

# 2024 计组考试 注意事项

考试时间 6月25日 2024/06/25 14:30-16:30 SX201  
(自己查系统)

请提前半小时到考场准备

李老师答疑时间：6月24日下午14:00-18:00

违规、违纪的处罚，查看学生手册

- 所有作业尽快提交
- 所有研究性学习报告尽快提交
- 所有实验报告尽快提交

纸质材料开卷考试

可能的题型：简答题、计算器、画图题、综合题等  
..

卷面分100分，占总成绩50分

# 2024 计组复习

- 数值表示和运算

- (原码补码反码移码、补码加减法、浮点数四则运算、原码补码乘法…)

- 存储器

- (概念、主存和CPU的连接 …)

- 指令系统

- (10种寻址方式、指令格式 …)

- CPU的结构和功能 控制单元的功能 控制单元的设计

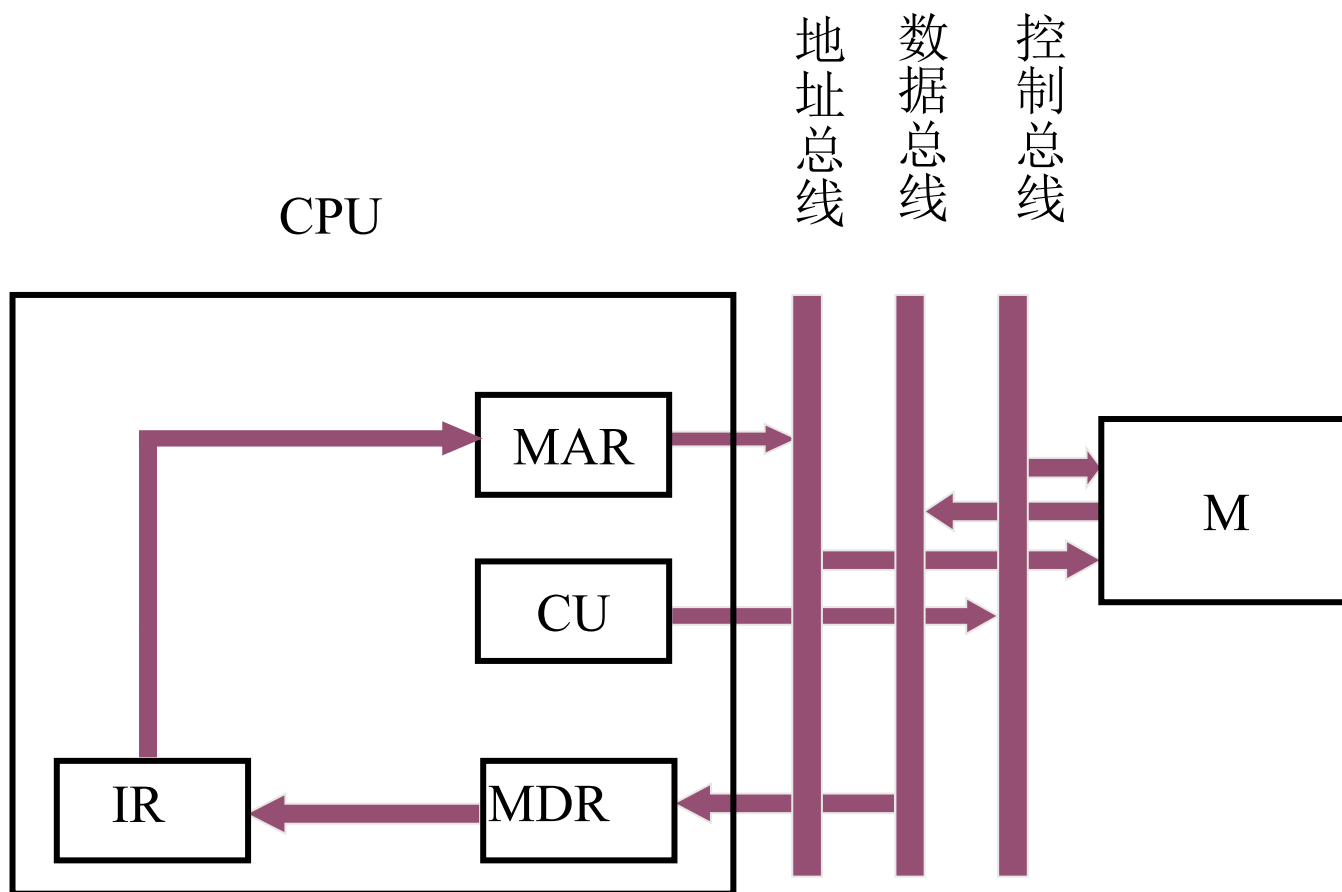
- (概念理解、指令周期、数据通路、微操作命令、微程序等 …)

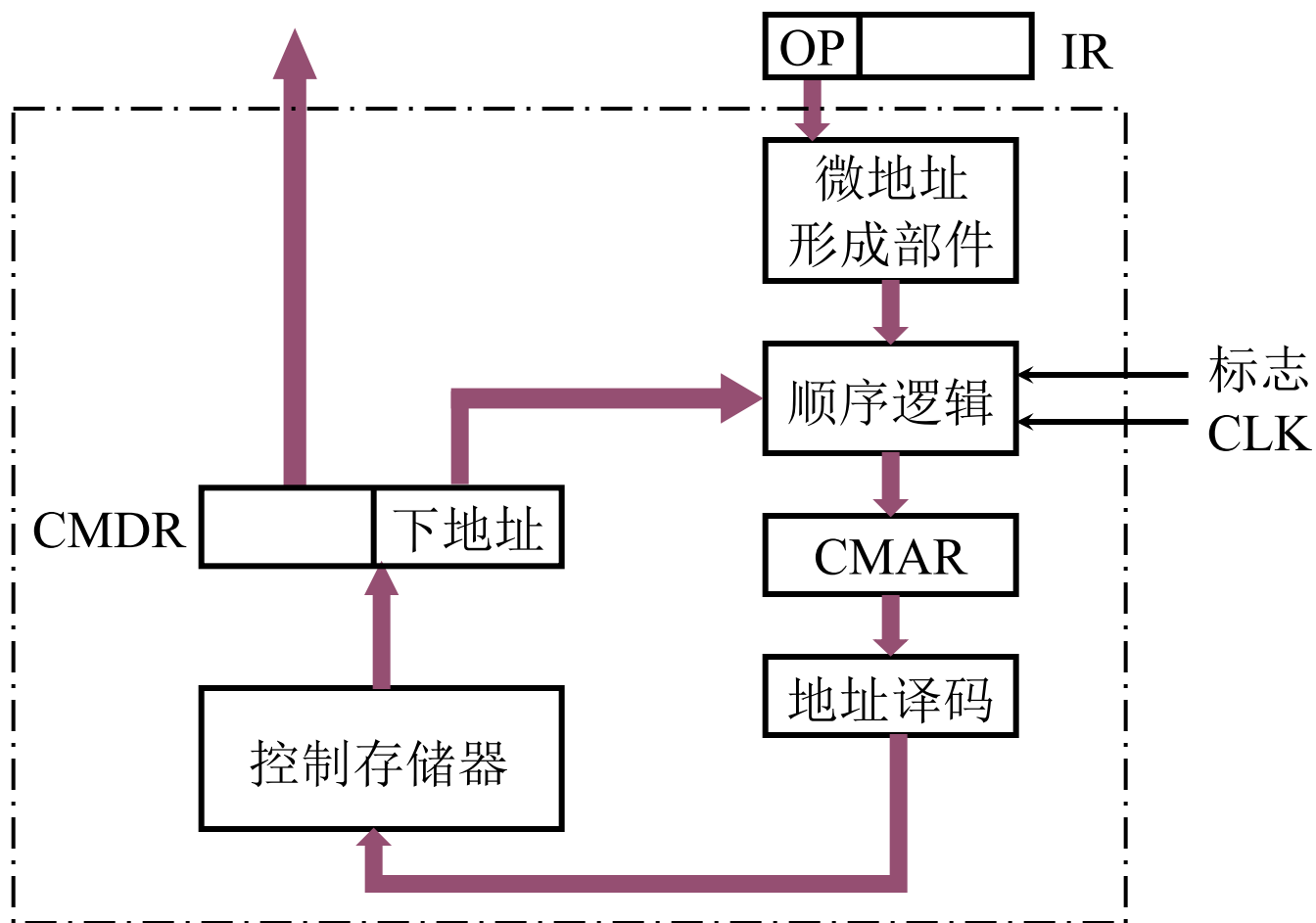
- I/O系统

- (程序查询方式、程序中断方式、DMA方式等 …)

- 总线

- (同步、异步、仲裁)





# 数值表示和运算

- 定点整数和定点小数的概念
- 机器数：原码 补码 反码 移码
- 补码 加减法
- 浮点数表示及加减运算
- 原码1位乘法，补码1位乘法（ Booth算法）

# 原码、补码和反码三种机器数的小结

- Ø 最高位为符号位，书写上用“,”（整数）或“.”（小数）将数值部分和符号位隔开
- Ø 对于正数，原码 = 补码 = 反码
- Ø 对于负数，符号位为 1，其数值部分  
原码除符号位外每位取反末位加 1 → 补码  
原码除符号位外每位取反 → 反码

例6.12 已知  $[y]_{\text{补}}$  求  $[-y]_{\text{补}}$

解：设  $[y]_{\text{补}} = y_0.y_1y_2 \cdots y_n$

<I>  $[y]_{\text{补}} = 0.y_1y_2 \cdots y_n$

$$y = 0.y_1y_2 \cdots y_n$$

$$-y = -0.y_1y_2 \cdots y_n$$

$$[-y]_{\text{补}} = 1.\overline{y_1}\overline{y_2} \cdots \overline{y_n} + 2^{-n}$$

<II>  $[y]_{\text{补}} = 1.y_1y_2 \cdots y_n$

$$[y]_{\text{原}} = 1.\overline{y_1}\overline{y_2} \cdots \overline{y_n} + 2^{-n}$$

$$y = -(0.\overline{y_1}\overline{y_2} \cdots \overline{y_n} + 2^{-n})$$

$$-y = 0.\overline{y_1}\overline{y_2} \cdots \overline{y_n} + 2^{-n}$$

$$[-y]_{\text{补}} = 0.\overline{y_1}\overline{y_2} \cdots \overline{y_n} + 2^{-n}$$

符号位取反，  
数值位取反+1



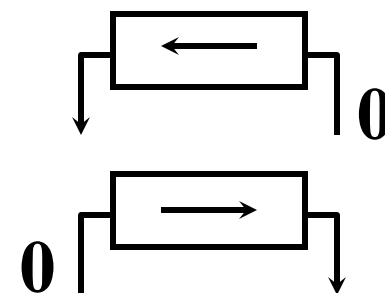
# 算术移位和逻辑移位的区别

算术移位      有符号数的移位

逻辑移位      无符号数的移位

逻辑左移      低位添 0，高位移丢

逻辑右移      高位添 0，低位移丢



例如      01010011

10110010

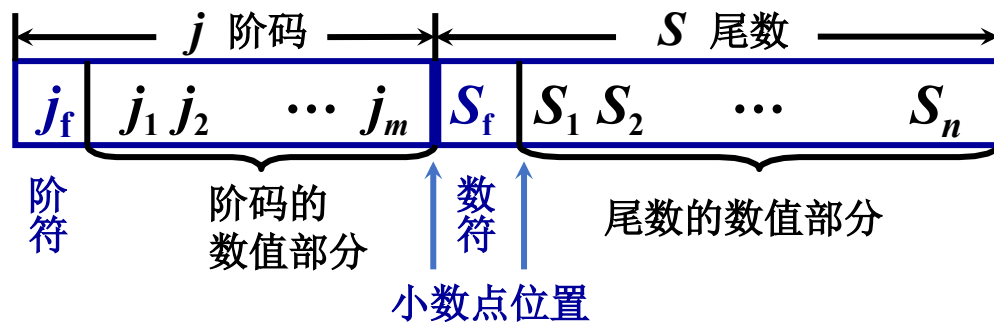
逻辑左移      10100110

逻辑右移      01011001

算术左移      00100110

算术右移      1011001 (补码)

# 浮点数的表示形式



$S_f$  代表浮点数的符号

$n$  其位数反映浮点数的精度

$m$  其位数反映浮点数的表示范围

$j_f$  和  $m$  共同表示小数点的实际位置

# 浮点数的规格化 运算尾数

## (1) 规格化数的定义

$$r = 2 \quad \frac{1}{2} \leq |S| < 1$$

## (2) 规格化数的判断

$S > 0$	规格化形式	$S < 0$	规格化形式
真值	$0.1 \times \times \cdot \times$	真值	$-0.1 \times \times \cdot \times$
原码	$0.1 \times \times \cdot \times$	原码	$1.1 \times \times \cdot \times$
补码	$0.1 \times \times \cdot \times$	补码	$1.0 \times \times \cdot \times$
反码	$0.1 \times \times \cdot \times$	反码	$1.0 \times \times \cdot \times$

原码 不论正数、负数，第一数位为1

补码 符号位和第一数位不同

# 特例

$$S = -\frac{1}{2} = -0.100 \dots 0$$

$$[S]_{\text{原}} = 1.100 \dots 0$$

$$[S]_{\text{补}} = \boxed{1.1}00 \dots 0$$

$\therefore [-\frac{1}{2}]_{\text{补}}$  不是规格化的数

$$S = -1$$

$$[S]_{\text{补}} = \boxed{1.0}00 \dots 0$$

$\therefore [-1]_{\text{补}}$  是规格化的数

### (3) 左规

尾数左移一位，阶码减 1，直到数符和第一数位不同为止

上例  $[x+y]_{\text{补}} = 00, 11; 11. 1001$

左规后  $[x+y]_{\text{补}} = 00, 10; 11. 0010$

$$\therefore x + y = (-0.1110) \times 2^{10}$$

### (4) 右规

当尾数溢出 ( $>1$ ) 时，需右规

即尾数出现  $01. \times \times$  或  $10. \times \times$  时..

