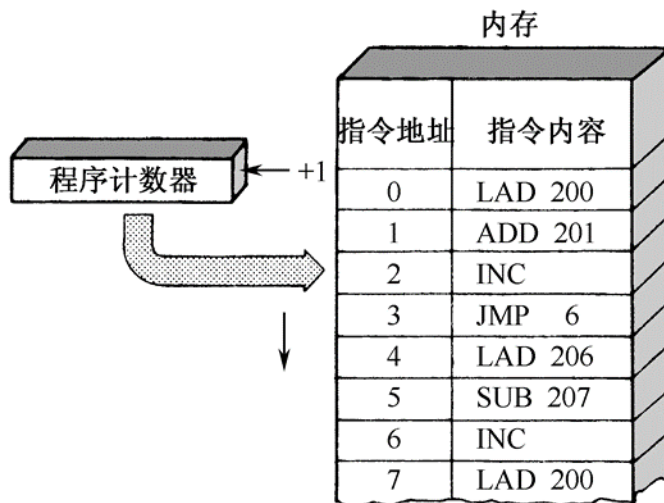
指令地址—PC寄存器



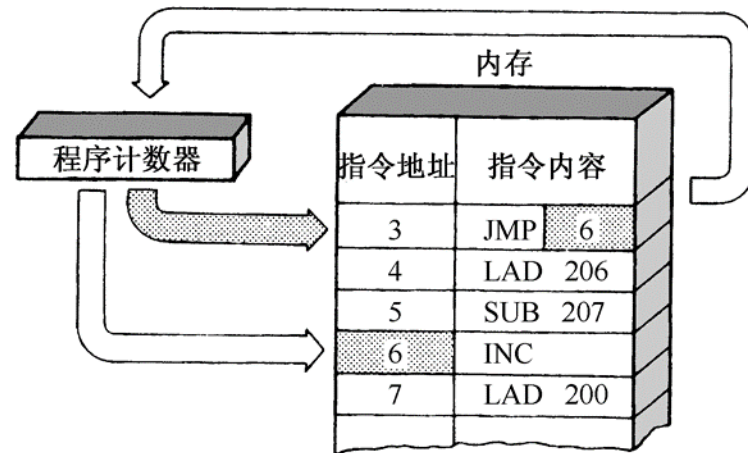


指令的寻址：EA=(PC)

- 顺序方式
 - $PC=PC+1$
- 跳跃方式
 - 无条件、条件转移指令
 - 例如，JMP语句



(a) 指令的顺序寻址方式



(b) 指令的跳跃寻址方式



顺序寻址方式

- PC寄存器具有自增功能
 - 每译码一条语句，PC寄存器**自动**指向下指令地址
 - 统一表示： $PC=PC+ "1"$

顺序寻址

$$(PC) + "1" \rightarrow PC$$

这里的1 理解为1个指令字长，实际加的值会因指令长度、编址方式而不同

- 思考：
 - 问题1：如果指令为双字长指令？
 - $PC=PC+2$ （每个指令为2个字长，按字寻址）
 - 问题2：如果指令字长不固定？
 - $PC=PC+n$ （n为目前指令的字长，译码后给出）



跳跃寻址方式

- 转移指令给出下一条语句执行地址
- 示例：JMP 7（无条件跳转）
 - 情况1：跳转语句后接绝对地址

- 执行语句后：PC=7

JMP：无条件转移
把PC中的内容改成7

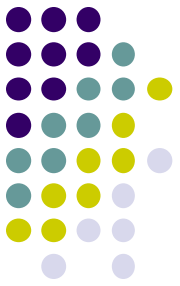
无条件转移指令，
类似C语言的 goto

- 情况2：跳转语句采用相对寻址（给出偏移量）
 - 执行语句后：PC=PC + 7 + “1”



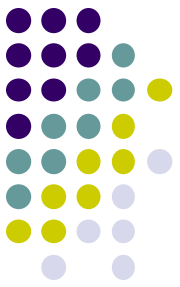
某计算机系统字长为32位（按字节寻址），当前PC=1000H，将执行JMP 8指令（采用相对寻址方式，单字长指令），请问下一条语句执行地址为

- ☐ A 0008H
- ☐ B 1008H
- ☐ C 100AH
- ☒ D 100CH



第四章 指令系统

- 指令寻址
- 数据寻址
 - 操作数类型与分类
 - 常见数据寻址方式



操作数来源分类

- 在指令执行过程中，操作数的来源一般有三个
 - **直接数**：由指令中的地址码部分直接给出操作数
 - 虽然简便快捷，但是操作数是固定不变的
 - **寄存器**：将操作数存放在CPU内的通用数据寄存器中
 - 很快获取操作数，但是可以存储的操作数的数量有限
 - **内存寻址**：将操作数存放在内存的数据区中
 - 需支持大范围寻址区间
- **寻址方式要求**
 - 支持直接数、寄存器、内存寻址等多种方式
 - 内存寻址方式扩大**寻址范围**（指令长度受限）

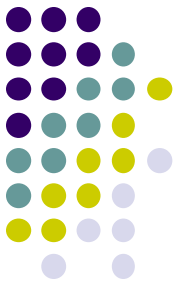


寻址核心内容 ($A \rightarrow EA$)

