河南大學
HENAN UNIVERSITY

计算机组成原理

主讲人：胡 萍

明德，新民

止于至善

第7章指令系统

本章内容

★7.1 机器指令

7.2 操作数类型和操作类型

★7.3 寻址方式

★7.4 指令格式举例

7.5 RISC 技术

7.1 机器指令

- 7.1.1 指令的一般格式
- 7.1.2 指令字长

7.1.1 指令的一般格式

- 指令：用二进制代码表示的结构形式，由操作码和地址码两部分组成。

操作码字段OP	地址码字段A

7.1.1 指令的一般格式

1. 操作码　　指明该指令所要完成的操作

操作码的位数反映了机器的操作种类。

操作码字段为8位，最多包含指令的数目为2^8=256条。

（1） 长度固定

- 特征：所有指令长度均相同。
- 优点：控制简单，速度快，适用于字长较长的计算机。

（2） 长度可变

- 特征：利用操作码扩展技术将操作码扩展到地址码字段，使各类指令的操作码长度不相同；
- 优点：压缩操作码平均长度，适用于字长较短的计算机。

单选题 1分

设置

Henan University

1. 某指令系统有200条指令，对操作码采用固定长度二进制编码。最少需要(　　)位操作码。

B

A 4

B 8

C 16

D 32

提交

扩展操作码技术

操作码的位数随地址数的减少而增加

	4	4	4	4	
	OP	A_1	A_2	A_3	
4 位操作码	0000	A_1	A_2	A_3	15条三地址指令
	0001	A_1	A_2	A_3	
	⋮	⋮	⋮	⋮	
	1110	A_1	A_2	A_3	
8 位操作码	1111	0000	A_2	A_3	16条二地址指令
	1111	0001	A_2	A_3	
	⋮	⋮	⋮	⋮	
	1111	1111	A_2	A_3	

三地址指令操作码每减少一种可多构成2^4种二地址指令

单选题 1分

设置

Henan University

2. 在指令格式中，采用扩展操作码设计方案的目的是(　)。

C

A 减少指令字长度

B 增加指令字长度

C 保持指令字长度不变而增加指令的数量

D 保持指令字长度不变而增加寻址空间

提交

在设计指令系统时，尽量安排使用频度高的指令占用短的操作码，以缩短指令译码时间。

	4	4	4	4	
	OP	A_1	A_2	A_3	
4 位操作码	0000	A_1	A_2	A_3	15条三地址指令
	0001	A_1	A_2	A_3	
	⋮	⋮	⋮	⋮	
	1110	A_1	A_2	A_3	
8 位操作码	1111	0000	A_2	A_3	15条二地址指令
	1111	0001	A_2	A_3	
	⋮	⋮	⋮	⋮	
	1111	1110	A_2	A_3	
12 位操作码	1111	1111	0000	A_3	15条一地址指令
	1111	1111	0001	A_3	
	⋮	⋮	⋮	⋮	
	1111	1111	1110	A_3	
16 位操作码	1111	1111	1111	0000	16条零地址指令
	1111	1111	1111	0001	
	⋮	⋮	⋮	⋮	
	1111	1111	1111	1111	

单选题 1分

设置

Henan University

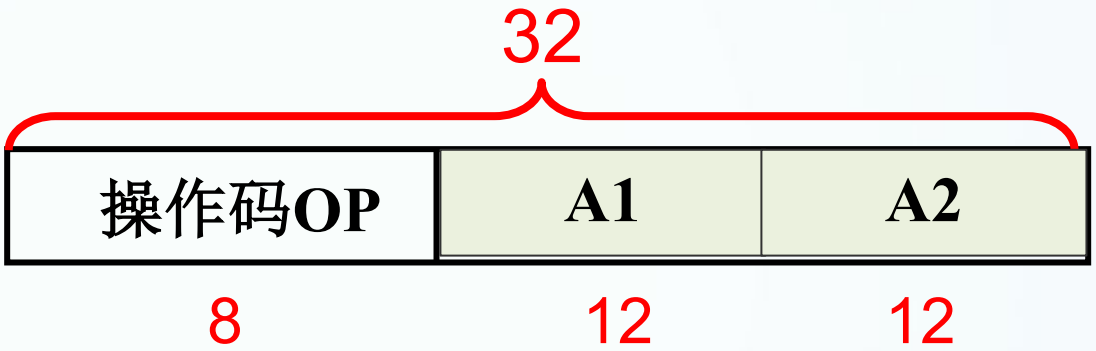
3. 一个计算机系统采用32位单字长指令，每个地址码12位，如果定义了250条二地址指令，那么还可以有(D)条单地址指令。

A 4K

B 8K

C 16K

D 24K



如果没有扩展，则应有256条二地址指令。

可求扩展入口有6个，每个入口可扩展2^{12}。

提交

单选题 1分

设置

Henan University

4. 【2017 机器指令构成】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(A)。

按字节编址说明，存储字长是8的倍数。

5位操作码

题目默认存储字长=机器字长=指令字长

A 24位

B 26位

C 28位

D 32位

提交

2. 地址码（了解）

（1） 四地址

8	6	6	6	6
OP	A_1	A_2	A_3	A_4

A_1 第一操作数地址

A_2 第二操作数地址

A_3 结果的地址

A_4 下一条指令地址

(A_1) OP $(A_2) \longrightarrow A_3$

设指令字长为 32 位

操作码固定为 8 位

4 次访存

寻址范围 $2^6 = 64$

若 PC 代替 A_4

2. 地址码（了解）

（2） 三地址

8	8	8	8
OP	A_1	A_2	A_3

(A_1) OP $(A_2) \longrightarrow A_3$

4 次访存

寻址范围 $2^8 = 256$

若 A_3 用 A_1 或 A_2 代替

2. 地址码（了解）

（3） 二地址

8	12	12
OP	A_1	A_2

(A_1) OP $(A_2) \longrightarrow A_1$

或

(A_1) OP $(A_2) \longrightarrow A_2$

4 次访存

寻址范围 $2^{12} = 4$ K

若结果存于 ACC　　3次访存　　若ACC 代替 A_1（或A_2）

2. 地址码（了解）

（4） 一地址

8	24
OP	A_1

2 次访存

(ACC) OP (A_1) ⟶ACC

寻址范围 $2^{24} = 16$ M

（5） 零地址　　无地址码

7.1.1指令的一般格式小结

- 当用一些硬件资源代替指令字中的地址码字段后
  - 可扩大指令操作数的寻址范围
  - 可缩短指令字长
  - 可减少访存次数
- 当指令的地址字段为寄存器时

三地址　OP　R_1，R_2，R_3

二地址　OP　R_1，R_2

一地址　OP　R_1

  - 可缩短指令字长
  - 指令执行阶段不访存

7.1.2 指令字长

指令字长决定于
- 操作码的长度
- 操作数地址的长度
- 操作数地址的个数

1. 指令字长 固定

指令字长 = 机器字长=存储字长

2. 指令字长 可变

半字长、单字长、双字长、多字长

7.2 操作数类型和操作类型

- 7.2.1 操作数类型
- 7.2.2 数据在存储器中的存放方式
- 7.2.3 操作类型

7.2.1 操作数类型

- 地址　　无符号整数
- 数字　　定点数、浮点数、十进制数
- 字符　　ASCII
- 逻辑数　　逻辑运算

7.2.2 数据在存储器中的存放方式

设存储字长32位，存储器按字节编址。12345678H 这个数据如何在主存储器中进行存储？

字地址 31 … 0

0	12_0	34_1	56_2	78_3
4	4	5	6	7

大端次序（big endian）

符合人类书写习惯。

如：网络传输和文件存储。

大端存储：字地址 为 数据高位字节 地址。如IBM 370

特点：数据的最高字节存储在最小字节地址位置。

读数据时，高位字节在前，低位字节在后。

7.2.2 数据在存储器中的存放方式

设存储字长32位，存储器按字节编址。12345678H 这个数据如何在主存储器中进行存储？

字地址

	31			0
0	12_3	34_2	56_1	78_0
4	7	6	5	4

小端次序（little endian）符合计算从低位起的规律。

如：计算机内部数据处理

小端存储：字地址 为 数据低位字节 地址。如如RISC-V处理器、Intel x86。

特点：数据的最低字节存储在最小字节地址位置。

读数据时，低位字节在前，高位字节在后。

单选题 1分

设置

Henan University

6. 若数据在存储器中采用以低字节地址为字地址的存放方式（小端存储），则十六进制数12345678H的存储字节顺序按地址由小到大依次为(　　)。 B

A 12345678

B 78563412

C 87654321

D 34127856

提交

设机器字长 = 存储字长为 32 位

7.2.2 数据在存储器中的存放方式

1. 不对准边界（从任意位置开始）

字地址为低字节地址　　低字节

0	c（地址3）	b（地址1）		a（地址0）
4	7	c（地址4）		
8	11	10	9	8

32位机器字长，VC程序:
char a; //字符型，1B
short b; //短整型，2B
int c; //整型，4B

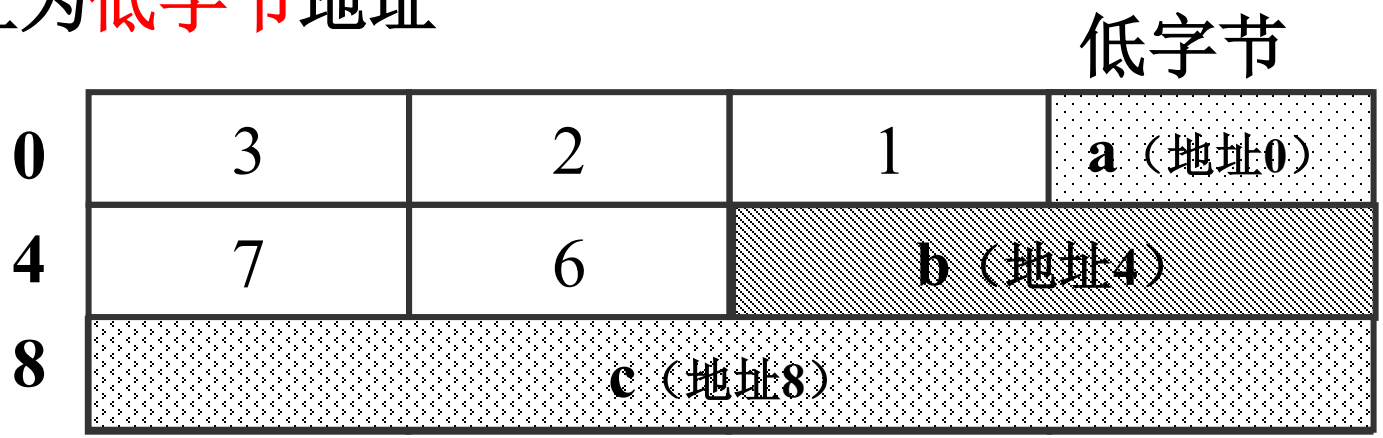
- 优点：无空间浪费
- 缺点：大部分数据可能需访存2次，花2个存取周期，读写控制复杂。