尾数右移一位，阶码加 1

# 加法 减法 乘法

- 书上例题和 PPT 熟练 细心
- 加减 p237~p242
- 浮点四则运算 p269~p274
- 乘法 p243~p256
- 结果是否规格化

# 原码两位乘和原码一位乘比较

## 6.3

	原码一位乘	原码两位乘
符号位	$x_0 \oplus y_0$	$x_0 \oplus y_0$
操作数	绝对值	绝对值的补码
移位	逻辑右移	算术右移
移位次数	$n$	$\frac{n}{2}$ ( $n$ 为偶数)
最多加法次数	$n$	$\frac{n}{2}+1$ ( $n$ 为偶数)

思考  $n$  为奇数时，原码两位乘 移？次 最多加？次



# Booth 算法递推公式

## 6.3

$$[z_0]_{\text{补}} = 0$$

$$[z_1]_{\text{补}} = 2^{-1} \{ (y_{n+1} - y_n) [x]_{\text{补}} + [z_0]_{\text{补}} \} \quad y_{n+1} = 0$$

⋮

$$[z_n]_{\text{补}} = 2^{-1} \{ (y_2 - y_1) [x]_{\text{补}} + [z_{n-1}]_{\text{补}} \}$$

$$[x \cdot y]_{\text{补}} = [z_n]_{\text{补}} + (y_1 - y_0) [x]_{\text{补}} \quad \text{最后一步不移位}$$

如何实现  
 $y_{i+1} - y_i$  ?

$y_i$	$y_{i+1}$	$y_{i+1} - y_i$	操作
0	0	0	$\rightarrow 1$
0	1	1	$+ [x]_{\text{补}} \rightarrow 1$
1	0	-1	$+ [-x]_{\text{补}} \rightarrow 1$
1	1	0	$\rightarrow 1$





例6.23 已知  $x = +0.0011$   $y = -0.1011$  求  $[x \cdot y]_{\text{补}}$  **6.3**

解: 
$$\begin{array}{r|l|l} 00.0000 & 1.0101 & 0 \\ + 11.1101 & & +[-x]_{\text{补}} \end{array}$$

补码  
右移

$$\begin{array}{r|l|l} 11.1101 & & \\ 11.1101 & 1.1010 & \rightarrow 1 \\ + 00.0011 & & +[x]_{\text{补}} \end{array}$$

补码  
右移

$$\begin{array}{r|l|l} 00.0001 & 1 & \\ 00.0000 & 11.1010 & \rightarrow 1 \\ + 11.1101 & & +[-x]_{\text{补}} \end{array}$$

补码  
右移

$$\begin{array}{r|l|l} 11.1101 & 11 & \\ 11.1101 & 111.101 & \rightarrow 1 \\ + 00.0011 & & +[x]_{\text{补}} \end{array}$$

补码  
右移

$$\begin{array}{r|l|l} 00.0001 & 111 & \\ 00.0000 & 1111.10 & \rightarrow 1 \\ + 11.1101 & & +[-x]_{\text{补}} \end{array}$$

$11.1101$  |  $1111$  | 最后一步不移位

$$[x]_{\text{补}} = 0.0011$$

$$[y]_{\text{补}} = 1.0101$$

$$[-x]_{\text{补}} = 1.1101$$

$$\therefore [x \cdot y]_{\text{补}} = 1.11011111$$



# 浮点四则运算

## 一、浮点加减运算

$$x = S_x \cdot 2^{j_x} \quad y = S_y \cdot 2^{j_y}$$

### 1. 对阶

#### (1) 求阶差

$$\Delta j = j_x - j_y = \begin{cases} = 0 & j_x = j_y & \text{已对齐} \\ > 0 & j_x > j_y & \begin{cases} x \text{ 向 } y \text{ 看齐} & S_x \leftarrow 1, j_x - 1 \\ y \text{ 向 } x \text{ 看齐} & \ddot{\text{ü}} S_y \rightarrow 1, j_y + 1 \end{cases} \\ < 0 & j_x < j_y & \begin{cases} x \text{ 向 } y \text{ 看齐} & \ddot{\text{ü}} S_x \rightarrow 1, j_x + 1 \\ y \text{ 向 } x \text{ 看齐} & S_y \leftarrow 1, j_y - 1 \end{cases} \end{cases}$$