- 把操作数的形式地址变换为有效地址
 - 有效地址EA：在指令中直接给出操作数的实际访存地址
 - 形式地址A：在指令执行时，将形式地址依据某种方式变换为有效地址再取操作数 ($A \rightarrow EA$)
- 单地址指令格式
 - 操作码OP、变址X、间址I、形式地址A

操作码 OP	变址 X	间址 I	形式地址 A
-----------	---------	---------	-----------

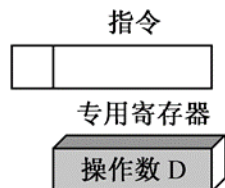
- 形式地址A
- 变址X、间址I：寻址方式特征位
 - 若无变址、间址要求，形式地址 = 有效地址
 - 否则需要进行变换



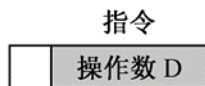
第四章 指令系统

- 指令寻址
- 数据寻址
 - 操作数类型与分类
 - 常见数据寻址方式（重点）

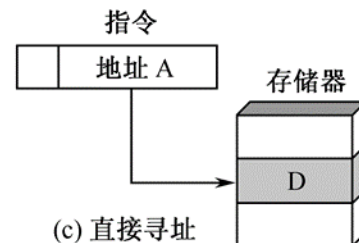
基本寻址方式—访存效率对比



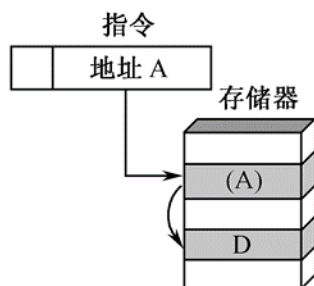
(a) 隐含寻址



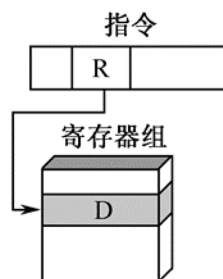
(b) 立即寻址



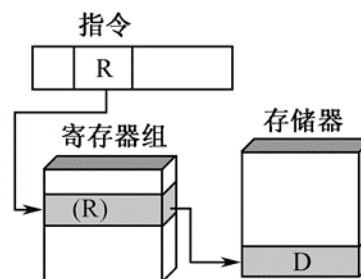
(c) 直接寻址



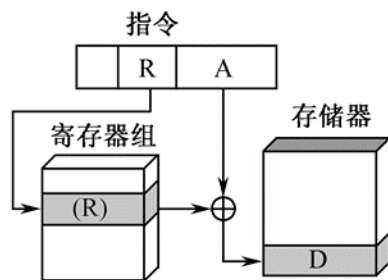
(d) 间接寻址



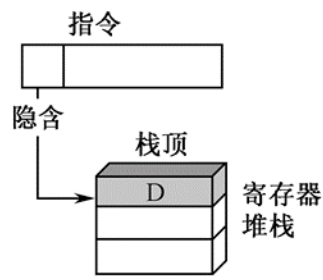
(e) 寄存器寻址



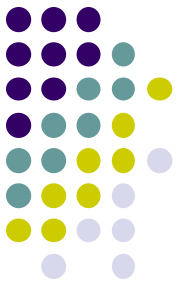
(f) 寄存器间接寻址



(g) 偏移寻址

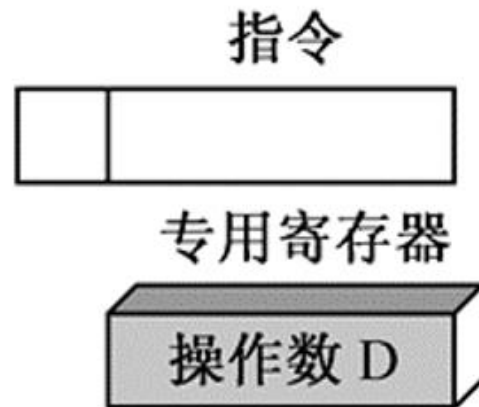


(h) 堆栈寻址

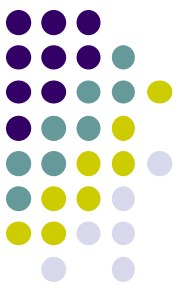


1. 隐含寻址

- 指令中隐含着操作数的地址
- 如某些运算，隐含了累加器AC作为源和目的寄存器
 - 如8086汇编中的STC指令（CLC）
 - 功能：设置标志寄存器的CF为1



(a) 隐含寻址



2. 立即寻址（立即数）

- 指令中在操作码字段后面的部分不是通常意义上的操作数地址，而是操作数本身
- 数据就包含在指令中，只要取出指令，就取出了立即使用的操作数
- 操作数被称为**立即数**
- 指令格式：操作码OP 操作数D
- 例如，MOV AH, 80H (**立即寻址**)