设机器字长 = 存储字长为 32 位

7.2.2 数据在存储器中的存放方式

2. 从存储字的起始位置开始

字地址为低字节地址



32位机器字长，VC程序:

```
char a;  //字符型，1B
short b;  //短整型，2B
int c;   //整型，4B
```

- 优点：大部分数据只需访存1次，读写控制简单。
- 缺点：浪费资源

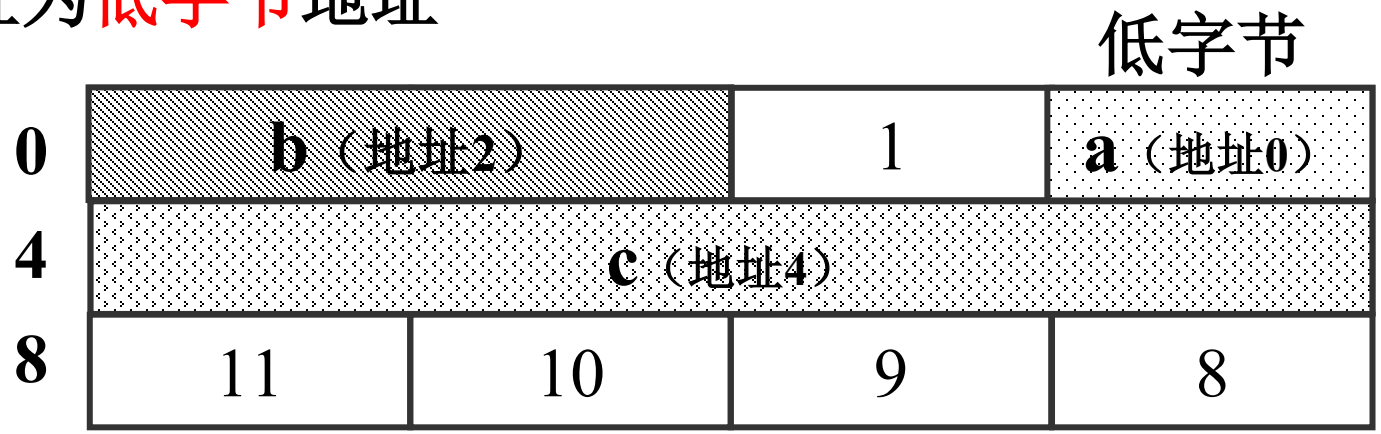
设机器字长 = 存储字长为 32 位

7.2.2 数据在存储器中的存放方式

3. 对准边界： 数据存放的起始地址是**数据长度（长度根据编址单位计算）**的整数倍。

若数据为16位，可按字节编址，则数据长度为16/8=2

字地址为**低字节**地址



32位机器字长，VC程序:

```
char a;  //字符型，1B
short b;  //短整型，2B
int c;   //整型，4B
```

- 优点：有利于提升数据访问速度

char型的数据长度是1；
short型的数据长度是2；
int型的数据长度是4。

设机器字长 = 存储字长为 32 位

7.2.2 数据在存储器中的存放方式

■ 对准边界（课本上的图）

				地址（十进制）
字（地址 0）				0
字（地址 4）				4
字节（地址11）	字节（地址10）	字节（地址 9）	字节（地址 8）	8
15	字节（地址14）	字节（地址13）	字节（地址12）	12
半字（地址18）		半字（地址16） ✓		16
半字（地址22）		半字（地址20）		20
双字（地址24） ▲				24
双字				28
双字（地址32） ▲				32
双字				36

7.2.3 操作类型（自学）

1. 数据传送

源	寄存器	寄存器	存储器	存储器
目的	寄存器	存储器	寄存器	存储器
例如	MOVE	STORE MOVE PUSH	LOAD MOVE POP	MOVE

置“1”，清“0”