#### (2) 对阶原则

小阶向大阶看齐



# 浮点乘除运算

$$x = S_x \cdot 2^{j_x} \quad y = S_y \cdot 2^{j_y}$$

## 1. 乘法

$$x \cdot y = (S_x \cdot S_y) \times 2^{j_x + j_y}$$

## 2. 步骤

- (1) 阶码采用 补码定点加（乘法） 减（除法） 运算
- (2) 尾数乘除同 定点 运算
- (3) 规格化

## 3. 浮点运算部件

阶码运算部件， 尾数运算部件



# 第4章 存储器

## 4.1 概述

## 4.2 主存储器

## 4.3 高速缓冲存储器

## 4.4 辅助存储器



# 存储器

## 存储器芯片和CPU连接

- 芯片选择范围
- 连接设计
- 等

## 高速缓冲存储器Cache

Cache 命中 Cache Miss

## 主存与Cache的地址映射关系

- (1) 直接相联映射
- (2) 全相联映射
- (3) 组相联映射

# 存储器与 CPU 的连接

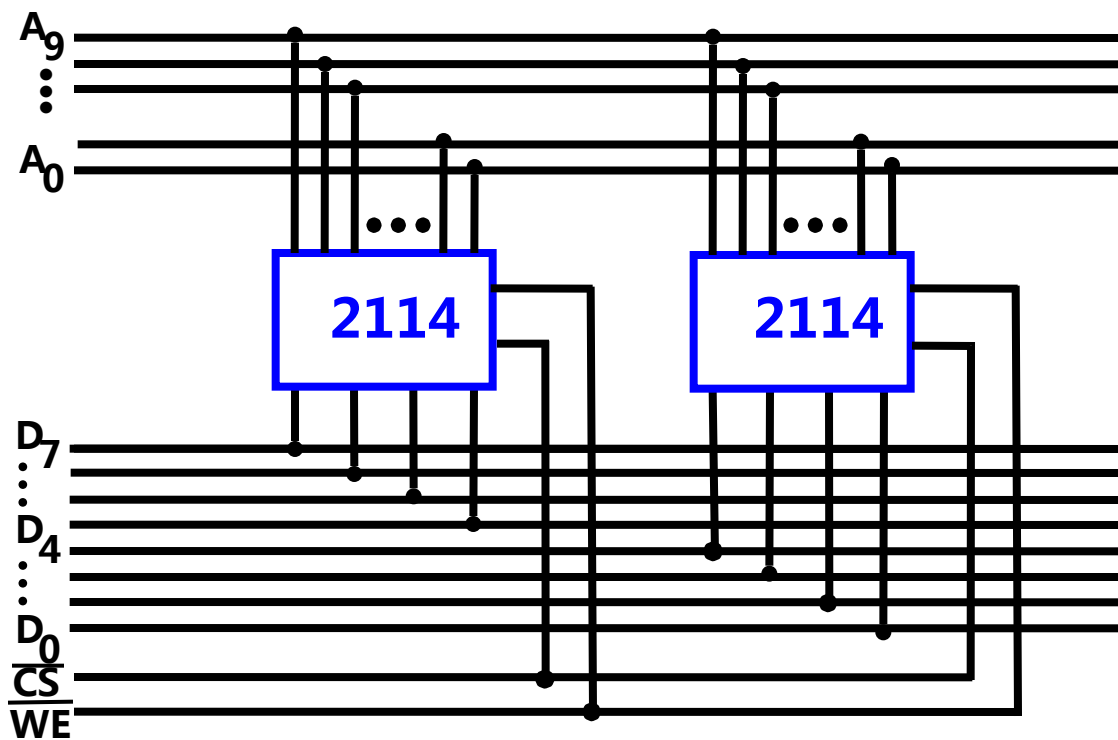
## 1. 存储器容量的扩展

### (1) 位扩展(增加存储字长)

用 ? 片  $1K \times 4$  位 存储芯片组成  $1K \times 8$  位的存储器

10根地址线

8根数据线

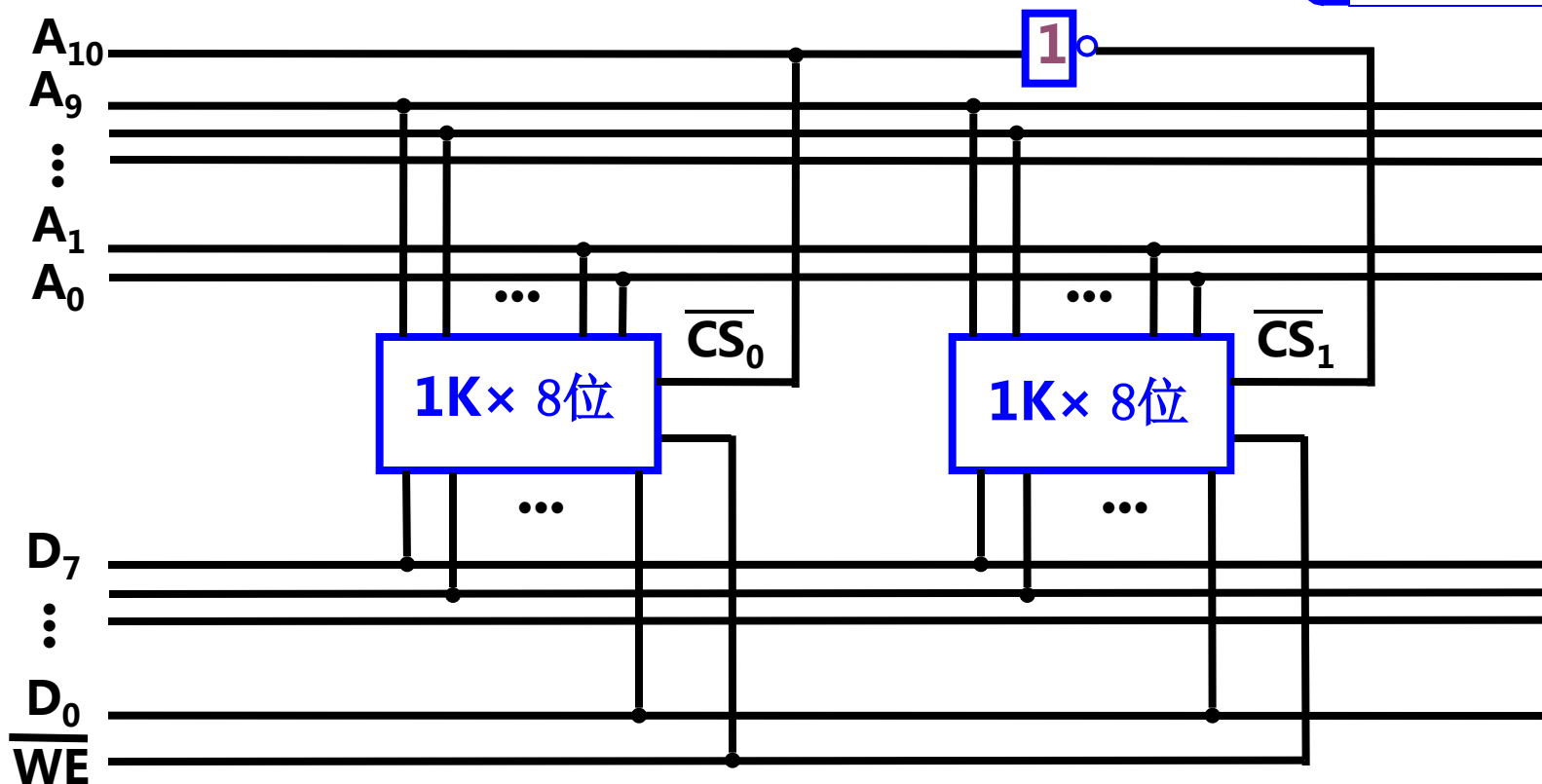


## (2) 字扩展（增加存储字的数量）

用 ? 片  $1\text{K} \times 8$  位 存储芯片组成  $2\text{K} \times 8$  位的存储器

11根地址线

8根数据线

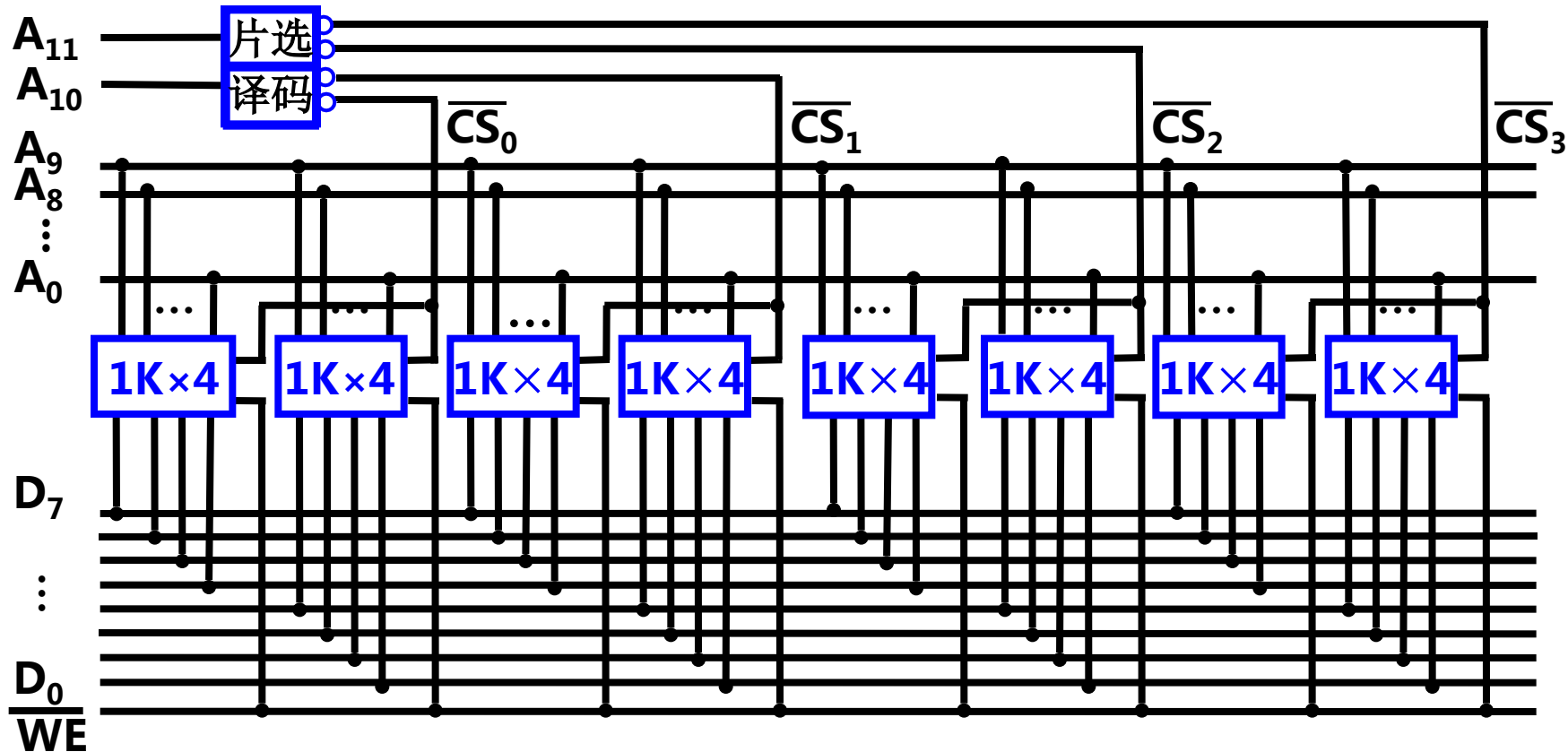


### (3) 字、位扩展

用 ? 片  $1\text{K} \times 4$  位 存储芯片组成  $4\text{K} \times 8$  位的存储器

12根地址线

8根数据线





# 存储器芯片和CPU连接

先把地址列出来 先把地址列出来 先把地址列出来

再：

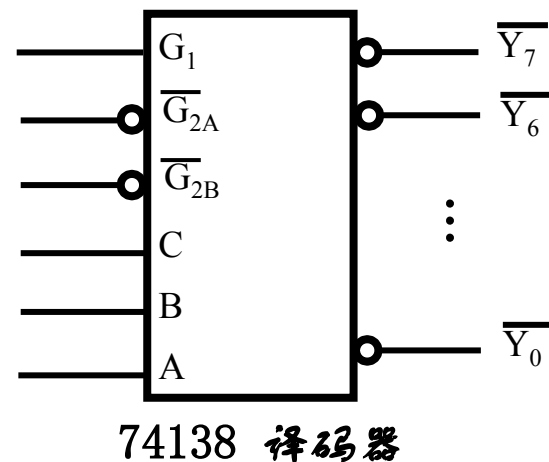
(1) 选择芯片型号及个数

(2) 连接地址线

(3) 连接数据线

(4) 连接读写控制线

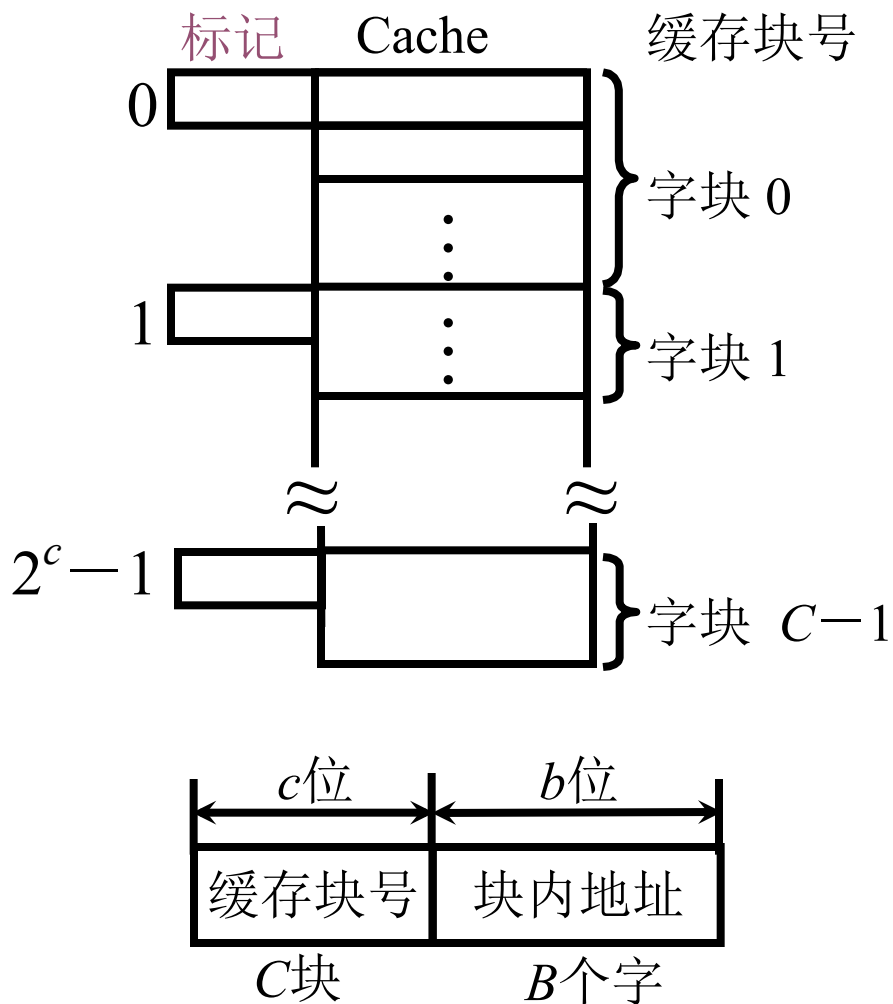
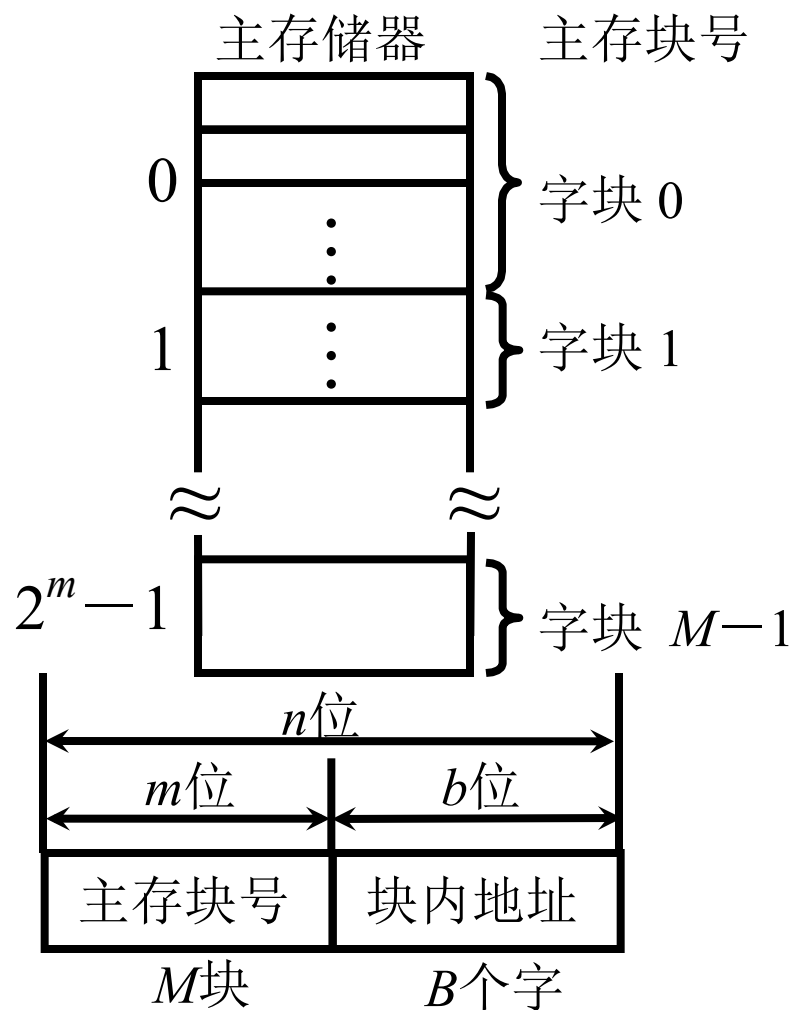
(5) 连接片选线（译码器一定要对应；对于每一个芯片所有的地址线都需要用到，面对芯片和译码器的细节）



# Cache 的工作原理

## 4.3

### (1) 主存和缓存的编址



主存和缓存按块存储

块的大小相同

$B$  为块长



## (2) 命中与未命中

缓存共有  $C$  块

主存共有  $M$  块  $M \gg C$

命中      主存块 调入 缓存

主存块与缓存块 建立 了对应关系

用 标记记录 与某缓存块建立了对应关系的 主存块号

未命中      主存块 未调入 缓存

主存块与缓存块 未建立 对应关系



### (3) Cache 的命中率

CPU 欲访问的信息在 Cache 中的 **比率**

**命中率** 与 Cache 的 **容量** 与 **块长** 有关

一般每块可取 4 ~ 8 个字

**块长**取一个存取周期内从主存调出的信息长度

CRAY_1	16体交叉	块长取 16 个存储字
IBM 370/168	4体交叉	块长取 4 个存储字
(64位×4 = 256位)		