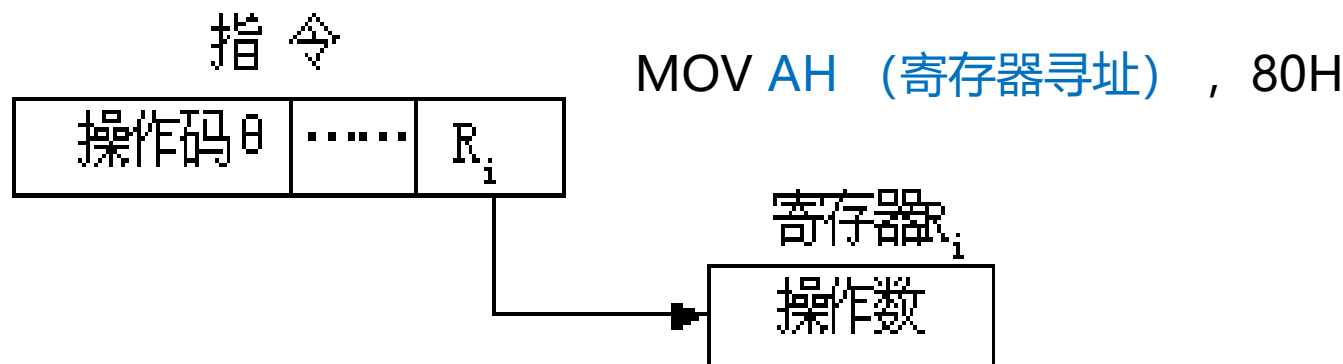


(b) 立即寻址



3. 寄存器寻址

- 在指令的地址码部分给出CPU内某一通用寄存器的编号，指令的操作数存放在相应的寄存器中，即 $EA=R_i$

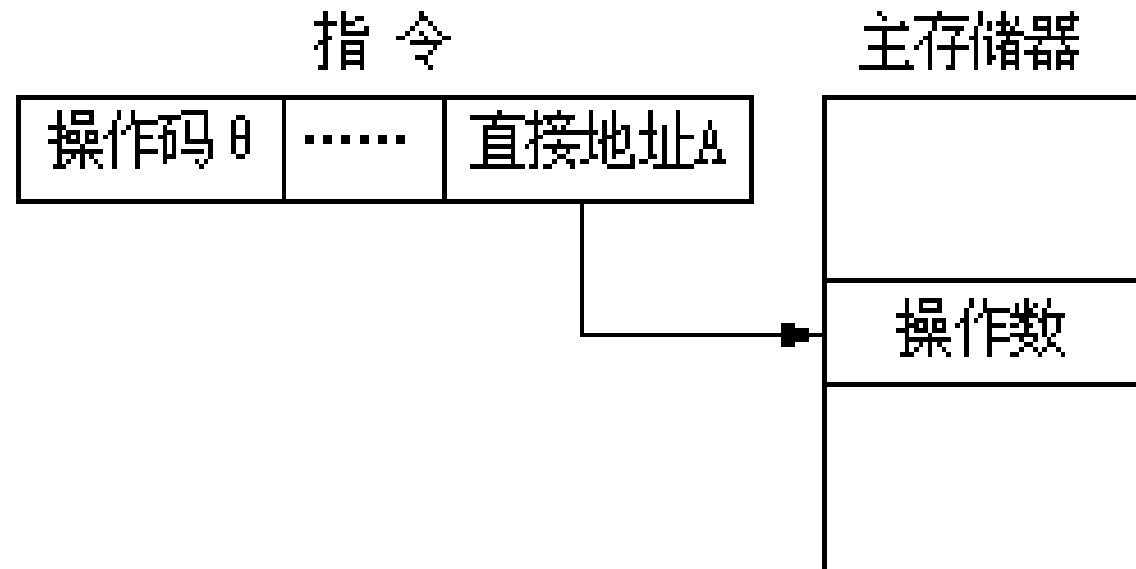


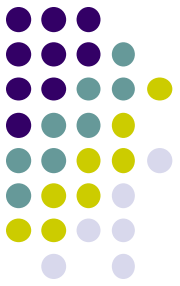
- 优点
 - 由于寄存器在CPU的内部，指令在执行时从寄存器中取操作数比访问主存要快得多
 - 由于寄存器的数量较少，因此寄存器编号所占位数也较少，从而可以有效减少指令的地址码字段的长度
- 缺点：寄存器数量少，不够灵活



4. 直接寻址

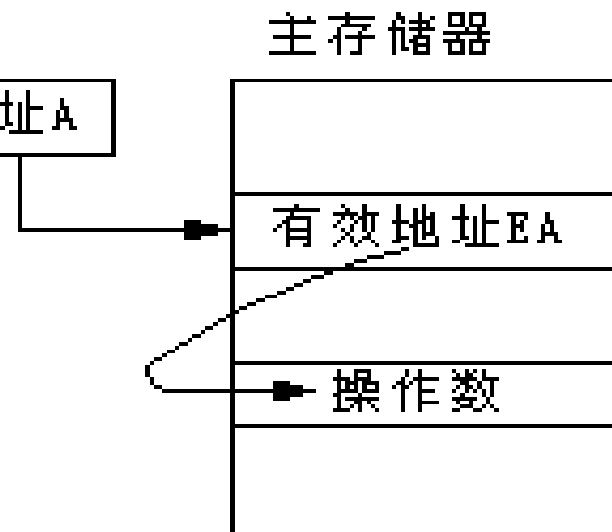
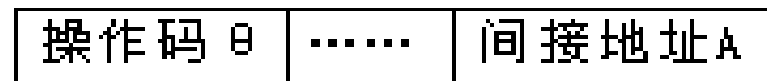
- 指令中地址码字段给出的地址A就是操作数的有效地址EA(Effective Address), 即 $EA = A$
 - 例如, ADD R0, [6]
 - 优点: 寻址简单
 - 缺点: 寻址范围受限