7.2.3 操作类型（自学）

2. 算术逻辑操作

加、减、乘、除、增 1、减 1、求补、浮点运算、十进制运算

与、或、非、异或、位操作、位测试、位清除、位求反

如 8086 ADD SUB MUL DIV INC DEC CMP NEG
AAA AAS AAM AAD
AND OR NOT XOR TEST

3. 移位操作

算术移位 逻辑移位

循环移位（带进位和不带进位）

7.2.3 操作类型（自学）

4. 转移

(1) 无条件转移　JMP

(2) 条件转移

结果为零转　（Z = 1）JZ

结果溢出转　（O = 1）JO

结果有进位转（C = 1）JC

跳过一条指令 SKP

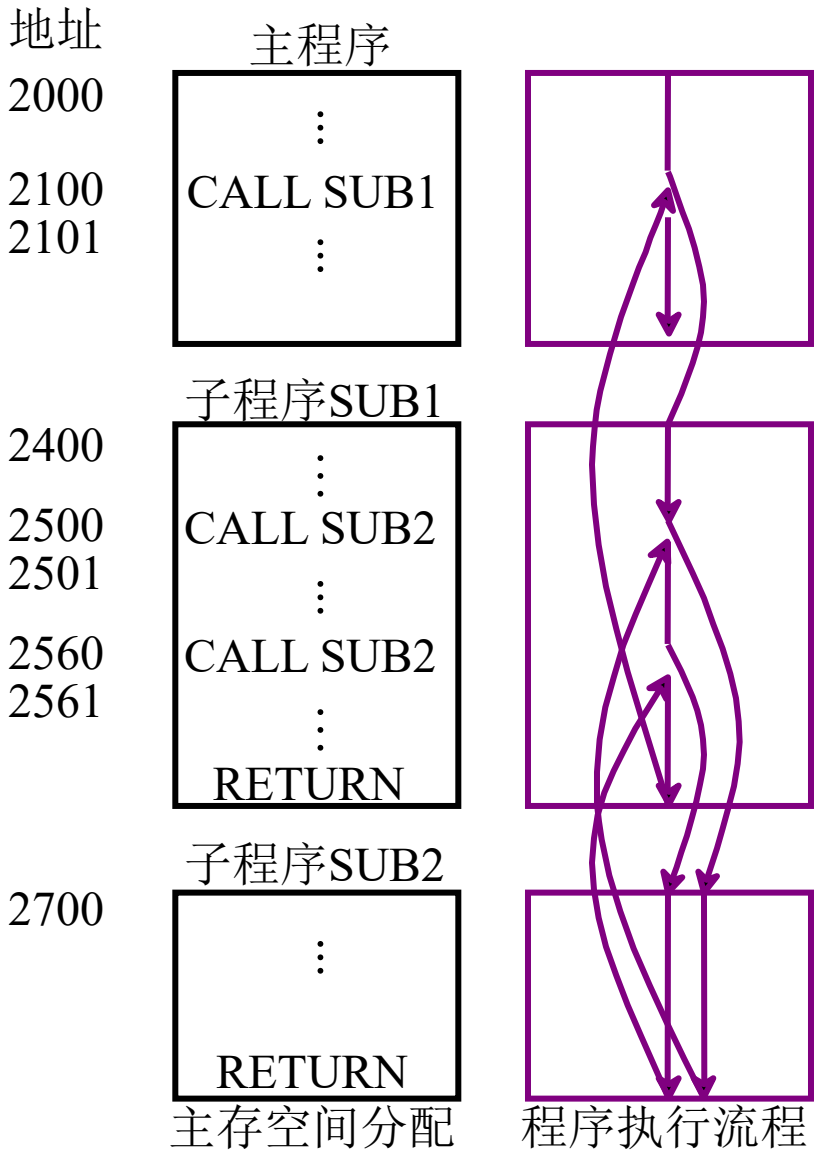
如

300
⋮
305　SKP　DZ　D = 0 则跳
306
307

完成触发器

7.2.3 操作类型（自学）

(3) 调用和返回



7.2.3 操作类型（自学）

(4) 陷阱（Trap）与陷阱指令

意外事故的中断

- 一般不提供给用户直接使用

在出现事故时，由 CPU 自动产生并执行（隐指令）

- 设置供用户使用的陷阱指令

如 8086　　INT　TYPE　　软中断

提供给用户使用的陷阱指令，完成系统调用

7.2.3 操作类型（自学）

5. 输入输出

入　　端口地址 ⟶ CPU 的寄存器

如　IN AX, *n*　　IN AL, DX

出　CPU 的寄存器 ⟶ 端口地址

如　OUT *n*, AX　　OUT DX, AL

7.3 寻址方式

- 7.3.1 指令寻址
- 7.3.2 数据寻址

7.3 寻址方式

寻址方式
- 指令寻址：下一条欲执行指令的地址
- 数据寻址：本条指令的 操作数地址

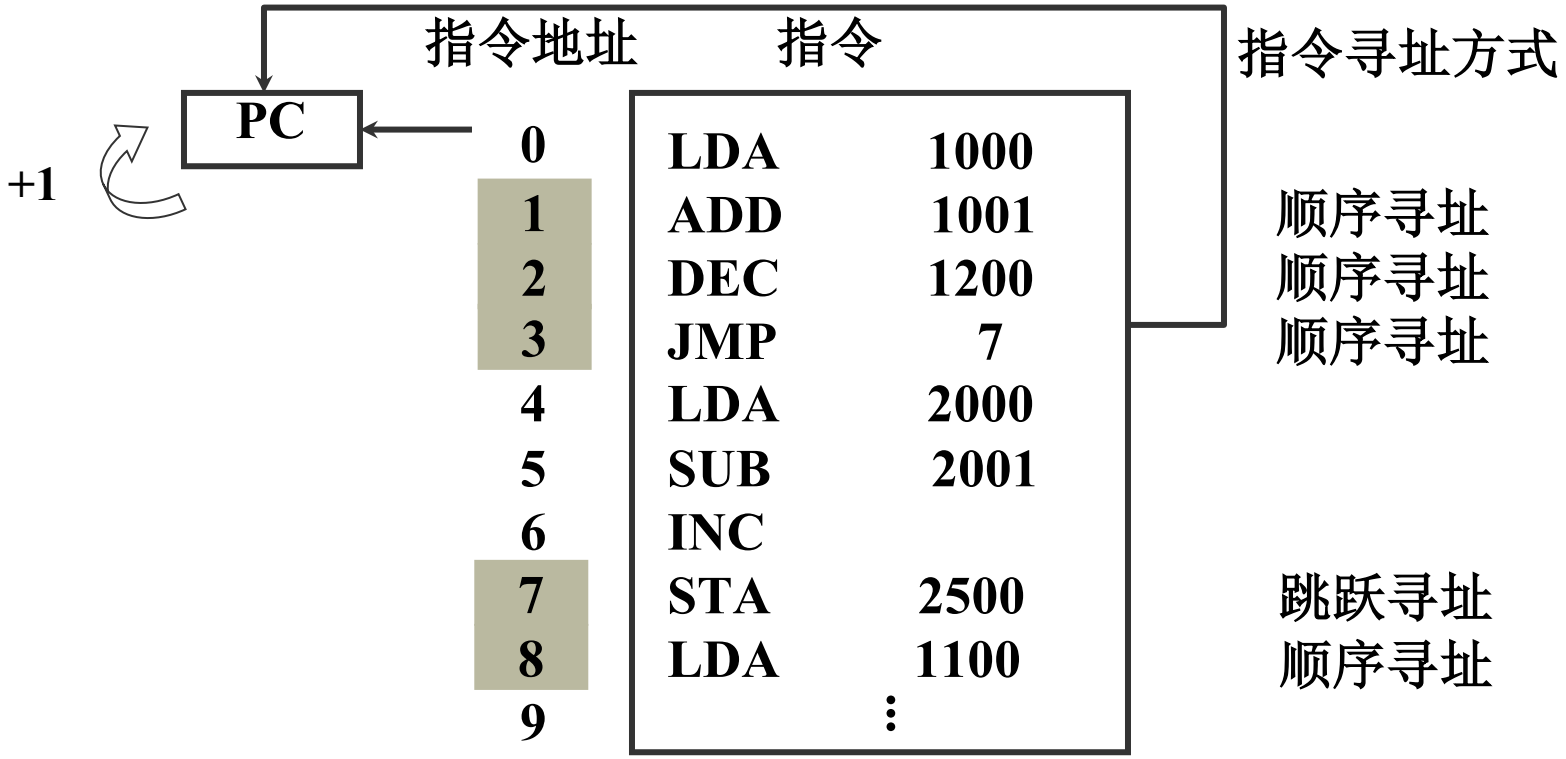
设机器字长 = 存储字长 = 指令字长

7.3.1 指令寻址

根据编址方式和指令长度不同，1是虚指（1个指令字长度）。

顺序寻址　(PC) + 1 → PC

跳跃寻址　PC的内容由转移指令指出



单选题 1分

设置

7. 设机器字长为16位，存储器按字编址，对于单字长指令而言，读取该指令后，PC自动加( )。

C

A 2

B 3

C 1

D 4

提交

7.3.2 数据寻址

第3阶段：执行指令阶段

操作码OP	寻址特征M	形式地址 A

形式地址A：指令字中的地址

有效地址EA: 操作数的真实地址

约定　指令字长 = 存储字长 = 机器字长

7.3.2 数据寻址

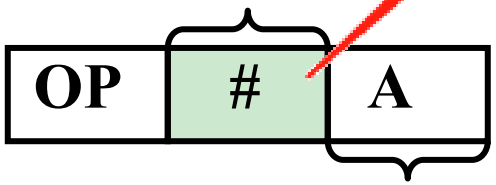
1. 立即寻址

形式地址 A 就是操作数

MCS-51单片机

立即寻址特征

MOV R0 , #1000B



立即数　　可正可负 补码

- 指令执行阶段不访存，执行速度最快
- A 的位数限制了立即数（用补码）的范围

单选题 1分

设置

9. 在指令寻址的各种方式中，获取操作数最快的方式是（　）。

B

A 直接寻址

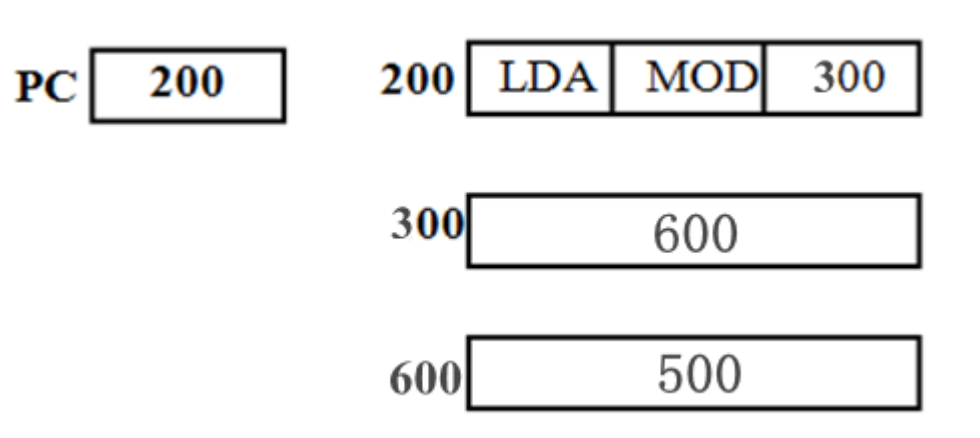
B 立即寻址

C 寄存器寻址

D 间接寻址

提交

- 假设存储器按字编址，现有单字长指令存放在内存200号单元，如图所示。该指令可将操作数调入累加器。若操作数的寻址方式为立即寻址，指令执行完后累加器中的值为__300__，PC中的值为__201__。

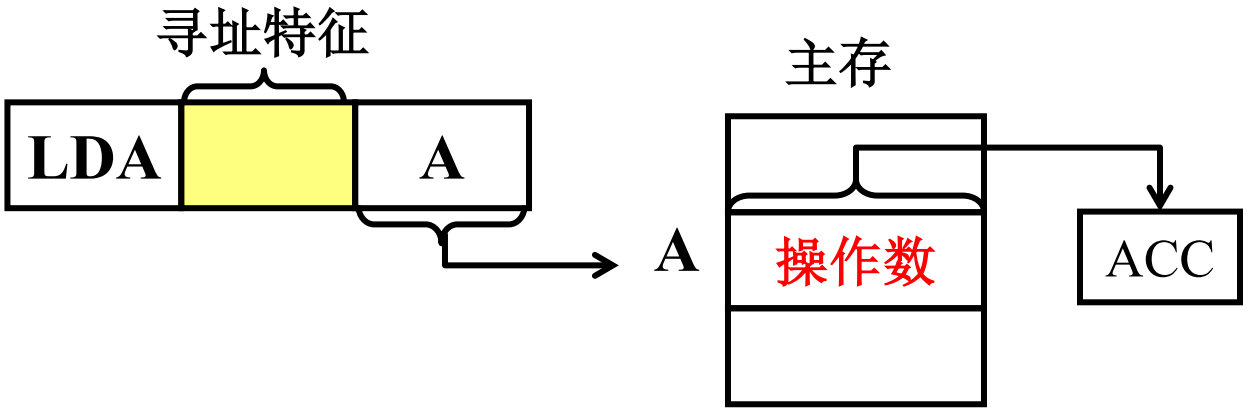


2. 直接寻址

EA = A

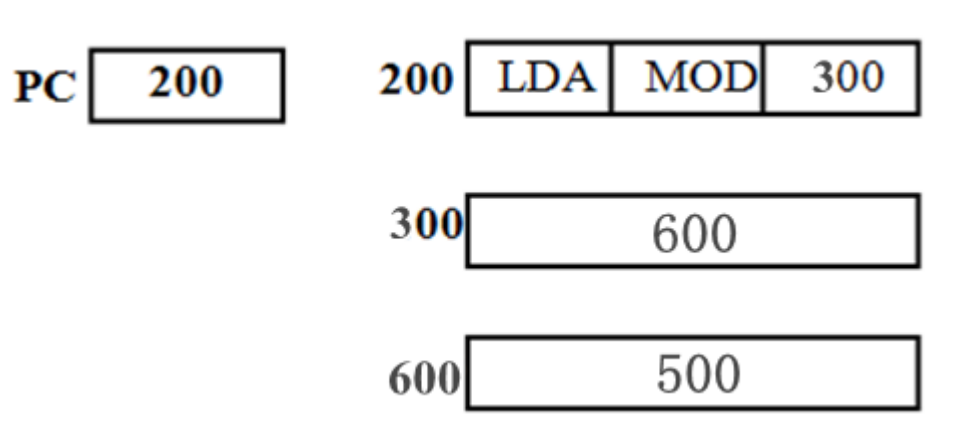
有效地址由形式地址直接给出

如：LDA 20H



- 执行阶段访问一次存储器
- A 的位数决定了该指令操作数的寻址范围
- 操作数的地址不易修改（必须修改A）

- 假设存储器按字节编址，存储字长为32位。现有单字长指令存放在内存200号单元，如图所示。该指令可将操作数调入累加器，若操作数的寻址方式为直接寻址，指令执行完成后累加器中的值为 600 ，PC中的值为 204 。

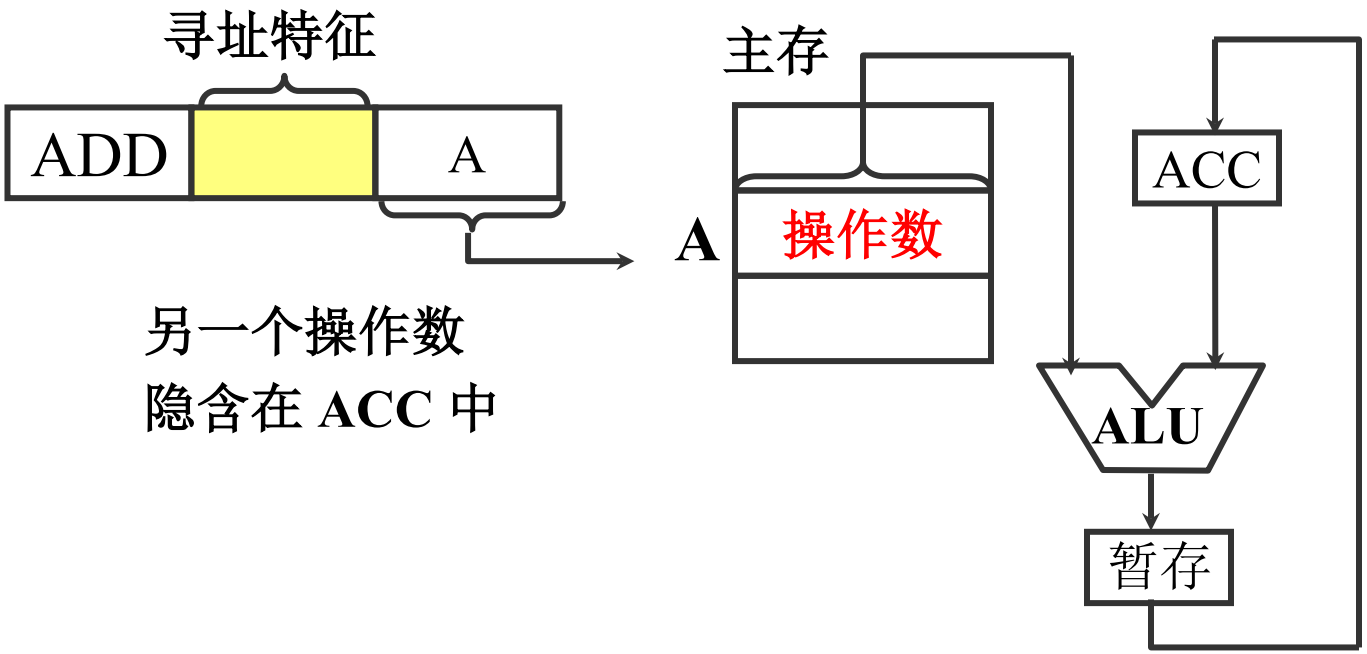


- 假设存储器按字编址，字长为16位。指令操作码占7位，寻址特征占1位（0为立即寻址；1为直接寻址）。现有单字长指令1324H存放在内存200号单元，则该指令的寻址方式为 直接寻址 。有效地址EA= 24H 。该寻址方式的寻址范围是 256 （十进制）。指令执行完成后PC的值为 201 。