## (4) Cache –主存系统的效率

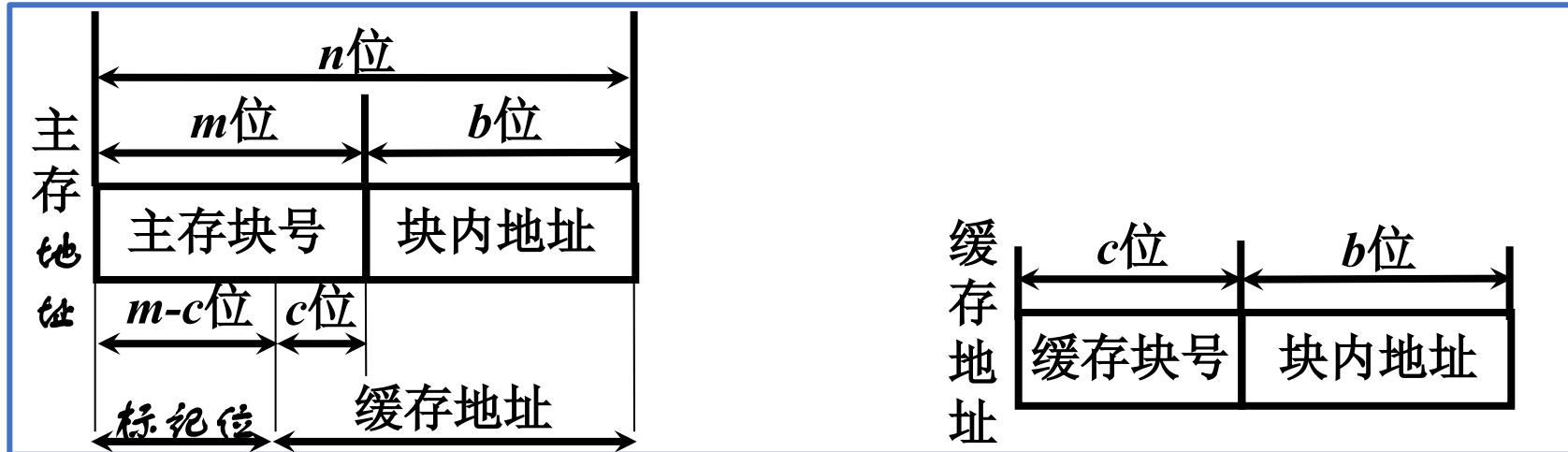
效率  $e$  与 命中率 有关

$$e = \frac{\text{访问 Cache 的时间}}{\text{平均访问时间}} \times 100\%$$

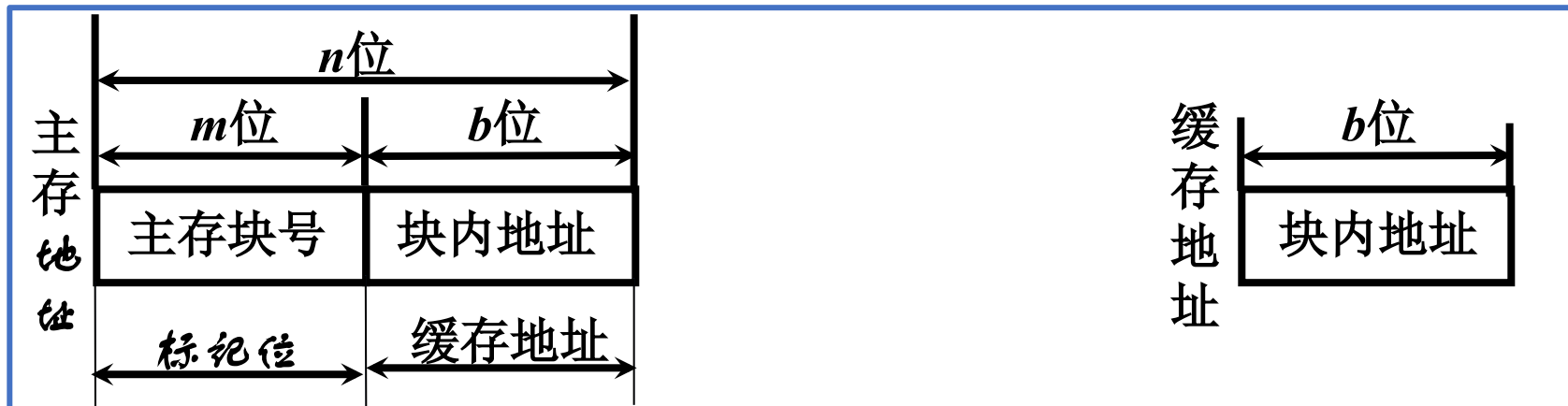
设 Cache 命中率为  $h$ ，访问 Cache 的时间为  $t_c$ ，  
访问 主存 的时间为  $t_m$