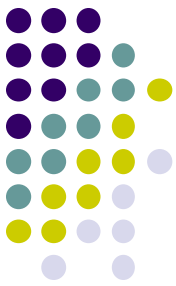




5. 间接寻址

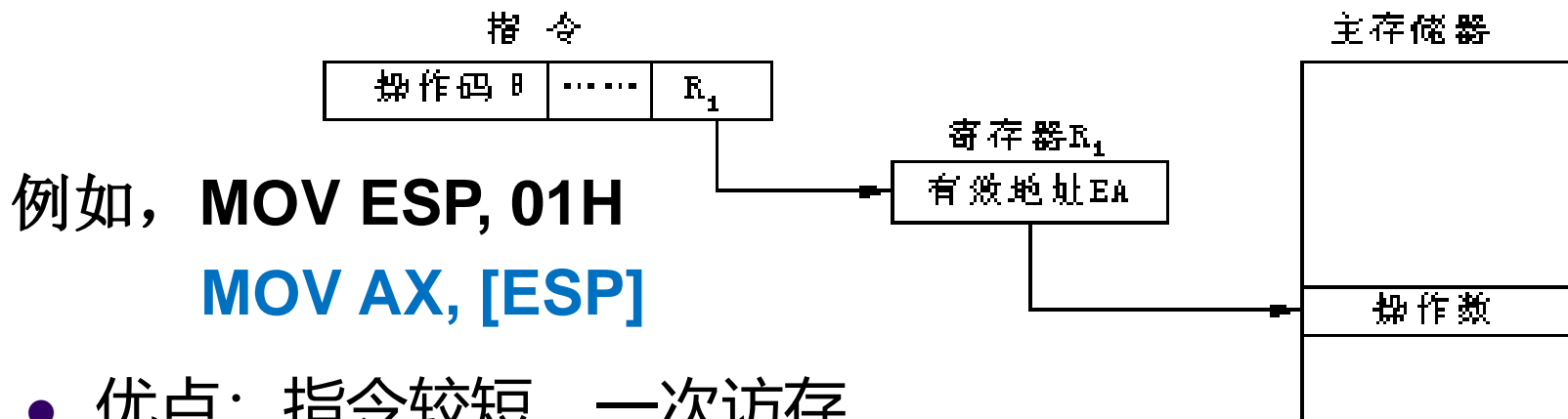
- 间接寻址是相对与直接寻址的方式
- 指令形式地址A不是操作数D的地址，而是给出存放操作数D地址的内存地址
- 即操作数地址的地址
- 缺点：两次访存，影响执行速度
- 记作： $EA = (A)$ 指令



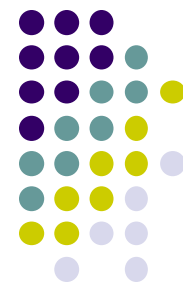


6. 寄存器间接寻址

- 寄存器间接寻址：克服间接寻址中多次访存的缺点
- 将操作数放在主存储器中，而操作数的地址放在某一通用寄存器中
- 指令的地址码部分给出该通用寄存器的编号
- 记作：EA=(Ri)