PC	200	200	1324H

3. 隐含寻址

操作数地址隐含在操作码中



- 指令字中少了一个地址字段，可缩短指令字长

单选题 1分

设置

12. 一地址加法指令ADD ,一个操作数在内存地址N中,另一个操作数隐含在累加器ACC中。这种寻址方式称为(　　)。

A

A 隐含寻址

B 立即寻址

C 寄存器寻址

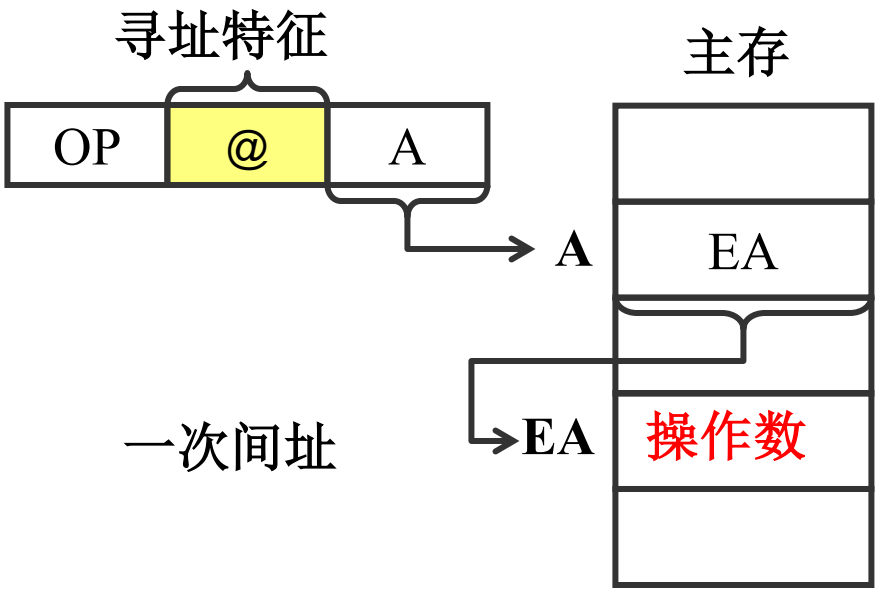
D 间接寻址

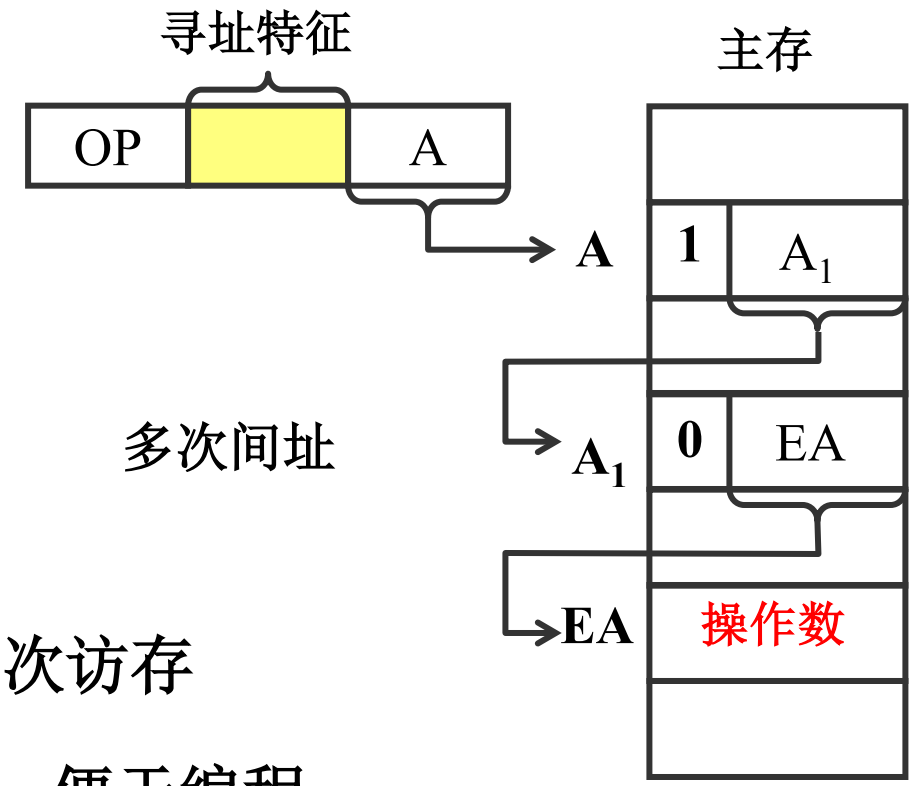
提交

4. 间接寻址

EA =（A）

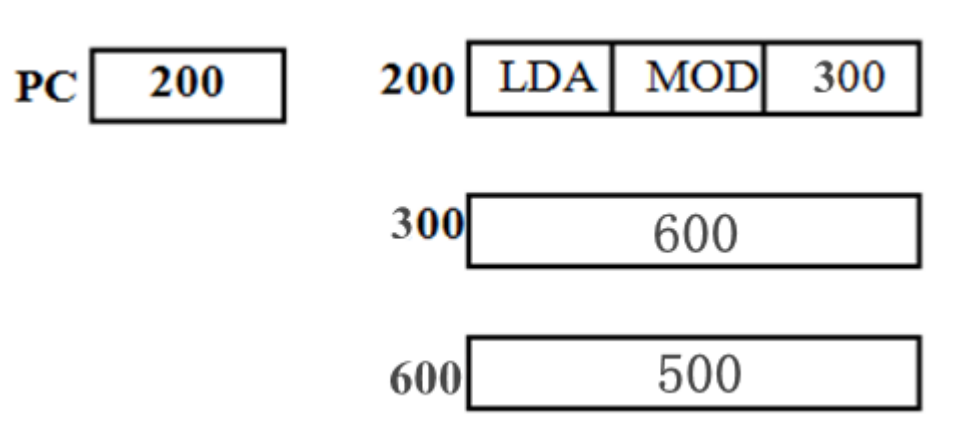
有效地址由形式地址间接提供





- 执行指令阶段 2 次访存
- 可扩大寻址范围，便于编程

- 假设单字长指令可将操作数调入累加器，若操作数的寻址方式为间接寻址，指令执行完成后累加器中的值为__500__。PC中的值为__201__。



- 假设存储器按字编址，字长为16位。指令操作码占7位，寻址特征占1位（0为直接寻址；1为间接寻址）。现有单字长指令1324H存放在内存200号单元，则该指令的寻址方式为 间接寻址 。有效地址EA= 703 。该寻址方式的寻址范围是 65536 （十进制）。指令执行完成后PC的值为 201 。