$$\text{则 } e = \frac{t_c}{h \times t_c + (1-h) \times t_m} \times 100\%$$

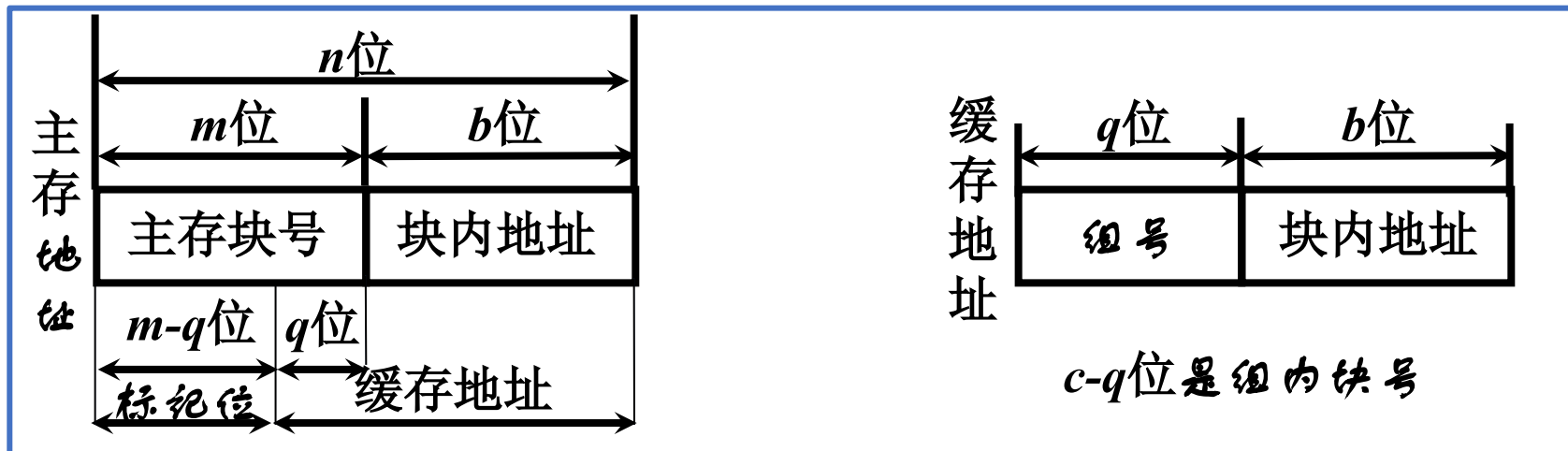




直接相联映射



全相联映射

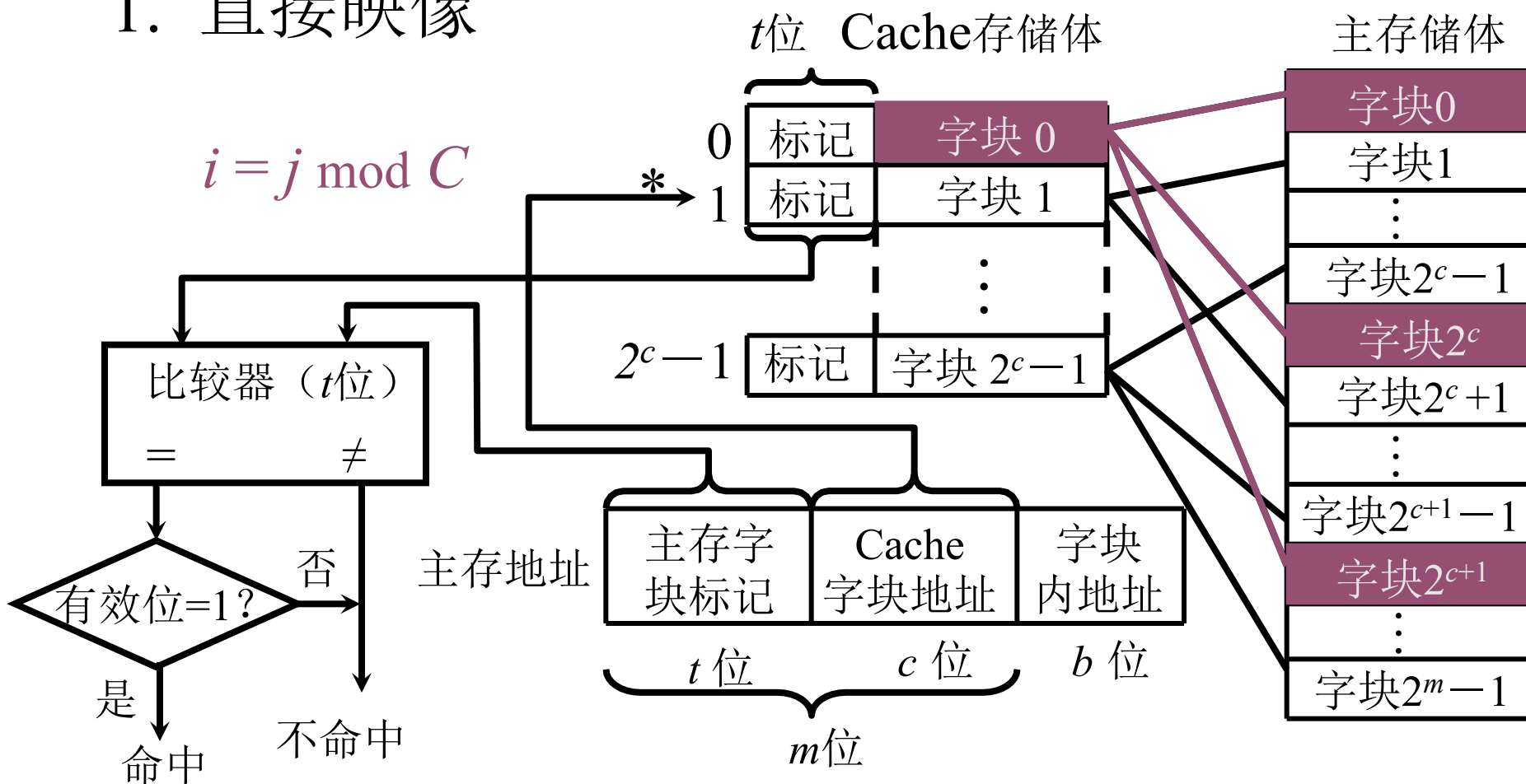


组相联映射

# Cache – 主存的地址映射

## 4.3

### 1. 直接映像

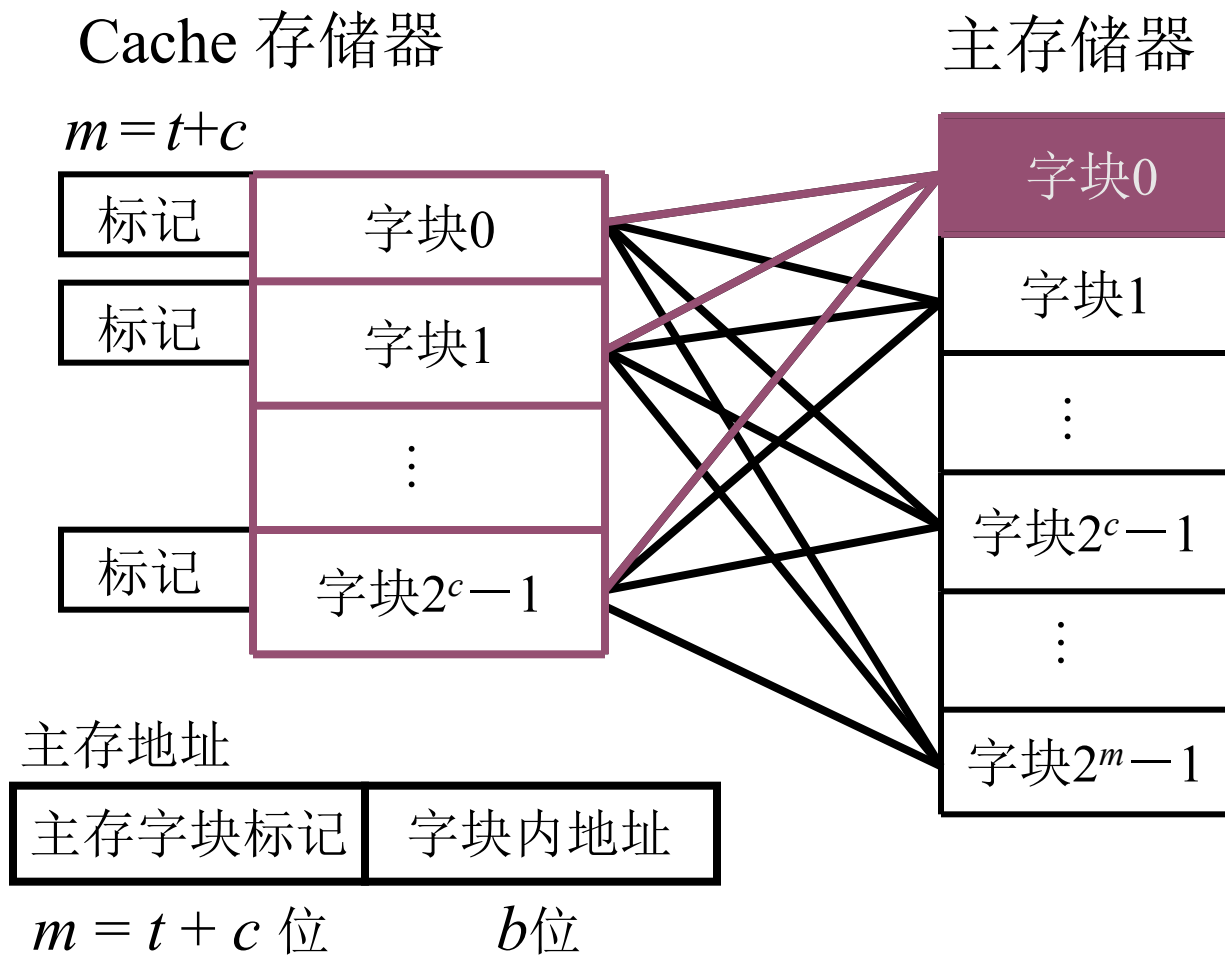


每个缓存块  $i$  可以和若干个主存块对应

每个主存块  $j$  只能和一个缓存块对应



## 2. 全相联映射

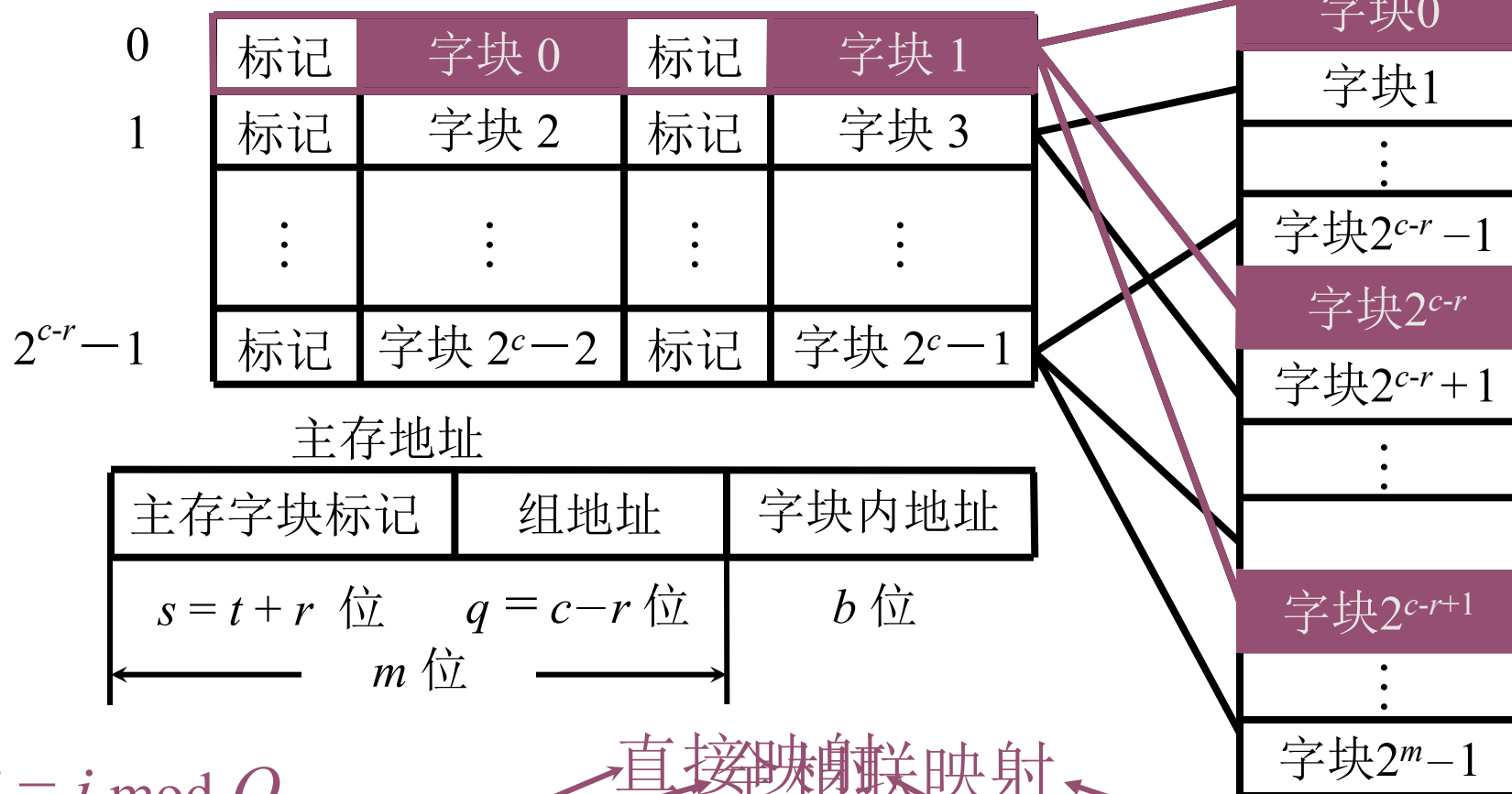


主存 中的 任一块 可以映射到 缓存 中的 任一块



## 3. 组相联映射

主存储器

组 Cache 共  $Q$  组，每组内两块 ( $r=1$ )

$$i = j \bmod Q$$

直接组相联映射

某一主存块  $j$  按模  $Q$  映射到 缓存 的第  $i$  组中的 任一块



# 指令系统

## 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



# 指令系统

- 指令 指令格式 指令周期 (几个)
- 形式地址, 有效地址 EA
- 指令寻址 2种
- 数据寻址 10种

## 7.3 寻址方式

寻址方式 确定 本条指令 的 操作数地址  
下一条 欲执行 指令 的 指令地址

寻址方式 { 指令寻址  
数据寻址



# 数据寻址

操作码	寻址特征	形式地址 A
-----	------	--------

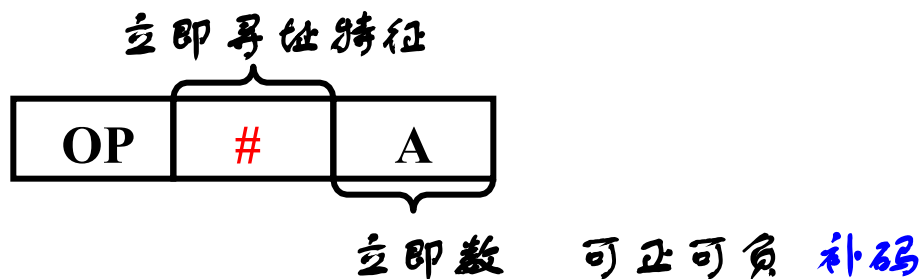
形式地址                  指令字中的地址

有效地址                  操作数的真实地址

约定    指令字长 = 存储字长 = 机器字长

## 1. 立即寻址

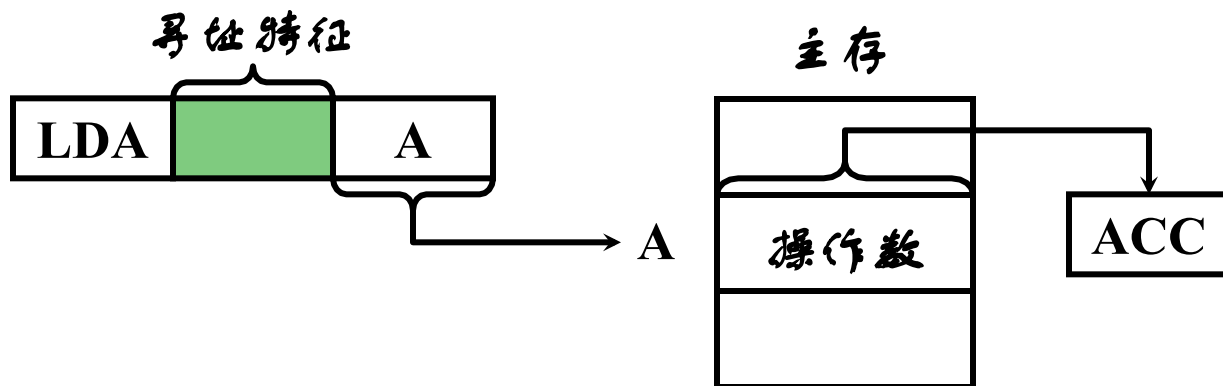
形式地址 A 就是操作数



- 指令执行阶段不访存
- A 的位数限制了立即数的范围

## 2. 直接寻址

$EA = A$       有效地址由形式地址直接给出



- 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

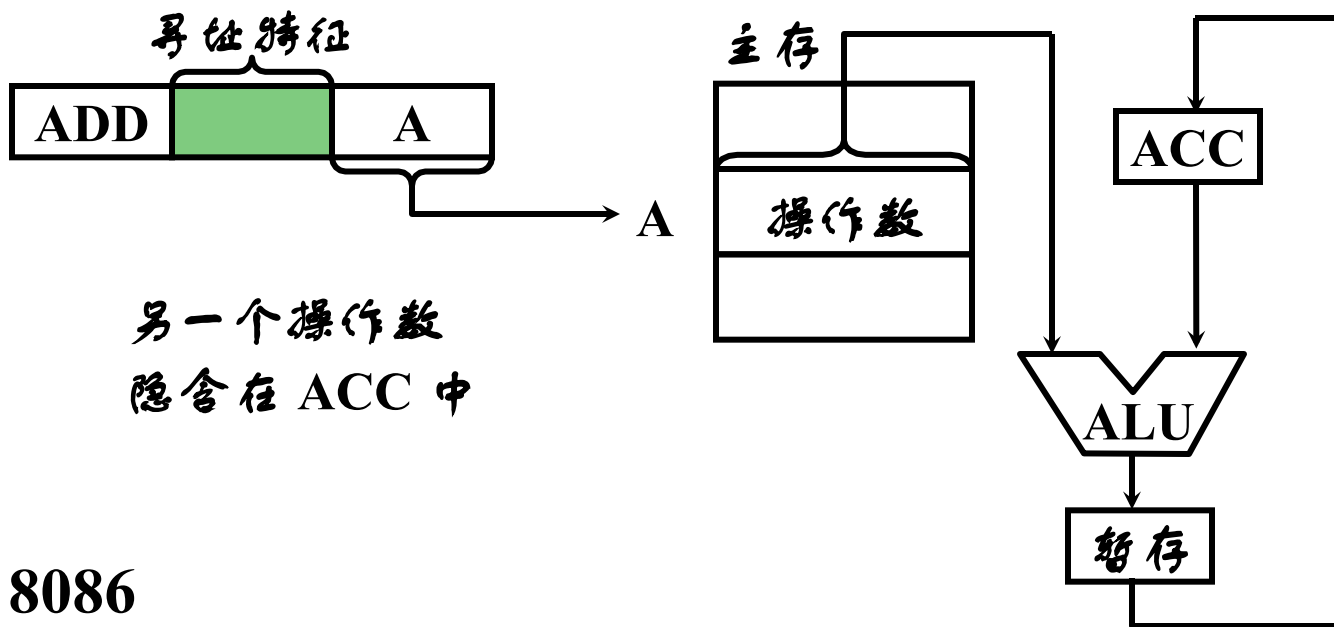
STAD **D** 2000

把ACC的内容存储到地址为2000的内存单元之中

**D**表示直接寻址

### 3. 隐含寻址

操作数地址隐含在操作码中



如 8086

MUL 指令 被乘数隐含在 AX (16位) 或 AL (8位) 中

MOVS 指令 源操作数的地址隐含在 SI 中

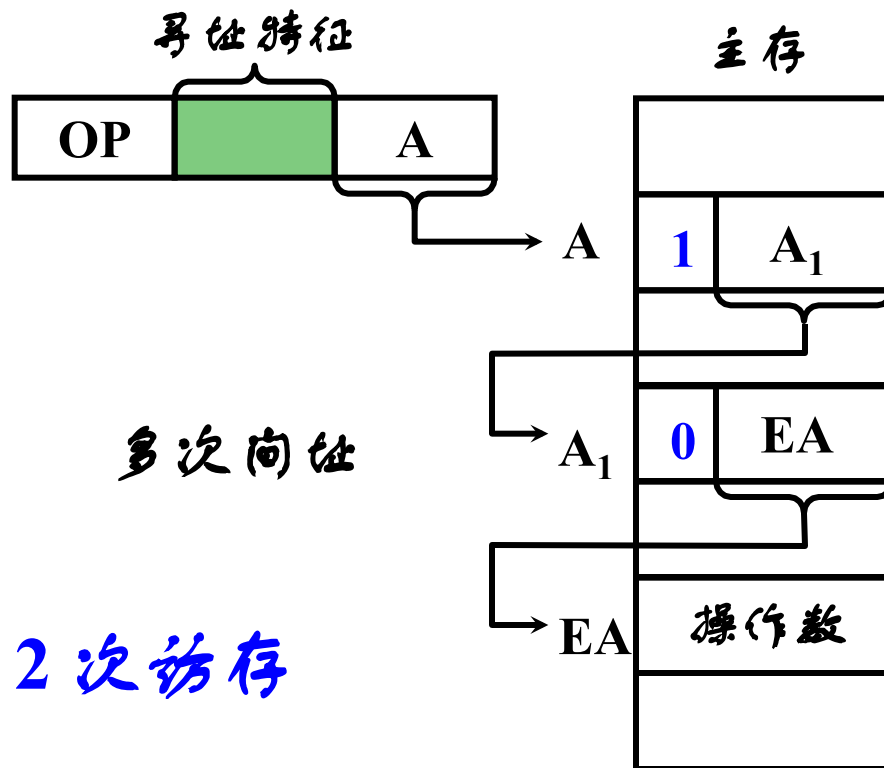
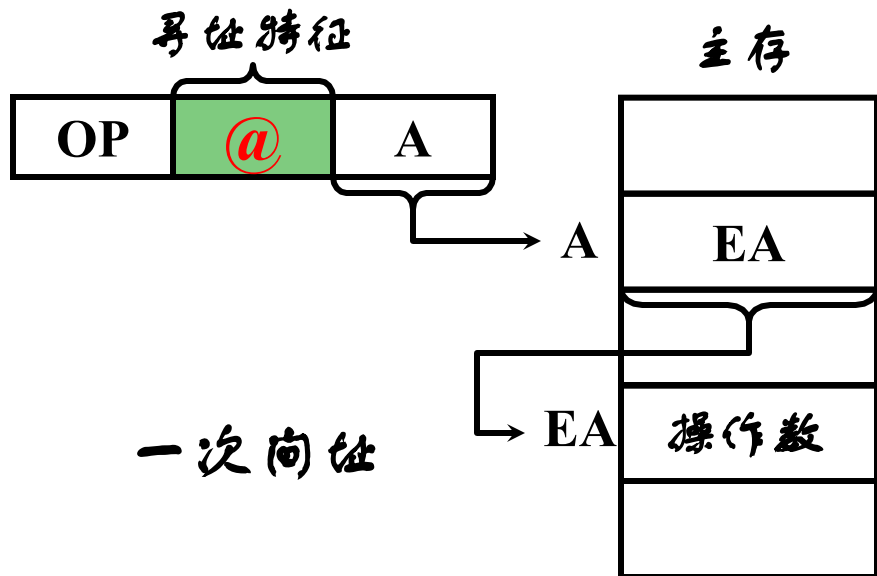
目的操作数的地址隐含在 DI 中