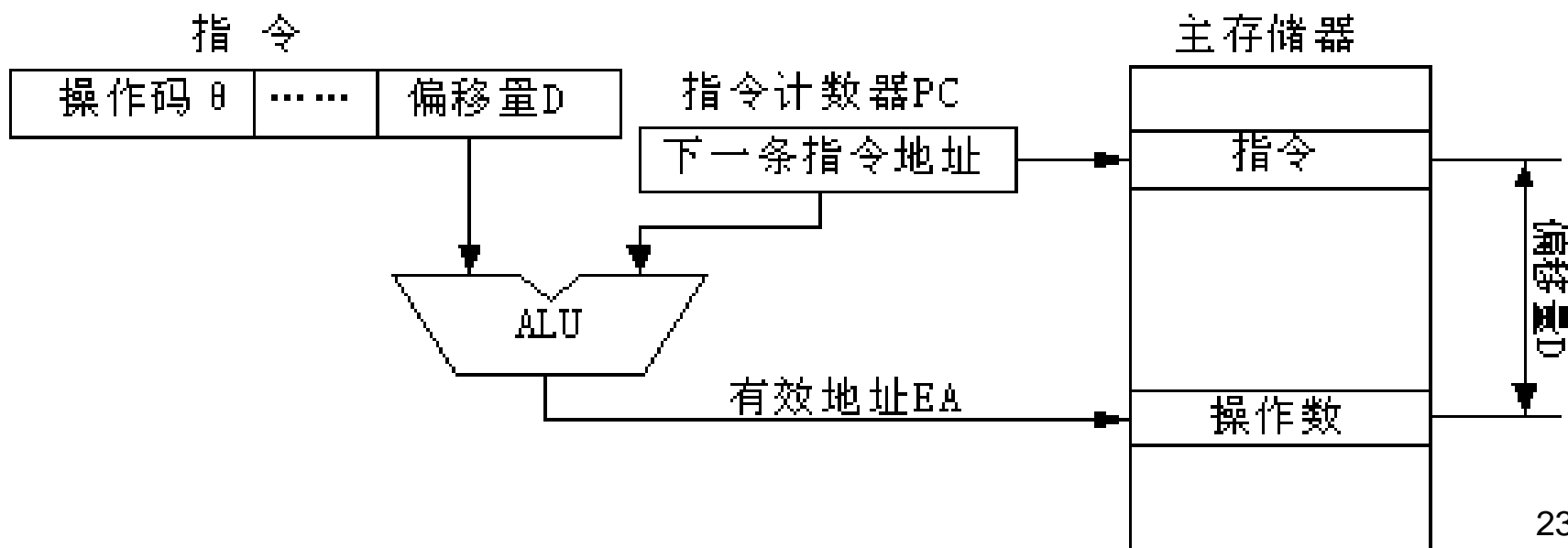
- 优点：指令较短，一次访存
- 是目前在计算机中使用较为广泛的一种寻址方式

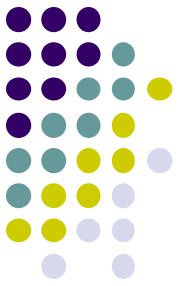


7. 偏移寻址（1）——相对寻址

- 相对寻址方式

- 示例：由程序计数器PC提供基准地址，而指令的地址码给出相对的位移量A，相加后作为操作数的有效地址
- 记作： $EA = (PC) + A$

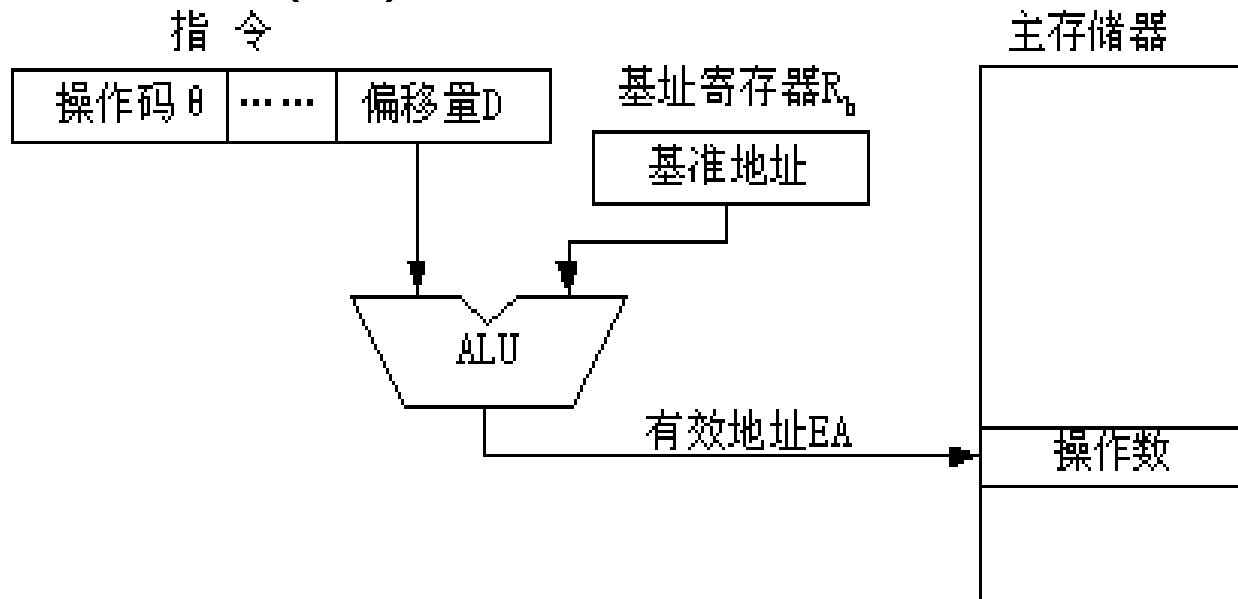


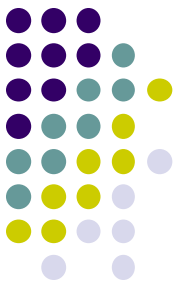


7. 偏移寻址 (2) — 基址寻址

- 基址寻址方式

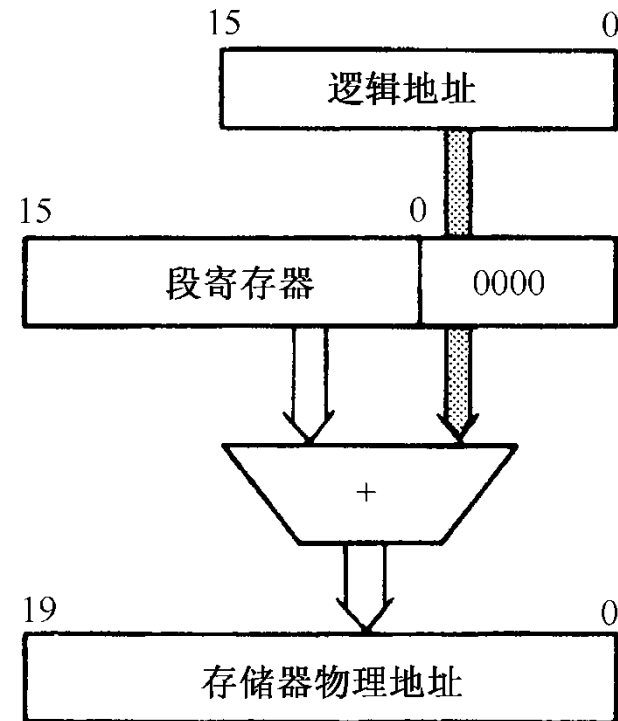
- 参考值发生改变：基址寄存器