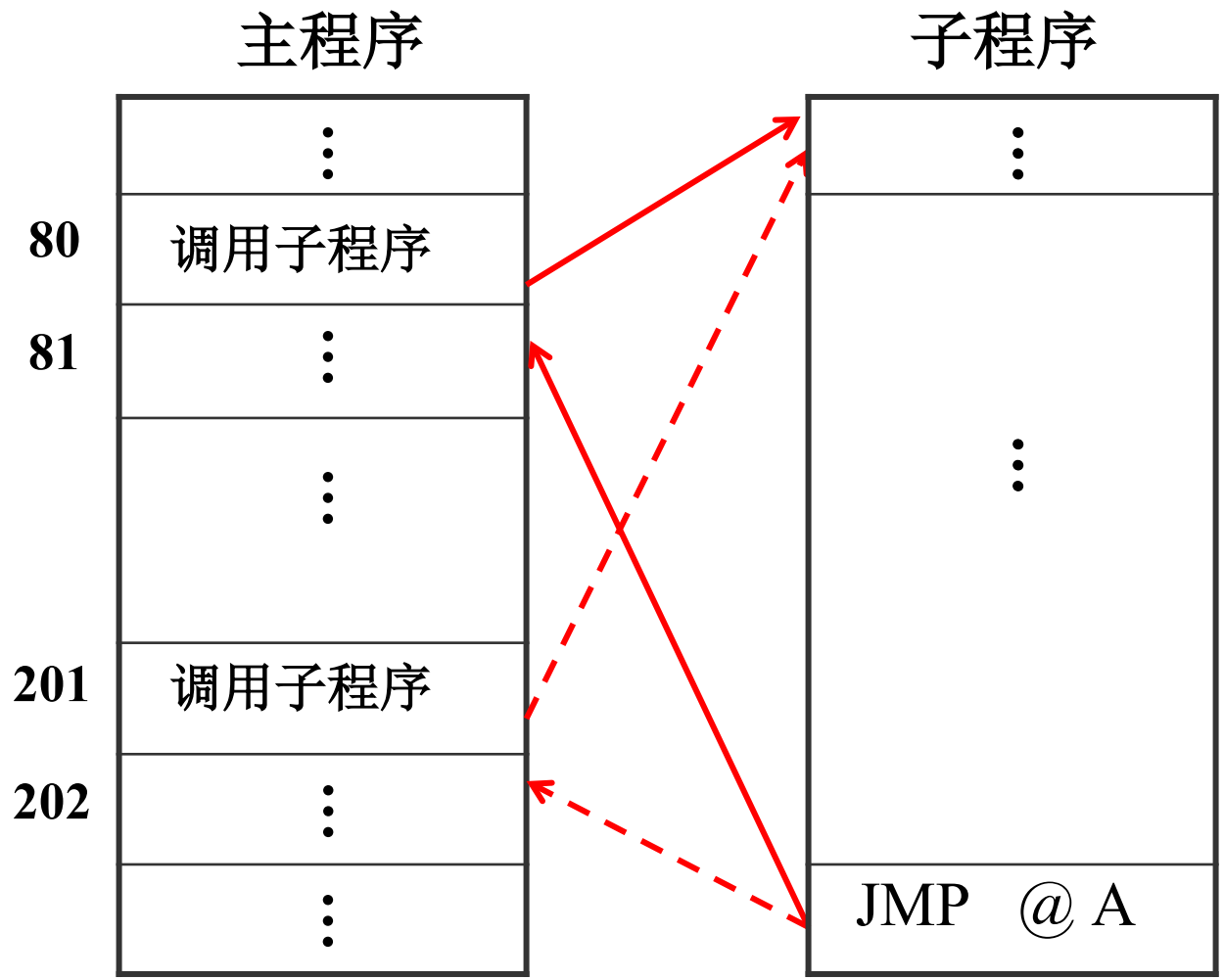
PC	200	200	1324H
		24H	703

4. 间接寻址

间接寻址编程举例

@ 间址特征

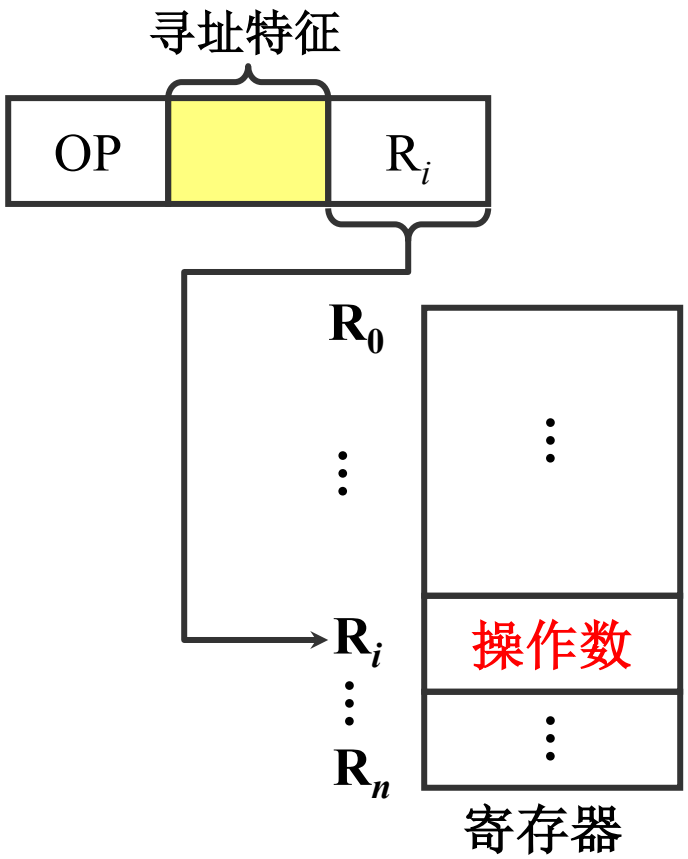
(A) = 202



5. 寄存器寻址

$EA = R_i$

有效地址即为寄存器编号



- 执行阶段不访存，只访问寄存器，执行速度快
- 寄存器个数有限，可有效缩短指令字长

单选题 1分

设置

Henan University

13. 操作数在寄存器中的寻址方式称为(　　)寻址。

B

A 直接

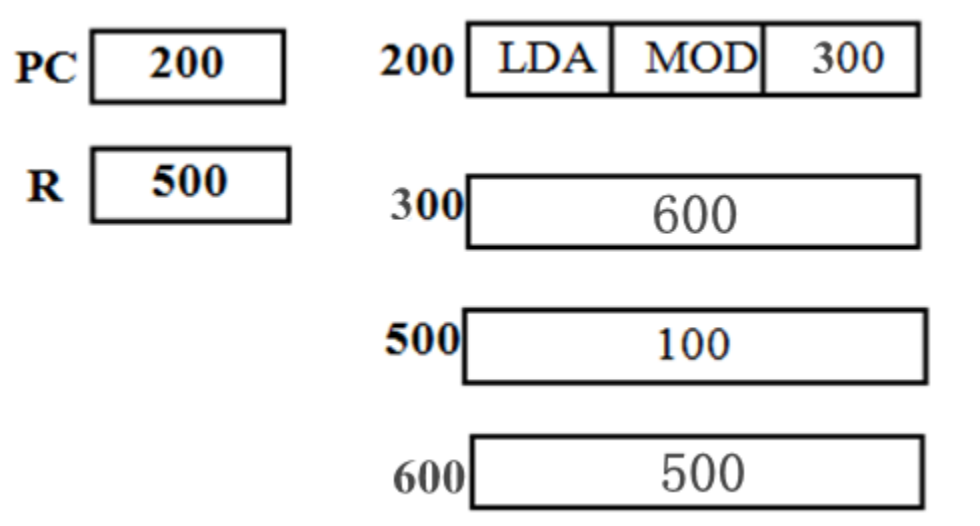
B 寄存器直接

C 寄存器间接

D 立即

提交

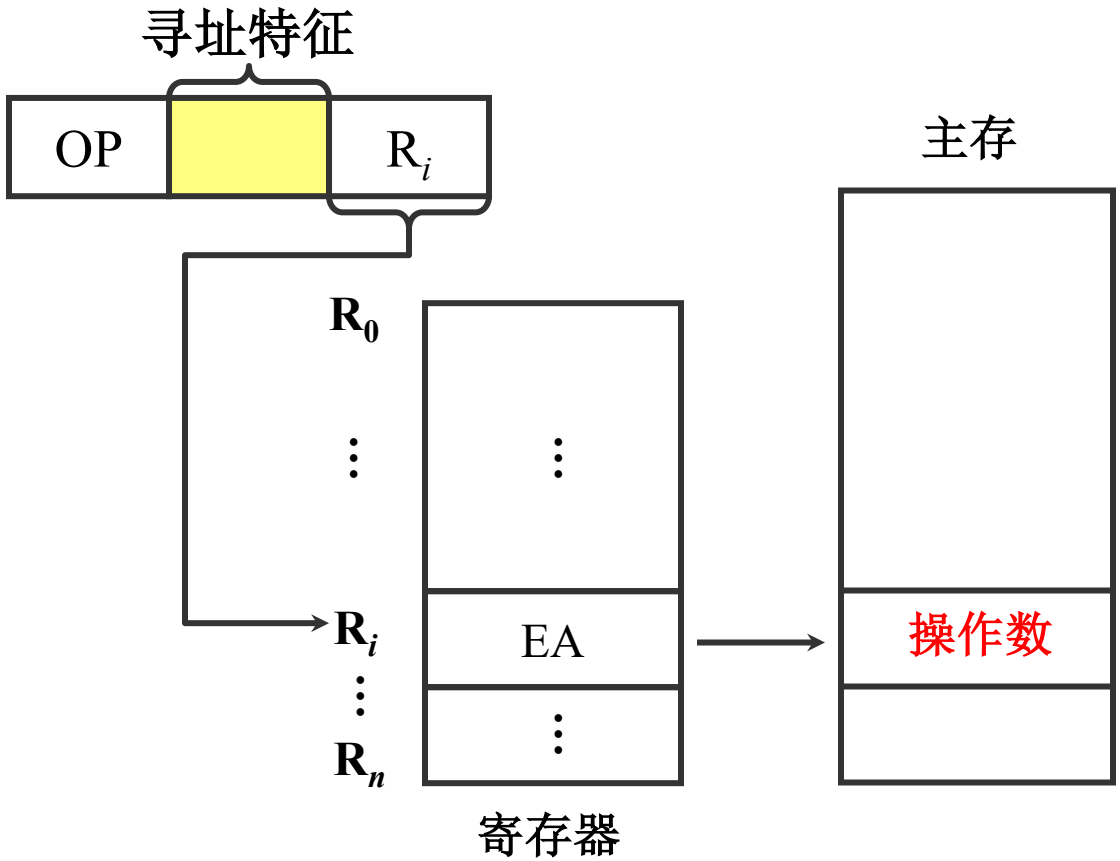
- 假设单字长指令可将操作数调入累加器，若操作数的寻址方式为寄存器寻址，指令执行完成后累加器中的值为500。PC的值为201。



6. 寄存器间接寻址

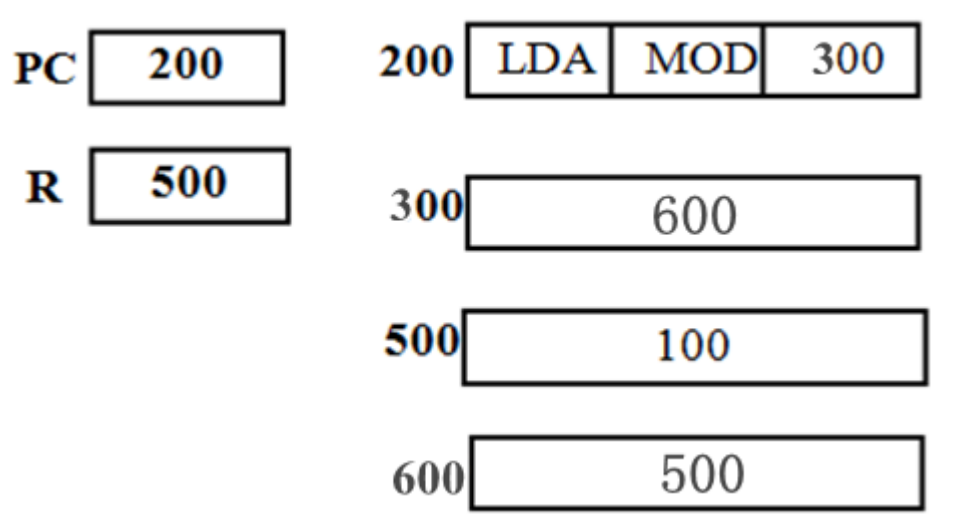
$EA = (R_i)$

有效地址在寄存器中



- 有效地址在寄存器中， 操作数在存储器中，执行阶段访存

- 假设单字长指令可将操作数调入累加器，若操作数的寻址方式为寄存器间接寻址，指令执行完成后累加器中的值为__100__。PC中的值为_____201_____。



补充：偏移寻址

- 直接寻址和寄存器间接寻址方式的结合。
- 有效地址EA=A + (R)
  - A是显式的形式地址字段，又称位移量；
  - R可以是显式的，也可以是隐含的某个专用的寄存器；
- 常用的偏移寻址：
  - 基址寻址：EA=(基址R)+A ，基址R不变，A可变
  - 变址寻址：EA=(变址R)+A ，A不变，变址R可变
  - 相对寻址：EA=(PC)+A

单选题 1分

设置

15. 【2011 偏移寻址】偏移寻址通过某个寄存器内容与一个形式地址相加而生成有效地址，下列寻址方式中，不属于偏移寻址方式的是(　　)。

A

A 间接寻址

B 基址寻址

C 相对寻址

D 变址寻址

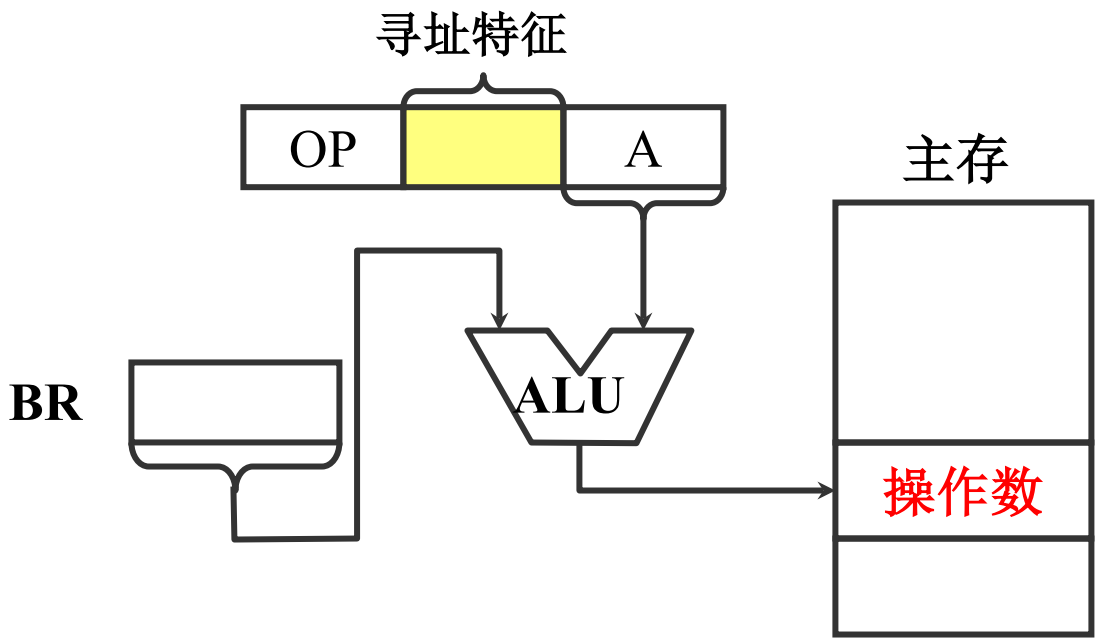
提交

7. 基址寻址

（1）采用专用寄存器作基址寄存器BR—隐式基址寻址

EA = (BR) + A

BR 为基址寄存器

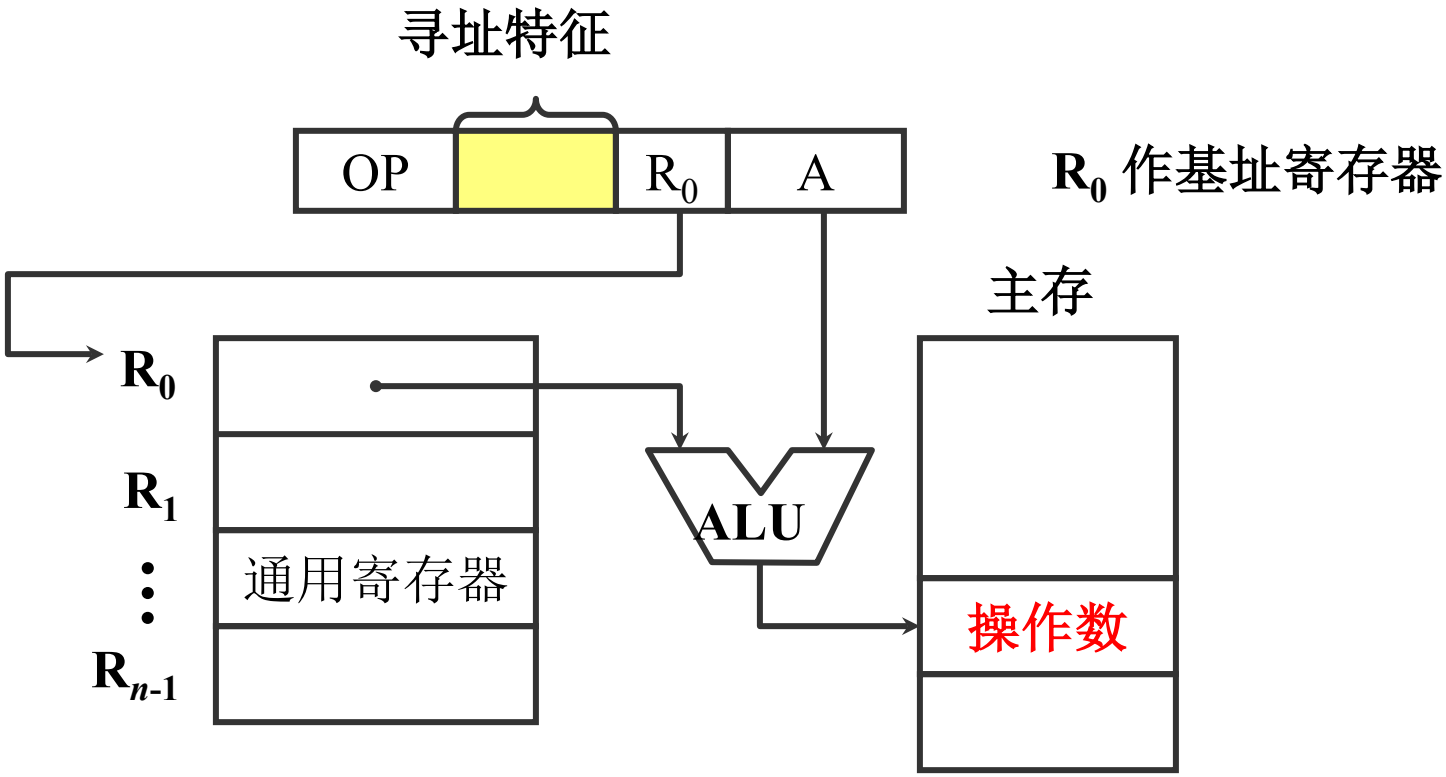


7. 基址寻址

（2）采用通用寄存器作基址寄存器—显式基址寻址

$EA = (R_0) + A$

R_0 为通用寄存器



7. 基址寻址

- 可扩大寻址范围

- 主要用于为程序或数据分配存储空间，基址寄存器的内容通常由操作系统或管理程序确定，在程序的执行过程中基址寄存器内容不变，指令中的形式地址 A 可变。

- 有利于多道程序：操作系统或管理程序根据内存使用情况，赋予基址寄存器一个初始值（即基地址），在程序执行过程中，改变形式地址A，完成程序执行。

单选题 1分

设置

16. 设指令中的地址码为A，基址寄存器为BR，则基址寻址方式的操作数有效地址EA是(　　)。

D

A ((PC)+A)