- 指令字中少了一个地址字段，可缩短指令字长

## 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

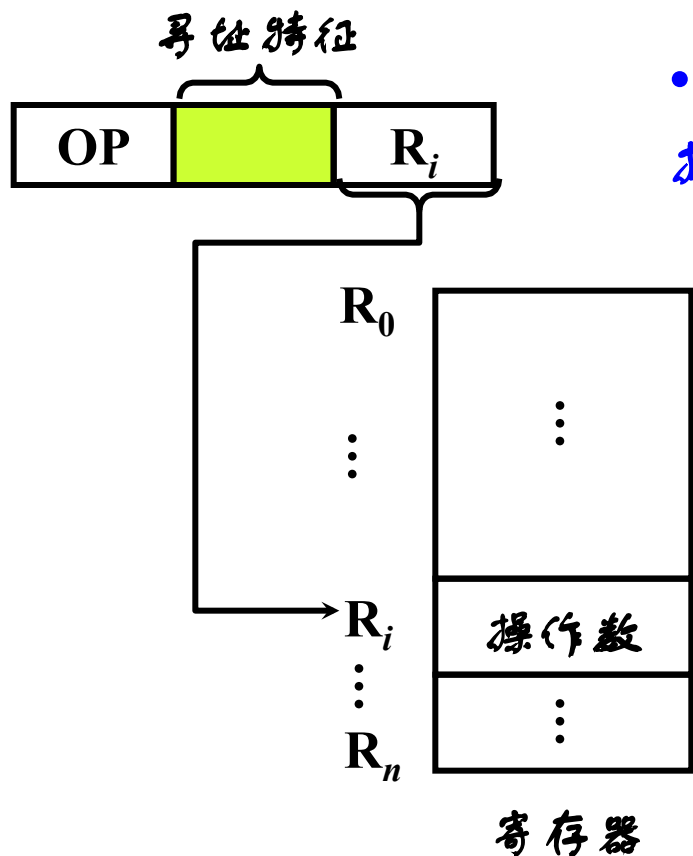
多次访存



## 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，可缩短指令字长

ADD R0,R1

表示  $R0 \leftarrow R0 + R1$

R代表寄存器寻址

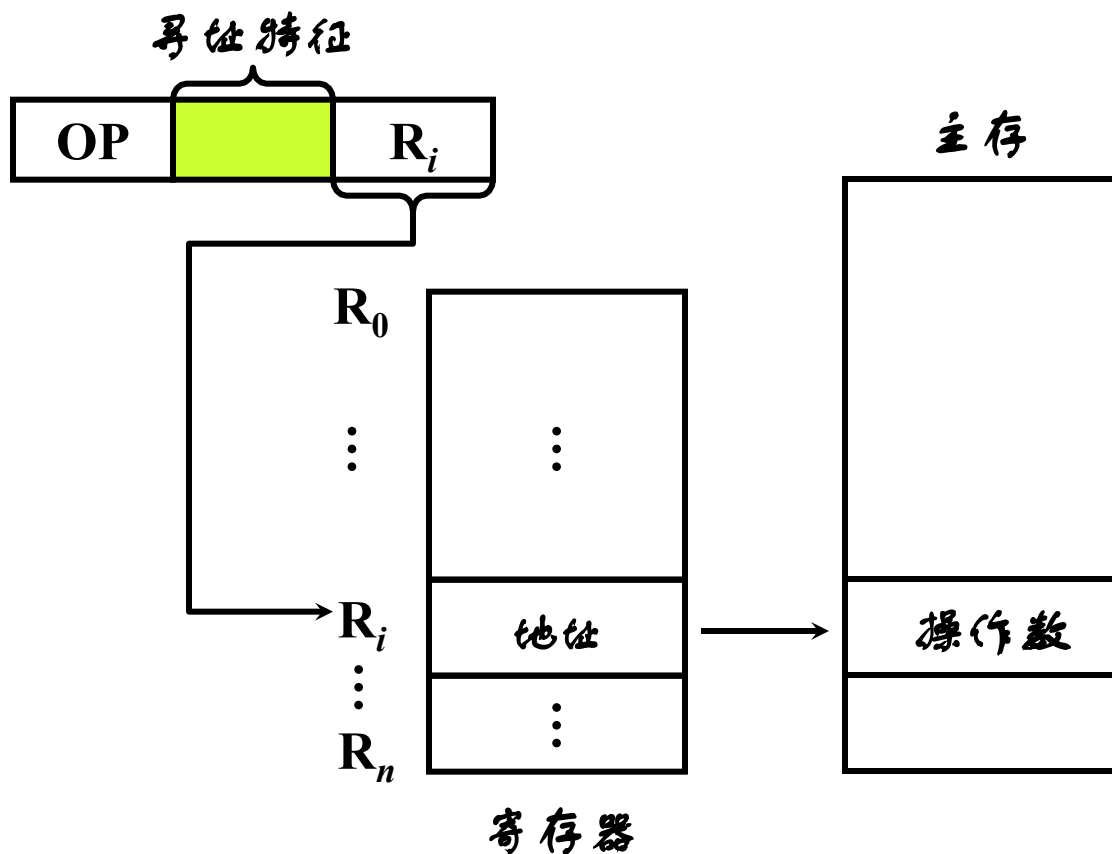
MVRR R0,R1

把寄存器R1的内容传送到寄存器R0;

## 6. 寄存器间接寻址

$$EA = (R_i)$$

有效地址在寄存器中

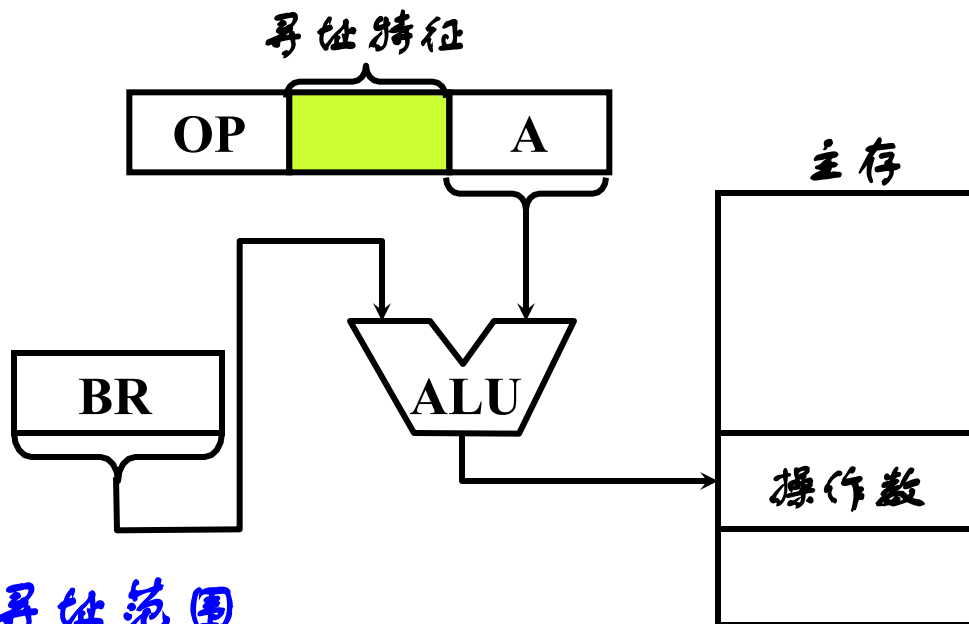


STRR [R8],R9 把R9的内容传送到以寄存器R8的内容为地址的内存单元之中；R字母两侧加上方括号，代表寄存器间接寻址，

## 7. 基址寻址

### (1) 采用专用寄存器作基址寄存器

$$EA = (BR) + A \quad BR \text{ 为基址寄存器}$$

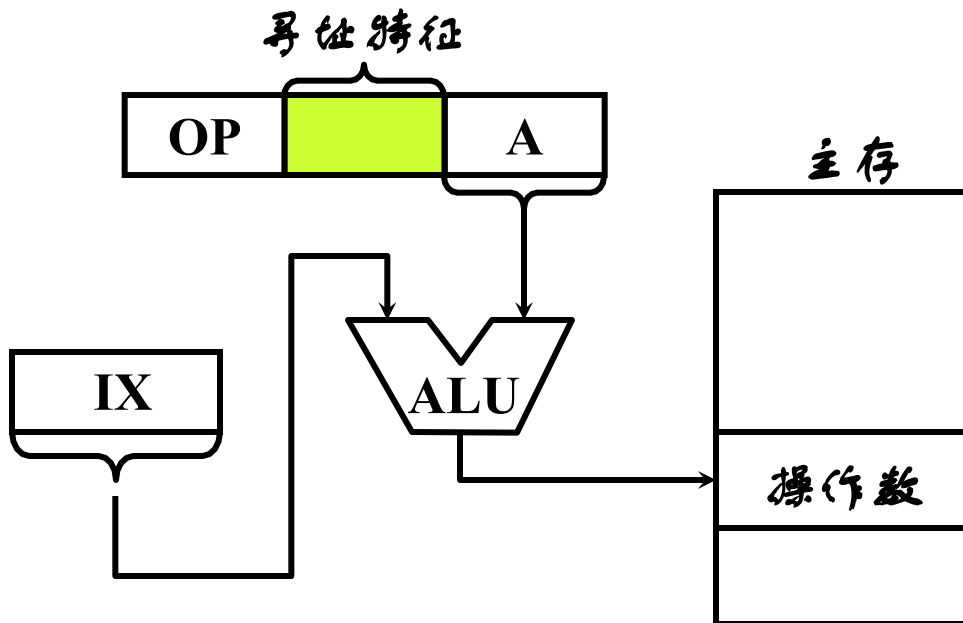


- 可扩大寻址范围
- 有利于多道程序
- BR 内容由操作系统或管理程序确定
- 在程序的执行过程中 BR 内容不变，形式地址 A 可变

## 8. 变址寻址

$EA = (IX) + A$      $IX$  为变址寄存器 (专用)