- 优点：基址寄存器的位数可以设置得很长，从而可以在较大的存储空间中寻址
- 记作： $EA = (Rb) + A$





7. 偏移寻址(2)—基址实例(段寻址)

- 本质：基址寻址
- 例如，8086-微机系统
 - 逻辑地址：16位
 - 物理地址：20位
 - 地址扩展：16位→20位
 - 方式：段寻址
 - 段寄存器：CS、DS等
 - 物理地址形成方式：
 - 段寄存器左移4位
 - 与逻辑地址相加

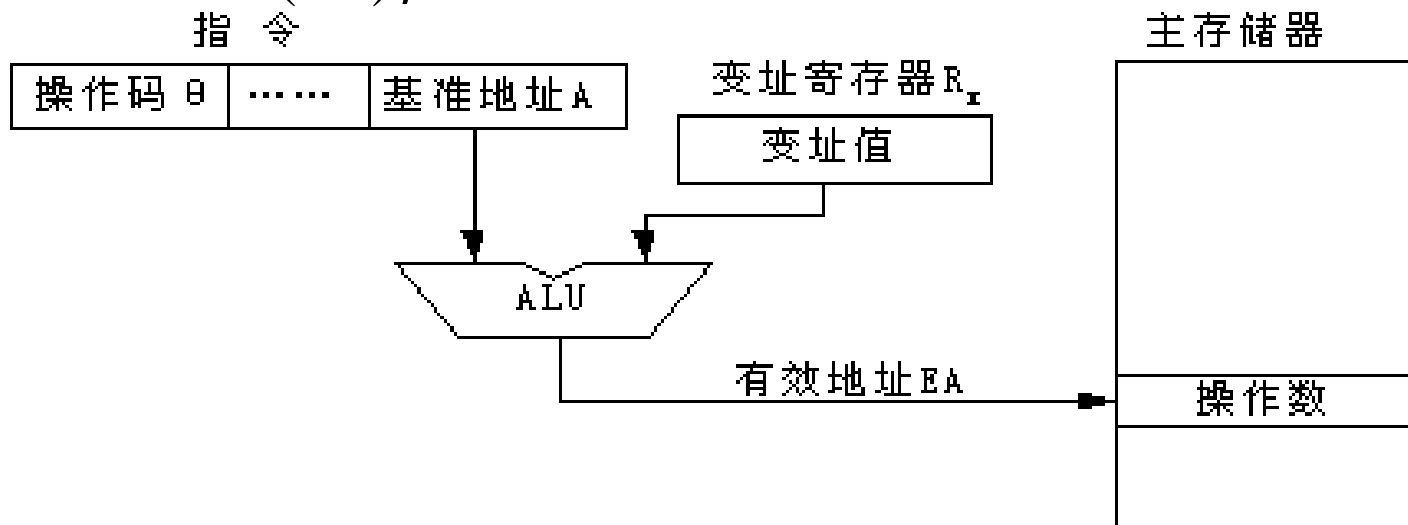




7. 偏移寻址 (3) —变址寻址

● 变址寻址

- 参考值：基准地址A（与基址寻址相反）
- 给出的基准地址A与CPU内某特定的变址寄存器R_x中的内容相加，以形成操作数的有效地址
- 例如，对数组进行相加
 - $EA = A + (R_x), R_x = R_x + 1$