B ((BR)+A)

C (BR)+(A)

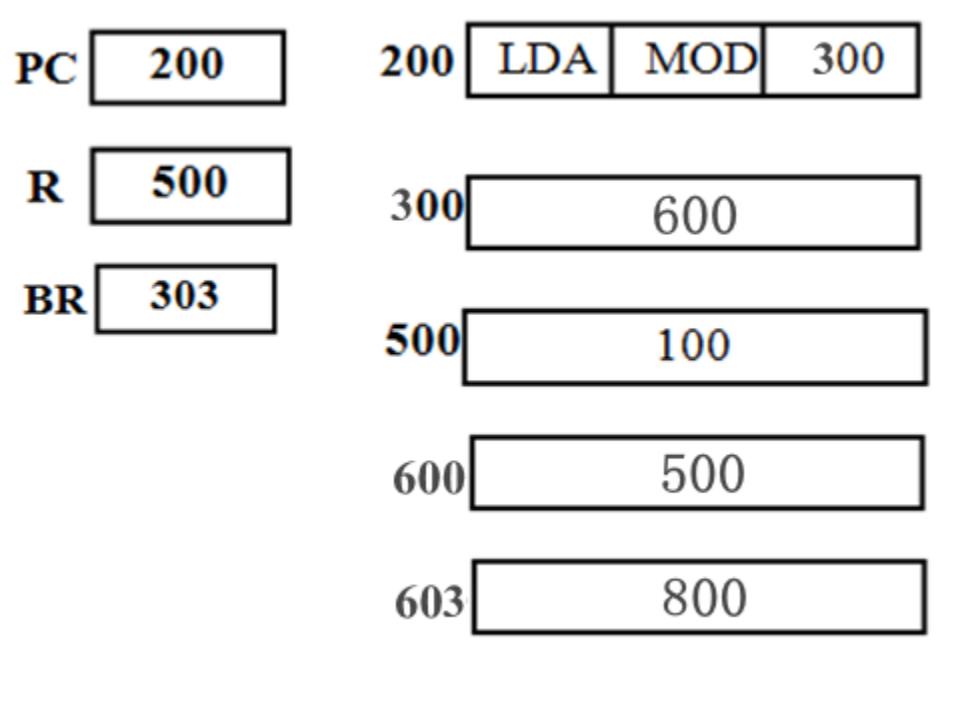
D (BR)+A

提交

- 假设存储器按字编址，字长为16位。指令操作码占7位，寻址特征占1位（0为立即寻址；1为基址寻址）。基址寄存器BR为16位。现有单字长指令1324H存放在内存200号单元，则该指令的寻址方式为 基址寻址 。有效地址EA= 35H 。

PC	200	200	1324H
BR	11H	24H	703
		35 H	100

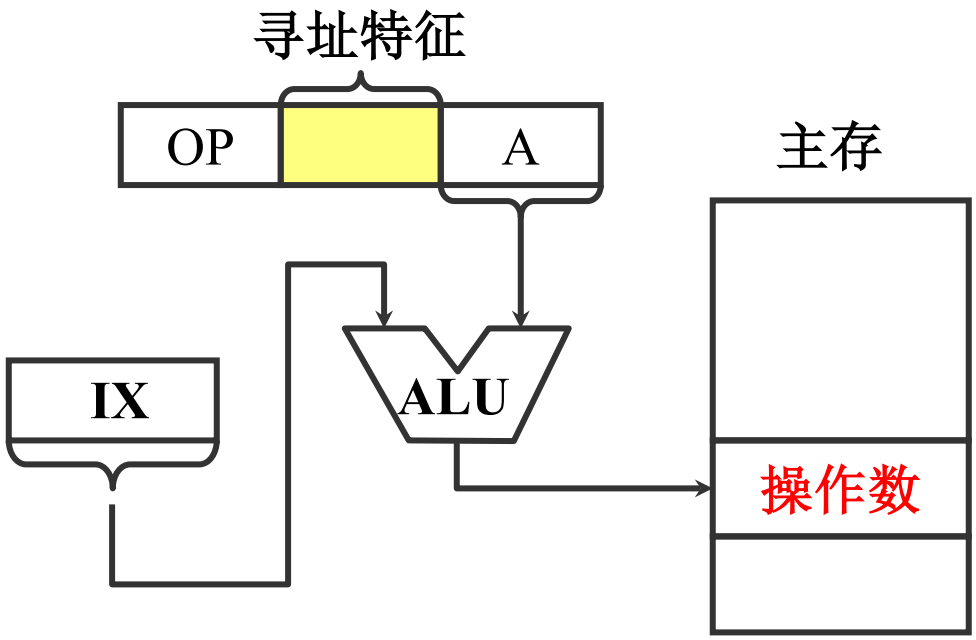
- 假设单字长指令可将操作数调入累加器，若操作数的寻址方式为基址寻址，则执行完成后累加器中的值为__800__。PC的值为__201__。



8. 变址寻址

（1）采用专用寄存器作变址寄存器IX—隐式变址寻址

$\boxed{EA = (IX) + A}$　　IX 为变址寄存器（专用）

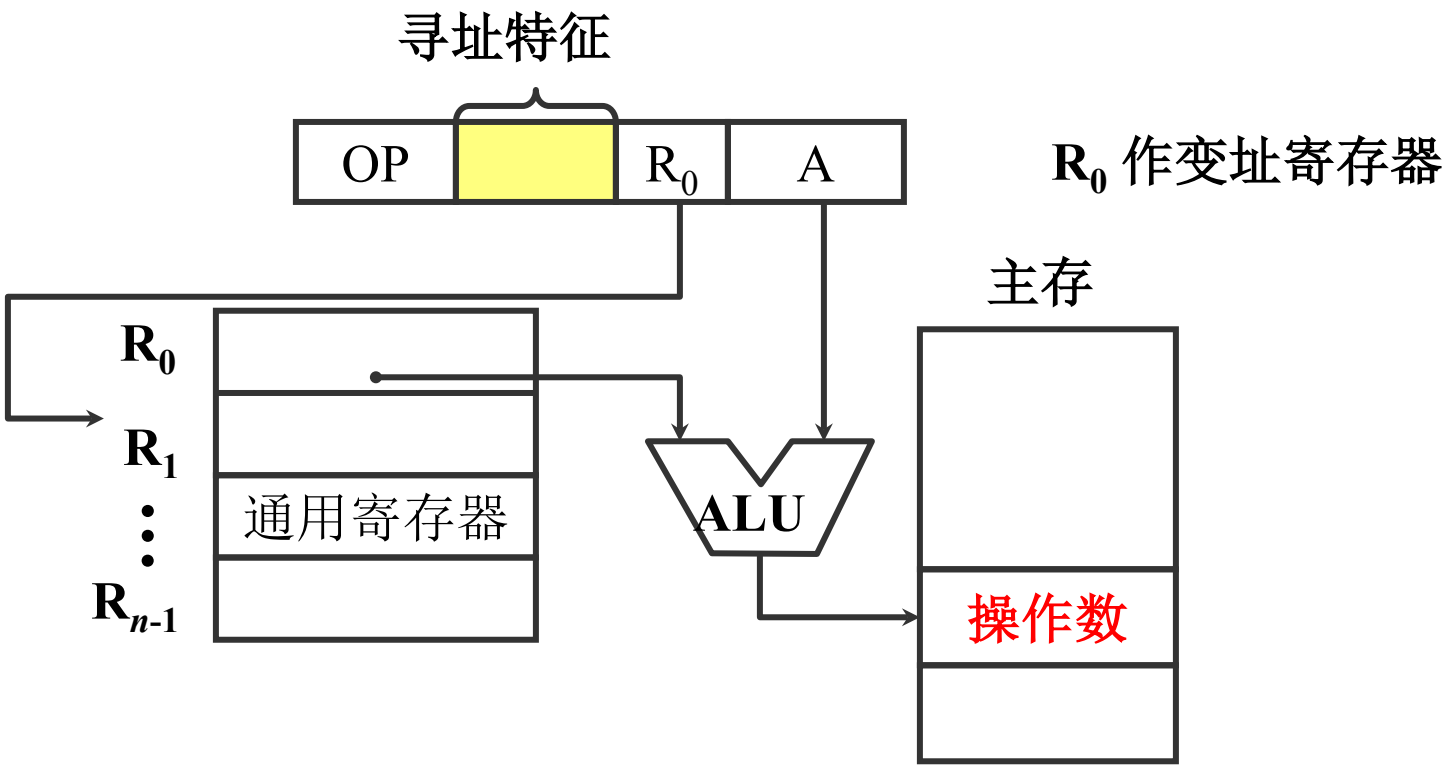


8. 变址寻址

（2）采用通用寄存器作变址寄存器IX—显式变址寻址

$EA = (R_0) + A$

R_0 为通用寄存器



8. 变址寻址

- 可扩大寻址范围
- 变址寻址中，变址寄存器的内容由用户设定，在程序的执行过程中其内容可变，而指令中的形式地址 A 不变。
- 变址寻址特别适合编写循环程序，主要用于处理数组问题：设定形式地址A为数组首地址，不断改变变址寄存器内容，即可形成数组中任意数据的地址。

例 设数据块首地址为 *D*，求 *N* 个数的平均值

直接寻址

```
LDA   D
ADD   D + 1
ADD   D + 2
      ⋮
ADD   D + ( N -1 )
DIV   # N
STA   ANS
```

共 *N* + 2 条指令

变址寻址

```
      LDA   # 0
      LDX   # 0   X 为变址寄存器
M     ADD   X, D        D 为形式地址
      INX               (X) +1 ⟶X
      CPX   # N        (X) 和 #N 比较
      BNE   M           结果不为零则跳转
      DIV   # N
      STA   ANS
```

共 8 条指令

单选题 1分

设置

17. 设指令中的地址码为A，变址寄存器为X，程序计数器为PC，则变址寻址方式的操作数有效地址EA是(　　)。

D

A ((PC)+A)

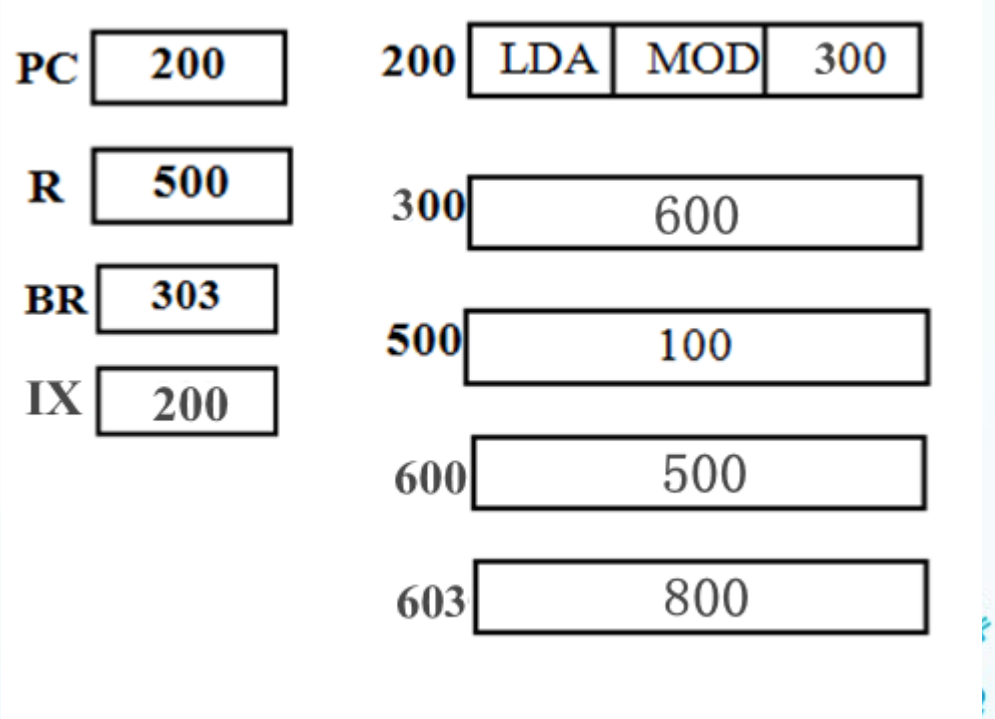
B ((X)+A)