通用寄存器也可以作为变址寄存器

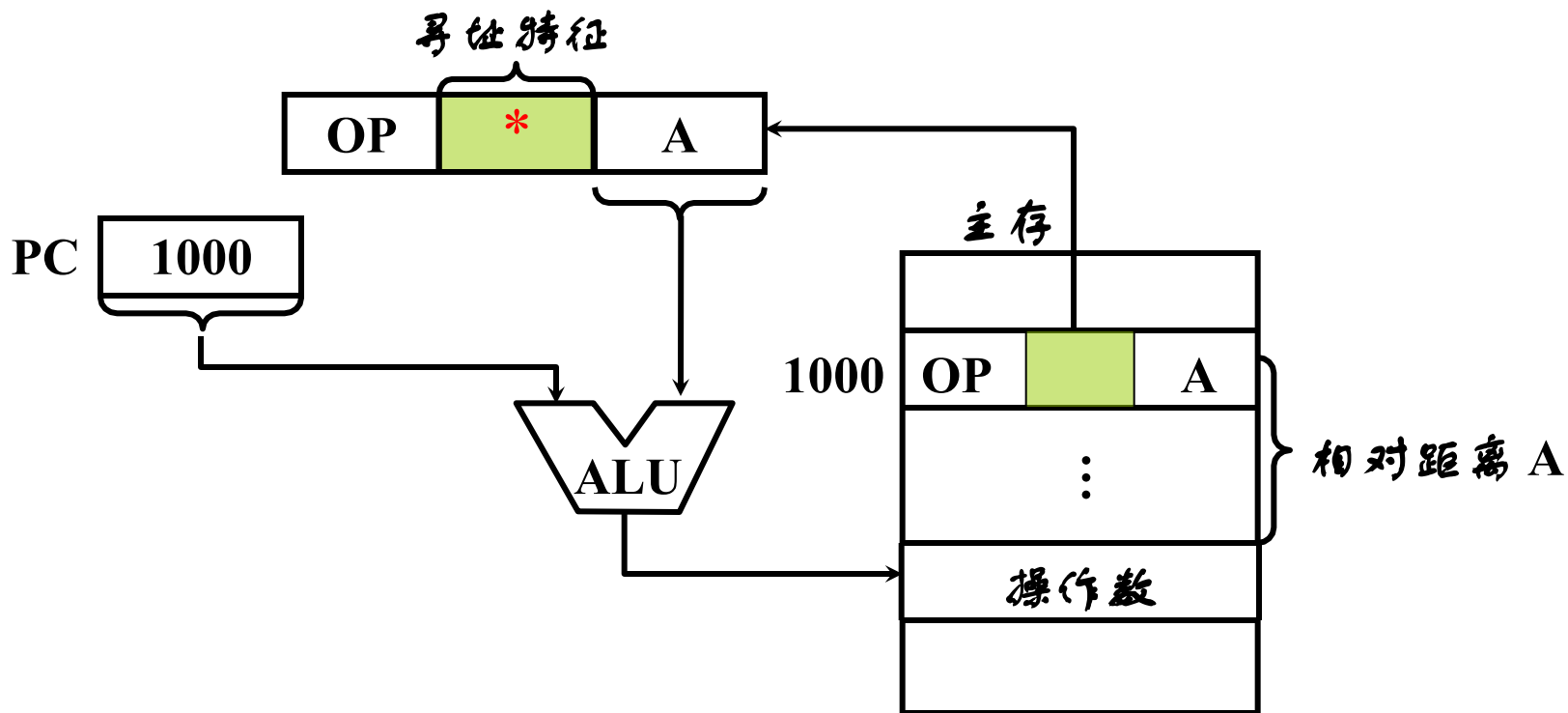


- 可扩大寻址范围
- $IX$  的内容由用户给定
- 在程序的执行过程中  $IX$  内容可变, 形式地址  $A$  不变
- 便于处理数组问题

## 9. 相对寻址

$$EA = (PC) + A$$

PC取值后自增，A 是相对于当前指令的位移量（可正可负，补码）



- A 的位数决定操作数的寻址范围
- 程序浮动
- 广泛用于转移指令

# CPU相关三章-中断系统

- 中断系统 基本概念
- 中断请求 与 中断判优
  - 排队器
  - 屏蔽字
  - 书上例题

# 微操作命令

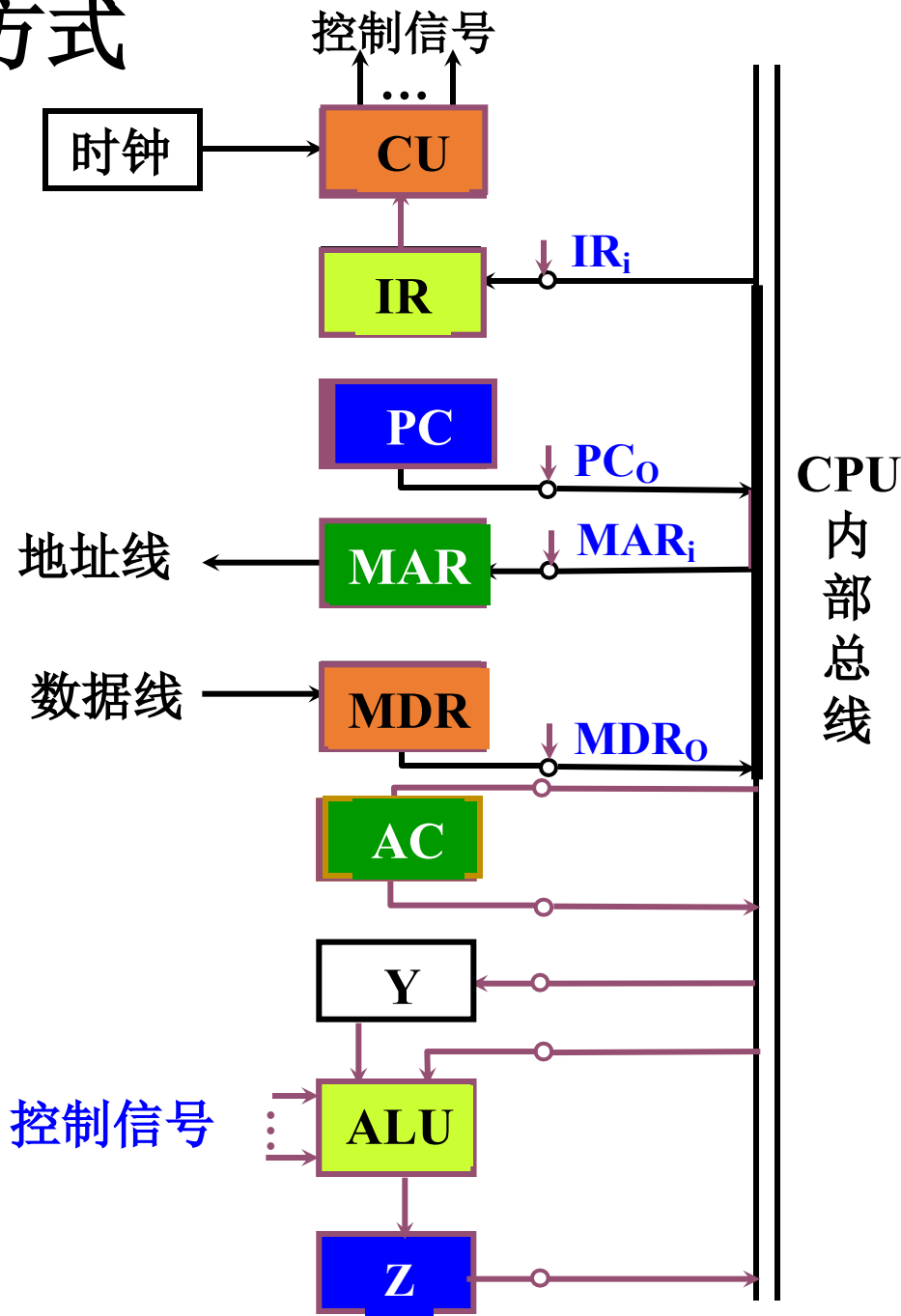
一个完整的机器周期包括（指令周期）

- 取值周期
- 寻址周期
- 执行周期
  - 非访存指令
  - 访存指令
  - 转移指令
- 中断周期

# 采用 CPU 内部总线方式

## ADD @ X 取指周期

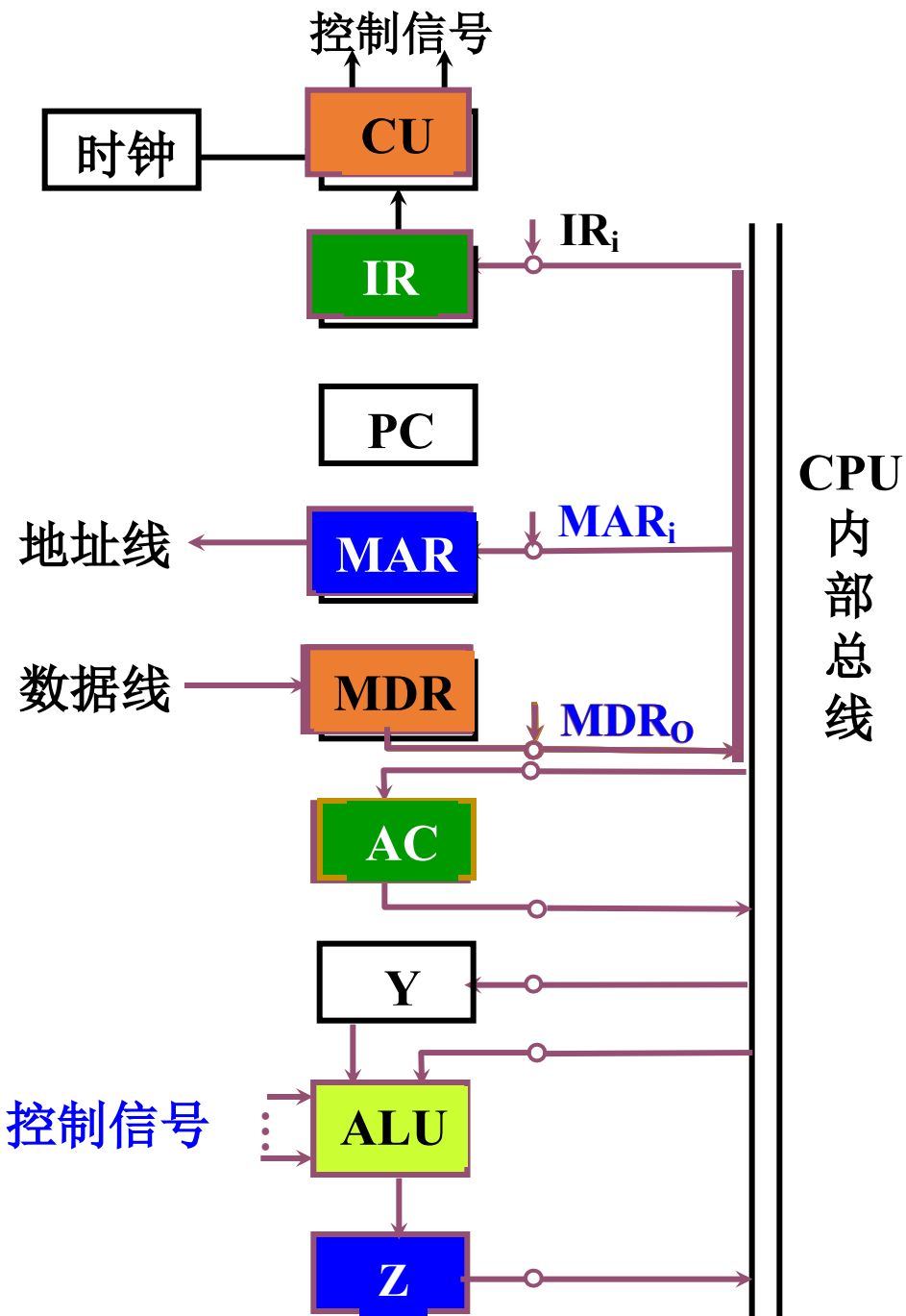
- PC  $\longrightarrow$  MAR  $\longrightarrow$  地址线  
 $PC_0$   $MAR_i$
- 1  $\longrightarrow$  R
- 数据线  $\longrightarrow$  MDR
- MDR  $\longrightarrow$  IR  
 $MDR_0$   $IR_i$
- OP (IR)  $\longrightarrow$  CU
- (PC) + 1  $\longrightarrow$  PC





# ADD @ X 间址周期

- MDR  $\longrightarrow$  MAR  $\longrightarrow$  地址线  
MDR<sub>0</sub> MAR<sub>i</sub>
- 1  $\longrightarrow$  R
- 数据线  $\longrightarrow$  MDR
- MDR  $\longrightarrow$  Ad(IR)  
MDR<sub>0</sub> IR<sub>i</sub>