7. 偏移寻址——总结

偏移寻址

相对寻址, $EA = (PC) + A$

转移指令

基址寻址, $EA = (Rb) + A$

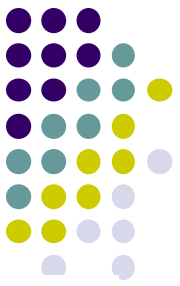
面向系统

扩大寻址空间

变址寻址, $EA = (RX) + A$

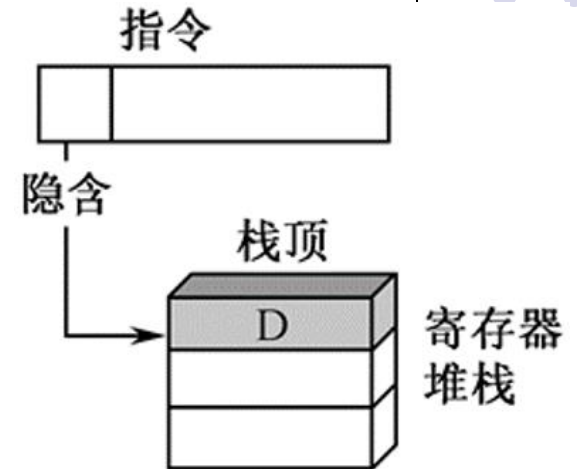
面向用户

A不变, RX自增, 循环语句

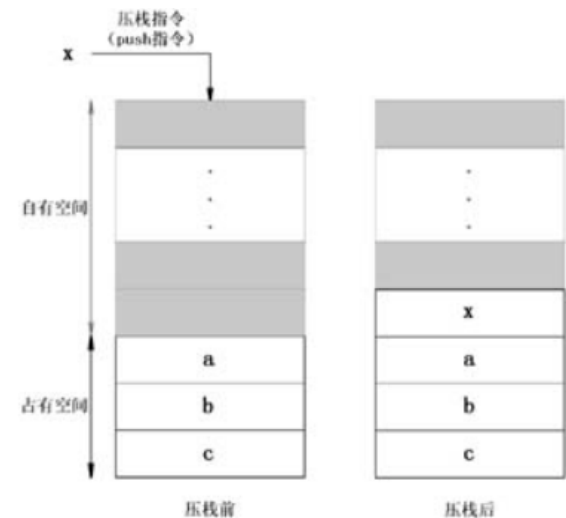


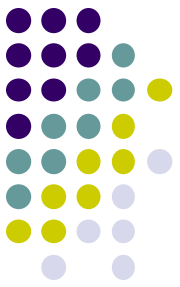
8. 堆栈寻址

- 分类
 - 寄存器堆栈、存储器堆栈两种形式
- 特征：隐含用堆栈指针作为地址
- 记作： $EA=(SS)+SP$
- 存储原理
 - 先进后出方式
 - 保存现场，函数调用等
- 操作
 - 压栈：PUSH, $SP-1$
 - 出栈：POP, $SP+1$



(h) 堆栈寻址





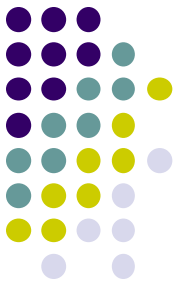
寻址方式组合

- 不同的指令系统采用不同的方式指定寻址方式
 - 有些指令固定使用某种寻址方式
 - 有些指令则允许使用多种寻址方式
 - 在指令中加入寻址方式字段指明
 - 对不同的寻址方式分配不同的操作码而把它们看作是不同的指令
 - 有些指令系统会把常见的寻址方式组合起来，构成更复杂的符合寻址方式
 - 例如，间接寻址方式
 - 特征位I=0（直接寻址）
 - 特征位I=1（间接寻址）

操作码	I	A
-----	---	---

MOV ESP, 01H

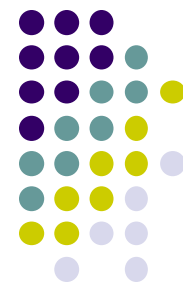
MOV AX, [ESP]



寻址方式总结（重要）

寻址方式	有效地址	访存次数	示例
隐含寻址	语句默认指定	0	STC、CLC
立即寻址	指令包含操作数	0	MOV AH, 80H
寄存器寻址	$EA=Ri$	0	MOV AH, 80H
直接寻址	$EA=A$	1	ADD R0, [6]
间接寻址	$EA=(A)$	2	
寄存器间接寻址	$EA=(Ri)$	1	MOV AX,[ESP]
相对寻址	$EA=(PC)+A$	1	转移指令
基址寻址	$EA=(Bx)+A$	1	扩大访存、段寻址
变址寻址	$EA=(Rx)+A$	1	循环语句
堆栈寻址	$EA=(SS)+SP$	1	POP、PUSH

偏移寻址



习题1

[例4] 一种二地址RS型指令的结构如下：

6 位		4 位	1 位	2 位	16 位
OP	—	通用寄存器	I	X	偏移量 D

其中I为间接寻址标志位，X为寻址模式字段，D为偏移量字段。通过I，X，D的组合，说明可构成的寻址方式名称

寻址方式	I	X	有效地址E算法	说明
(1)	0	00	$E=D$	
(2)	0	01	$E=(PC) \pm D$	PC为程序计数器
(3)	0	10	$E=(R_2) \pm D$	R_2 为变址寄存器
(4)	1	11	$E=(R_3)$	
(5)	1	00	$E=(D)$	
(6)	0	11	$E=(R_1) \pm D$	R_1 为基址寄存器