C (X)+(A)

D (X)+A

提交

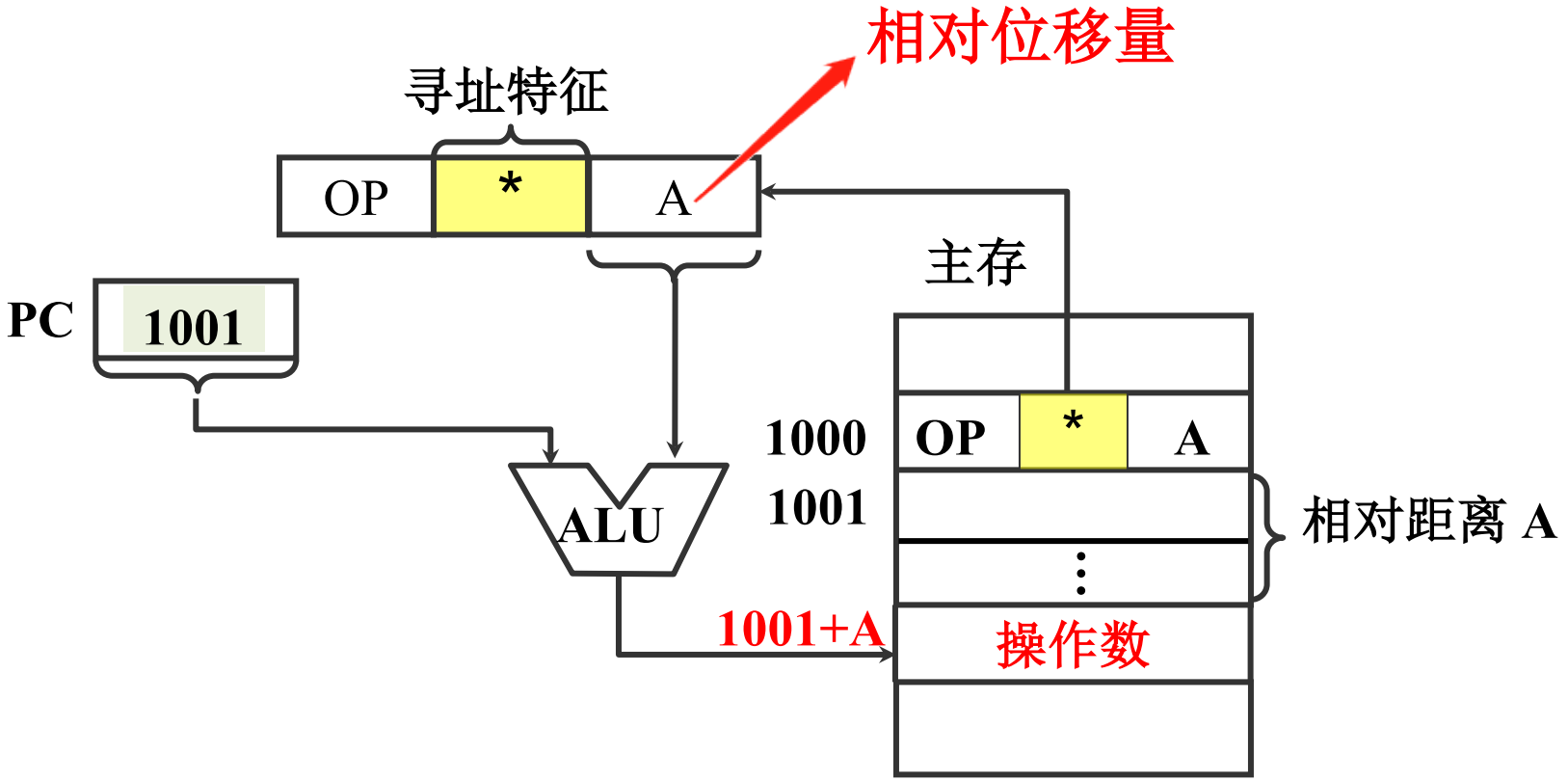
- 假设单字长指令可将操作数调入累加器，如图所示。若操作数的寻址方式为变址寻址，指令执行完成后执行完指令后则累加器中的值为__100__，PC的值为__201__。



9. 相对寻址

$$\boxed{EA = (PC) + A}$$

A 是相对于当前指令的位移量（可正可负，补码）



单选题 1分

设置

19. 设指令中的地址码为A，变址寄存器为X，程序计数器为PC，则相对寻址方式的操作数有效地址EA是(　　)。

A (PC)+A

B ((X)+A)

C (X)+(A)

D (X)+A

A

提交

9. 相对寻址

- 位移量A 的位数决定操作数的寻址范围，用补码表示
- 有利于程序浮动
- 广泛用于转移指令

（1） 相对寻址举例

	LDA	# 0
	LDX	# 0
M	ADD	X, D
M+1	INX	
M+2	CPX	# N
M+3	BNE	M
	DIV	# N
	STA	ANS

→ ✻ – 3

✻ 相对寻址特征

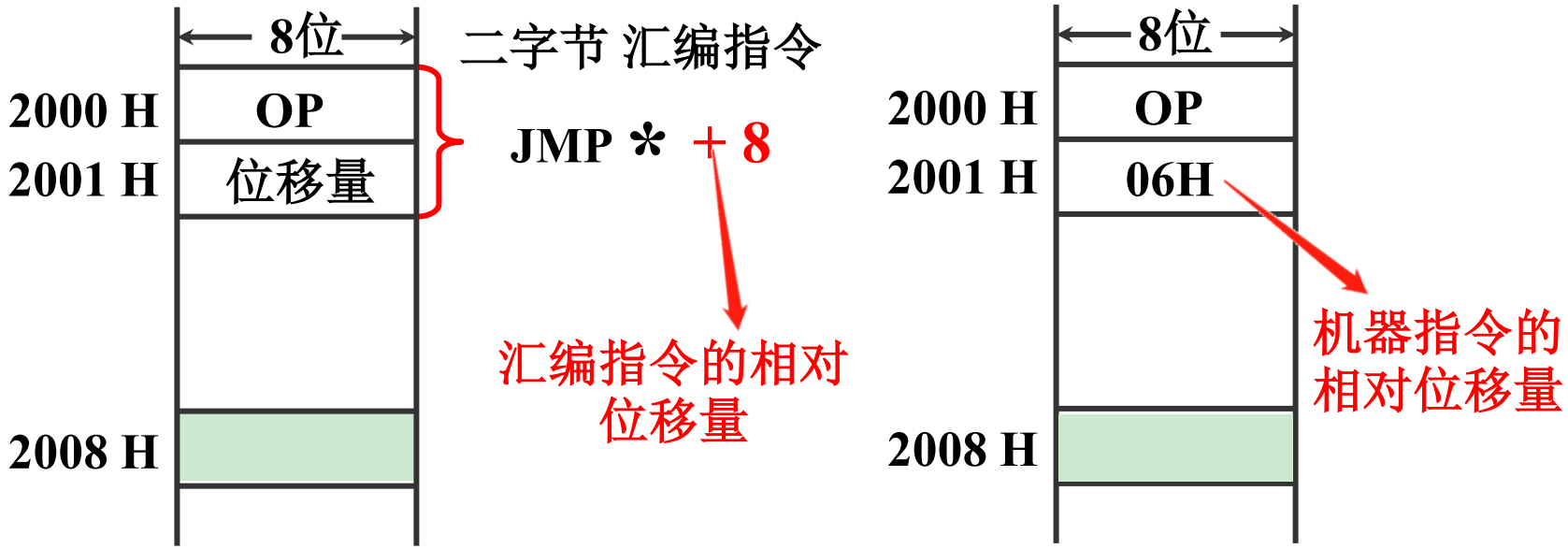
M 随程序所在存储空间的位置不同而不同

而指令 BNE ✻ – 3 与 指令 ADD X, D 相对位移量不变

指令 BNE ✻ – 3 操作数的有效地址为

EA = (M+3) – 3 = M

（2） 按字节寻址的相对寻址举例



指令JMP * + 8 的地址为 2000H

转移后的目的地址为 2008H 求其对应机器指令的位移量 = ?

因为 取出 JMP * + 8 后 PC = 2002H

故 JMP * + 8 指令对应机器指令 的第二字节为 2008H - 2002H = 06H

- 假设存储器按字编址，某单字长指令可将操作数调入累加器，若操作数的寻址方式为相对寻址，取出指令后PC的值为__201__。指令执行完成后累加器中的值为__700__。

PC	200
R	500
BR	303
IX	200

200	LDA	MOD	300
300	600		
500	100		
501	700		
600	500		
603	800		

例7.2 设相对寻址的转移指令占 3 个字节，第一字节为操作码，第二、三字节为相对位移量（补码表示）。数据在存储器中采用以低字节地址为字地址的存放方式。每当CPU 从存储器取出一个字节时，即自动完成(PC)+1→PC 。

（1）若PC 当前值为 $(240)_{10}$，要求转移到 $(290)_{10}$，则转移指令的第二、三字节的机器代码是什么？

第3字节	第2字节	第1字节
00	2F	操作码OP

该指令取出后PC = 243，要求转移到290，即相对位移量为290-243 = 47，转换成补码为 2FH = 00 2FH。

例7.2 设相对寻址的转移指令占 3 个字节，第一字节为操作码，第二、三字节为相对位移量（补码表示）。数据在存储器中采用以低字节地址为字地址的存放方式。每当CPU 从存储器取出一个字节时，即自动完成(PC)+1→PC 。