- 1) 直接寻址
- 2) 相对寻址
- 3) 变址寻址
- 4) 寄存器间接寻址
- 5) 间接寻址
- 6) 基址寻址

习题2



处理机指令格式如下

1位		3位	
X	OP	寄存器	地址

其中，寄存器为3位（12位通用寄存器）；最高位X用来指明是否选用变址寻址（X=1，通用寄存器为变址寄存器）；主存最大容量为16384字节（按字节寻址）。

(1) 要求不使用通用寄存器也能访问每个一主存地址，且操作码位数至少支持88条指令，地址码域应分配多少位，OP码域应分配多少位，指令字共多少位？

(2) X=0，指令字中明确使用通用寄存器，且通用寄存器为基址寄存器。给出硬件设计规则，使通用寄存器能访问主存每一个位置。

(3) 假设主存容量扩充到32768字节，硬件结构已确定不变，采用什么方法可完成寻址？

习题2



处理机指令格式如下

1位		3位	
X	OP	寄存器	地址

其中，寄存器为3位（12位通用寄存器）；最高位X用来指明是否选用变址寻址（X=1，通用寄存器为变址寄存器）；主存最大容量为16384字节（按字节寻址）。

（1）要求不使用通用寄存器也能访问每个一主存地址，且支持88条指令，地址码域应分配多少位，OP码域应分配多少位，指令字共多少位

OP：7位 ($2^7=128$)

地址码：14位 ($2^{14}=16384$)

指令字长度 $14+7+3+1=25$ 位

习题2



处理机指令格式如下

1位	7位	3位	14位
X	OP	寄存器	地址

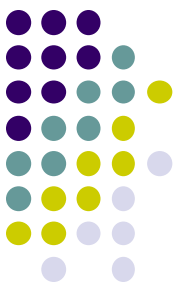
(2) $X=0$ ，指令字中明确使用通用寄存器，且通用寄存器为基址寄存器。给出硬件设计规则，使通用寄存器能访问主存每一个位置。

基址寄存器（12位，左移2位），与形式地址相加

(3) 假设主存容量扩充到32768字节，硬件结构已确定不变，采用什么方法可完成寻址？

15位，采用间接寻址方式

间接寻址	$EA=(A)$
寄存器间接寻址	$EA=(Ri)$
相对寻址	$EA=(PC)+A$
基址寻址	$EA=(Bx)+A$
变址寻址	$EA=(Rx)+A$

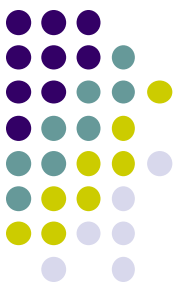


习题3

- 某机双操作数指令占用3字节格式，OP为操作码，S/D表示Reg中存放的是源还是目的操作，M为寻址模式：0为直接寻址；1为基址寻址；2为相对寻址；3为立即寻址。A为偏移量，有符号数，补码表示。通用寄存器有8个，其中R7为基址寄存器。

OP (7)		S/D (1)
M (2)	Reg (3)	R (3)
A (8)		

- 问：（1）该指令格式可定义多少种不同的操作？（2）立即寻址操作数的范围是多大？（3）若基址寄存器为14位，采用基址寻址方式的寻址范围为多少？（4）写出4种寻址方式计算有效地址的表达式。



习题3

- 某机双操作数指令占用3字节格式，OP为操作码，S/D表示Reg中存放的是源还是目的操作，M为寻址模式：0为直接寻址；1为基址寻址；2为相对寻址；3为立即寻址。A为偏移量，有符号数，补码表示。通用寄存器有8个，其中R7为基址寄存器。

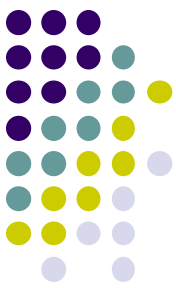
OP (7)		S/D (1)
M (2)	Reg (3)	R (3)
A (8)		

- 问：（1）该指令格式可定义多少种不同的操作？

OP位数给出， $2^7=128$ 种

- （2）立即寻址操作数的范围是多大？

立即寻址操作数为A，8位补码范围：-128~127



习题3

- 某机双操作数指令占用3字节格式，OP为操作码，S/D表示Reg中存放的是源还是目的操作，M为寻址模式：0为直接寻址；1为基址寻址；2为相对寻址；3为立即寻址。A为偏移量，有符号数，补码表示。通用寄存器有8个，其中R7为基址寄存器。

OP (7)		S/D (1)
M (2)	Reg (3)	R (3)
A (8)		

- 问：(3) 若基址寄存器为14位，采用基址寻址方式的寻址范围是多少？ 基址寻址 $EA = (R7) + A$ ， $2^{14} = 16KB$
- (4) 写出4种寻址方式计算有效地址的表达式
直接 $EA = A$ ，基址 $EA = (R7) + A$ ，相对 $EA = (PC) + A$ ，立即A操作数



总结——寻址方式（重点）

寻址方式	有效地址	访存次数	示例
隐含寻址	语句默认指定	0	STC、CLC
立即寻址	指令包含操作数	0	MOV AH, 80H
寄存器寻址	$EA=R_i$	0	MOV AH, 80H
直接寻址	$EA=A$	1	ADD R0, [6]
间接寻址	$EA=(A)$	2	
寄存器间接寻址	$EA=(R_i)$	1	MOV AX,[ESP]
相对寻址	$EA=(PC)+A$	1	转移指令
基址寻址	$EA=(B_x)+A$	1	扩大访存、段寻址
变址寻址	$EA=(R_x)+A$	1	循环语句
堆栈寻址	$EA=(SS)+SP$	1	POP、PUSH

偏移寻址

总结