（2）若PC 当前值为 $(240)_{10}$，要求转移到 $(200)_{10}$，则转移指令的第二、三字节的机器代码是什么？

第3字节	第2字节	第1字节
FF	D5	操作码OP

该指令取出后PC = 243，要求转移到200，即相对位移量为200-243 = -43，转换成补码为 D5H = FF D5H。

设指令字长=存储字长，按字编址

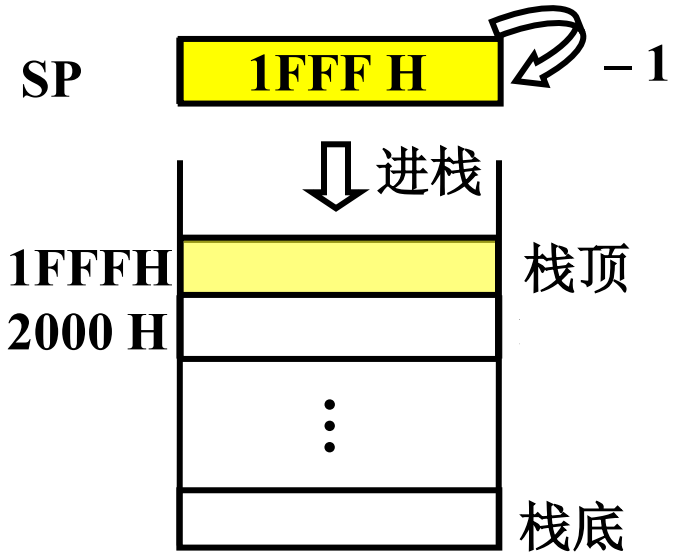
10. 堆栈寻址

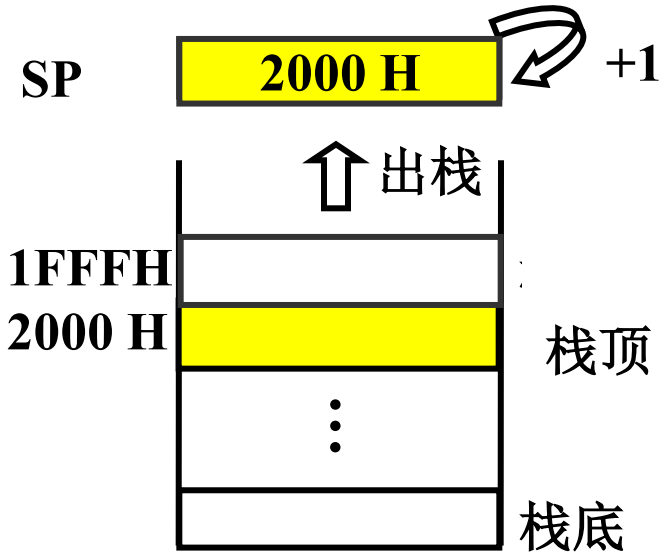
（1） 堆栈的特点

堆栈 { 硬堆栈　　多个寄存器
　　软堆栈　　指定的存储空间

先进后出（一个入出口）　栈顶地址 由 SP 指出

进栈（push）（SP）–1 ⟶SP　　出栈（pop）（SP）+ 1 ⟶ SP





10. 堆栈寻址（自学）

（3） SP 的修改与主存编址方法有关

① 按 字 编址
- 进栈（SP）– 1 ⟶ SP
- 出栈（SP）+ 1 ⟶ SP

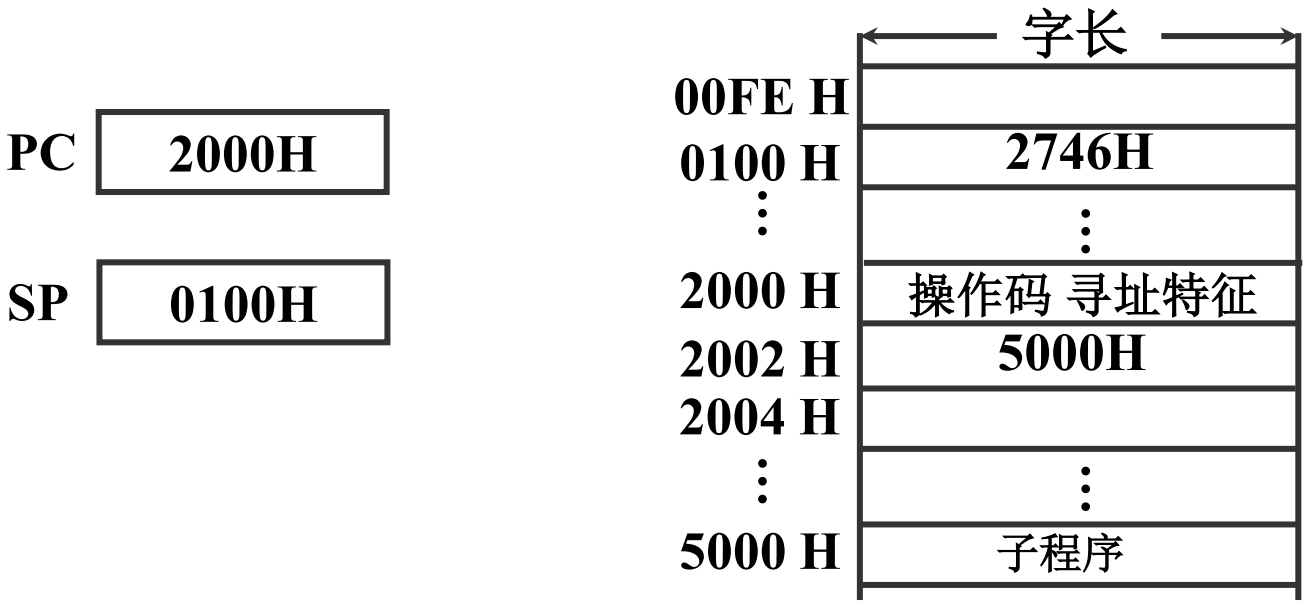
② 按 字节 编址

存储字长 16 位
- 进栈（SP）– 2 ⟶ SP
- 出栈（SP）+ 2 ⟶ SP

存储字长 32 位
- 进栈（SP）– 4 ⟶ SP
- 出栈（SP）+ 4 ⟶ SP

例7.3 一条双字长直接寻址的子程序调用指令，第一个字为操作码和寻址特征，第二个字为地址码5000H。假设PC当前的值为2000H，SP的内容为0100H，栈顶内容为2746H，存储器按字节编址，而且进栈操作是先执行(SP)-Δ→SP，后存入数据。试回答下列几种情况下，PC、SP及栈顶的内容各是多少？

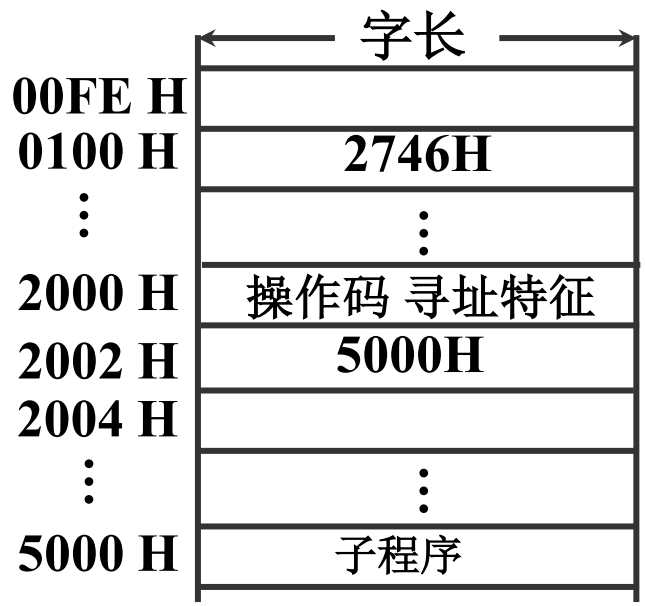
字长?位 =16位



例7.3 一条双字长直接寻址的子程序调用指令，第一个字为操作码和寻址特征，第二个字为地址码5000H。假设PC当前的值为2000H，SP的内容为0100H，栈顶内容为2746H，存储器按字节编址，而且进栈操作是先执行(SP)-Δ→SP，后存入数据。试回答下列几种情况下，PC、SP及栈顶的内容各是多少？

字长?位=16位

PC	2000H
SP	0100H



解：（1）CALL指令被读取前。

PC的值为2000H

SP的内容为0100H

栈顶内容为2746H

（2） CALL指令被执行之后

字长?位 =16位

解： CALL指令被取出之后，PC=2004H。

PC	5000H
SP	00FEH

	字长
00FE H	2004H
0100 H	2746H
⋮	⋮
2000 H	操作码 寻址特征
2002 H	5000H
2004 H	
⋮	⋮
5000 H	子程序

调用子程序之前需要保护现场，即需要将程序断点地址（2004H）进栈保存。

此时SP=SP-2=0100H-2=00FEH，即栈顶内容为2004H。

执行指令后，PC的内容被更新为子程序的入口地址5000H

（3） 子程序返回后

PC	2004H
SP	0100 H

字长?位 =16位

	字长
00FE H	2004H
0100 H	2746H
⋮	⋮
2000 H	操作码 寻址特征
2002 H	5000H
2004 H	
⋮	⋮
5000 H	子程序

解：子程序返回，需恢复现场

此时SP=SP+2=00FEH+2=0100 H

栈顶内容2004H出栈送给PC。

栈顶内容为2746H。

7.4 指令格式举例

- 7.4.1 设计指令格式应考虑的各种因素
- 7.4.2 指令格式举例
- 7.4.3 指令格式设计举例

例7.5 某机配有基址寄存器和变址寄存器，采用一地址格式指令系统，允许直接和间接寻址，且指令字长、机器字长和存储字长为16位。

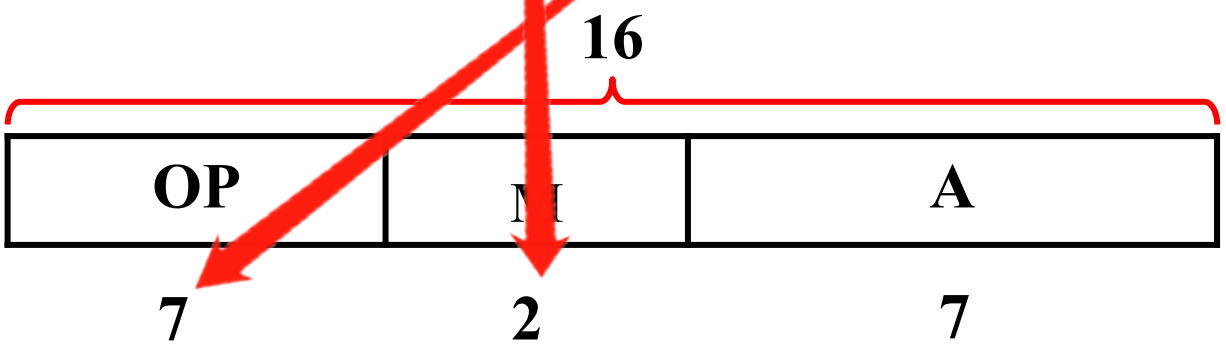
（1）若采用单字长指令，共能完成105种操作，则指令可直接寻址的范围是？一次间接寻址的寻址范围是？画出指令格式。

（2）若存储字长不变，可用什么方法直接访问容量为16MB的主存？

例7.5 某机配有基址寄存器和变址寄存器，采用一地址格式指令系统，允许直接和间接寻址，且指令字长、机器字长和存储字长为16位。

（1）若采用单字长指令，共能完成105种操作，则指令可直接寻址的范围是？一次间接寻址的寻址范围是？画出指令格式。

解：（1）单字长指令共16位，一地址格式如下：



A的位数为7位，直接寻址EA=A，其寻址范围范围为 $2^7=128$

一次间接寻址EA=(A)，其寻址范围为$2^{16}=65536$

例7.5 某机配有基址寄存器和变址寄存器，采用一地址格式指令系统，允许直接和间接寻址，且指令字长、机器字长和存储字长为16位。

（2）若存储字长不变，可用什么方法直接访问容量为16MB的主存？

解：（2）16MB=8M×16位，可得寻址范围为8M=2^{23}

可采用双字长指令，如下：

OP	M	A
A		

相对转移指令：

OP(6)	M(2)	A(8)

例7.6 某机共有64种操作，操作码位数固定，且：

OP=6位

① 采用一地址或二地址；

② 有寄存器寻址、直接寻址和相对寻址（位移量为-128～+127）三种寻址方式；

寻址特征位M=2

③ 16个通用寄存器，算术运算和逻辑运算的操作数均在寄存器中，结果也在寄存器中；

OP(6)	M(2)	R_1(4)	R_2(4)

④ 取数/存数指令在通用寄存器和存储器之间传送数据；

OP(6)	M(2)	R_1(4)	A(20)

⑤ 存储器容量为1MB，按字节编址。

存储器地址20位

设计算逻指令、取数/存数和相对转移指令的格式。

7.5 RISC技术（自学）

- 7.5.1 RISC的产生和发展
- 7.5.2 RISC的主要特征
- 7.5.3 RISC和CISC的比较

本章重点

1. 指令的格式
2. 操作码扩展技术
3. 数据的存储方式
4. 指令寻址
5. 数据寻址
6. 指令系统设计