# 9.1 操作命令的分析

## 一、取指周期

PC  $\rightarrow$  MAR  $\rightarrow$  地址线

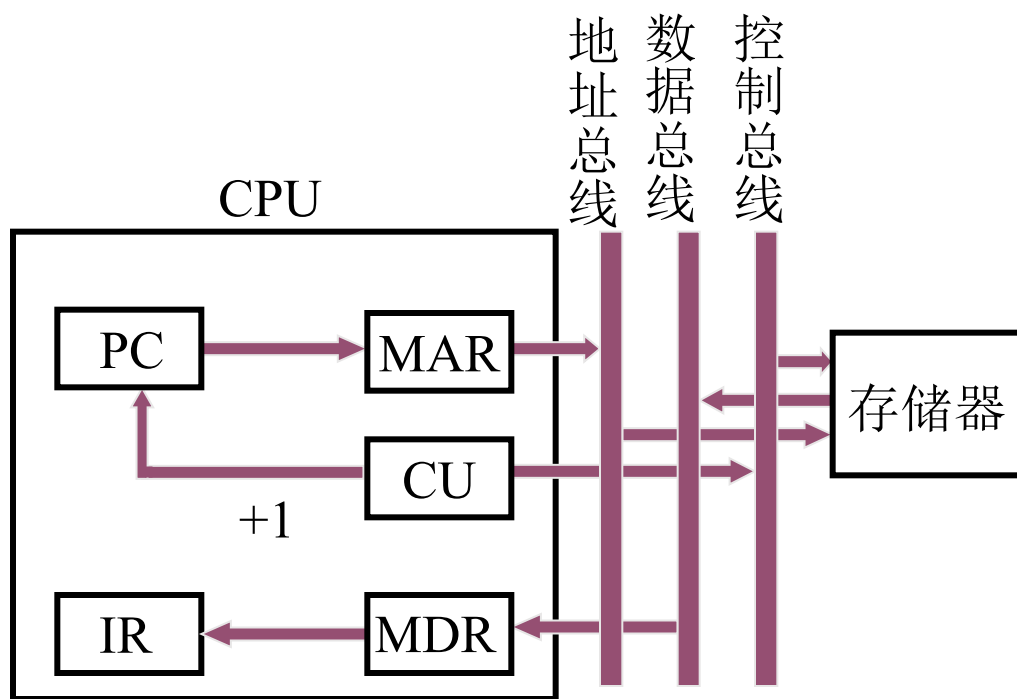
1  $\rightarrow$  R

M ( MAR )  $\rightarrow$  MDR

MDR  $\rightarrow$  IR

OP ( IR )  $\rightarrow$  CU

( PC ) + 1  $\rightarrow$  PC



## 二、间址周期

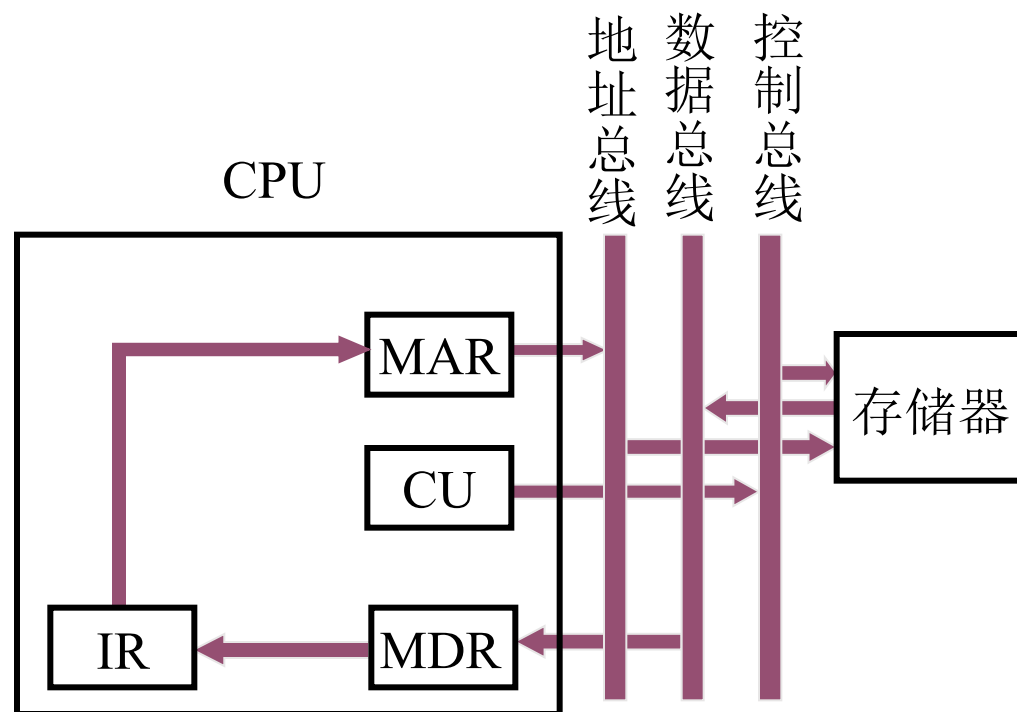
指令形式地址  $\longrightarrow$  MAR

$Ad(IR) \longrightarrow MAR$

$1 \longrightarrow R$

$M(MAR) \longrightarrow MDR$

$MDR \longrightarrow Ad(IR)$



## 2. 访存指令

### (1) 加法指令

ADD X

$\text{Ad(IR)} \rightarrow \text{MAR}$

$1 \rightarrow R$

$\text{M(MAR)} \rightarrow \text{MDR}$

$(\text{ACC}) + (\text{MDR}) \rightarrow \text{ACC}$

### (2) 存数指令

STA X

$\text{Ad(IR)} \rightarrow \text{MAR}$

$1 \rightarrow W$

$\text{ACC} \rightarrow \text{MDR}$

$\text{MDR} \rightarrow \text{M(MAR)}$



(3) 取数指令 **LDA X**

$$\text{Ad ( IR )} \rightarrow \text{MAR}$$

$$1 \rightarrow \text{R}$$

$$\text{M ( MAR )} \rightarrow \text{MDR}$$

$$\text{MDR} \rightarrow \text{ACC}$$

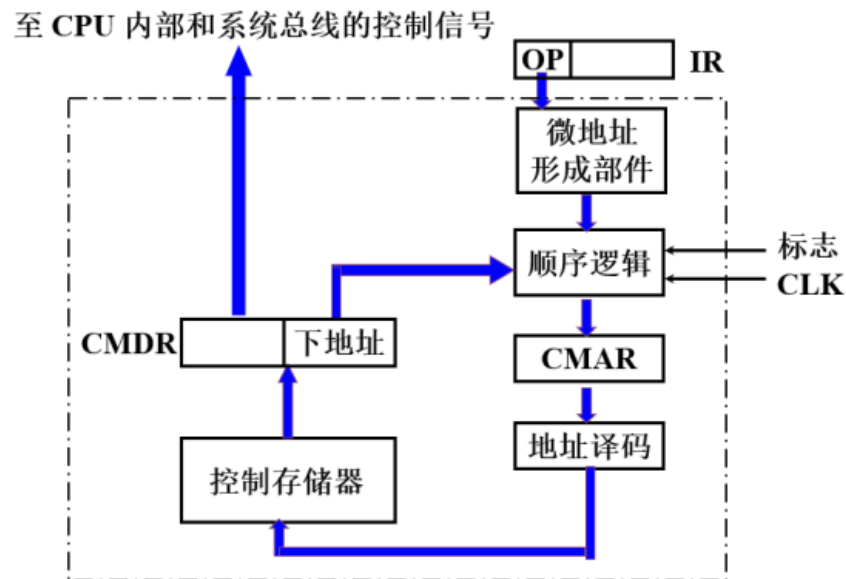
## 3. 转移指令

(1) 无条件转 **JMP X**

$$\text{Ad ( IR )} \rightarrow \text{PC}$$
(2) 条件转移 **BAN X** (负则转)
$$\text{A}_0 \cdot \text{Ad ( IR )} + \bar{\text{A}}_0 ( \text{PC} ) \rightarrow \text{PC}$$


微程序设计

- 微程序控制的基本概念
  - 微指令和微操作
  - 微指令编码方式及格式
  - 操作控制字段，下地址字段
- 工作基本原理
  - 微程序控制器，微程序控制
- 微程序设计节拍安排
- 组合逻辑设计的节拍安排



# 例如：取指阶段的微操作及节拍安排

考虑到需要 形成后续微指令的地址

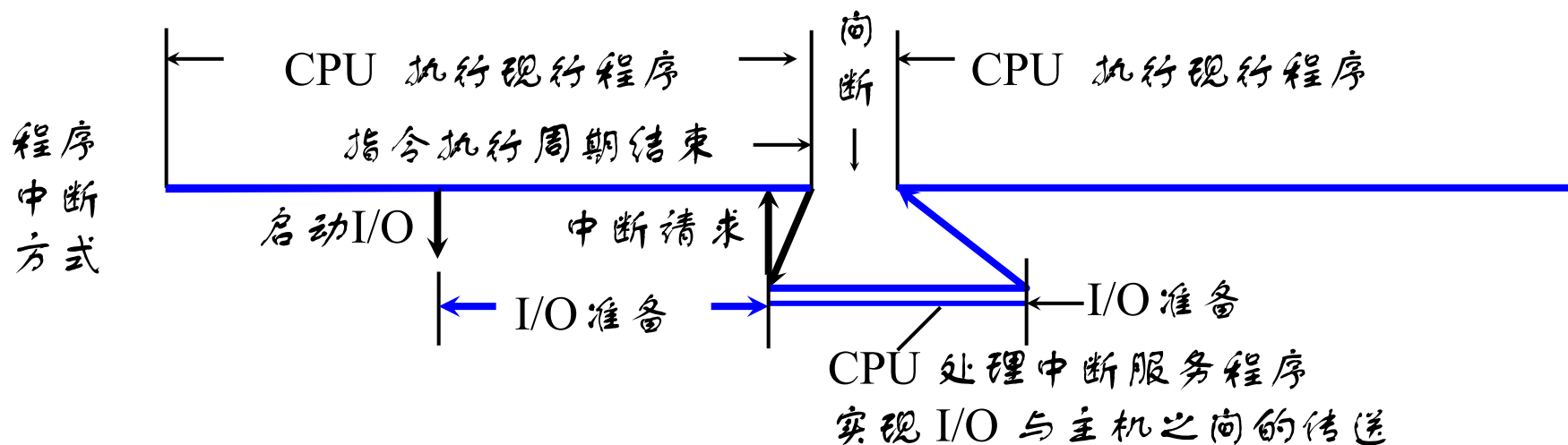
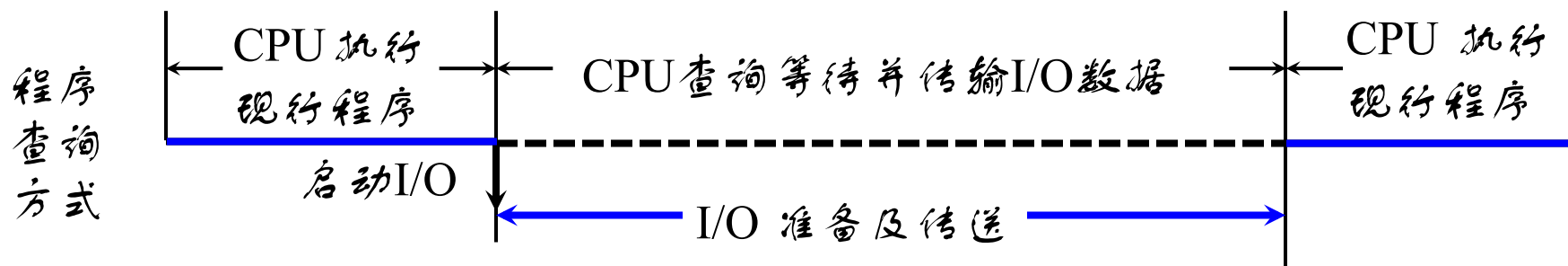
$T_0$	$PC \longrightarrow MAR$	$1 \longrightarrow R$	*	*
$T_1$	$Ad ( CMDR ) \longrightarrow CMAR$		*	
$T_2$	$M ( MAR ) \longrightarrow MDR$	$( PC ) + 1 \longrightarrow PC$	*	*
$T_3$	$Ad ( CMDR ) \longrightarrow CMAR$			
$T_4$	$MDR \longrightarrow IR$		*	
$T_5$	$OP ( IR ) \longrightarrow$ 微地址形成部件 $\longrightarrow CMAR$		*	

# I/O 系统

- 三种访问方式
  - 程序访问
  - 中断访问
  - DMA 方式



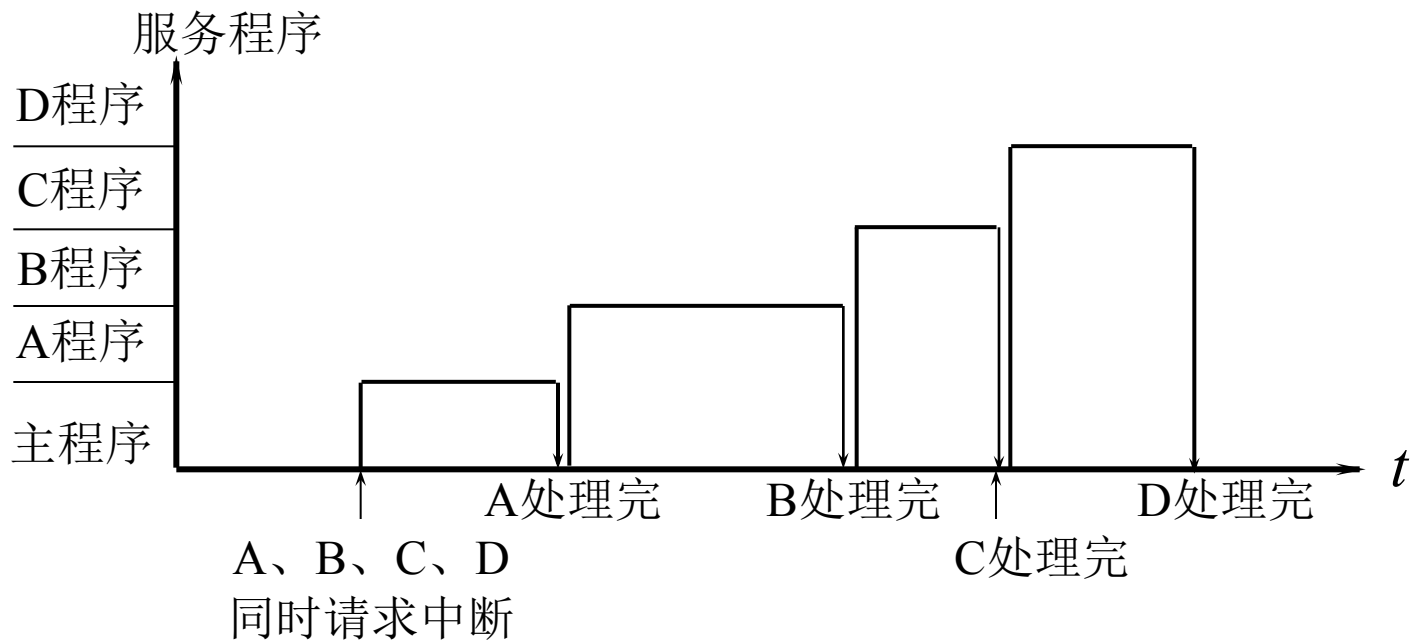
# 程序查询方式、程序中断方式



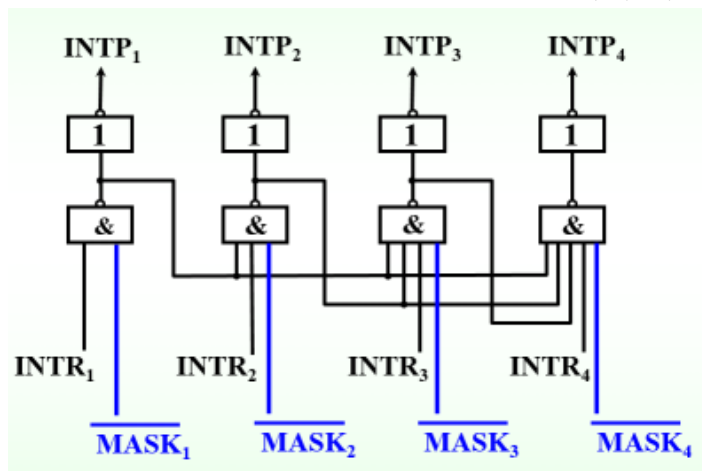
# 中断系统以及中断方式

- 掌握中断请求标记和中断判优逻辑；
- 掌握中断服务程序入口地址的寻找；
- 掌握中断响应过程；
- 掌握保护现场和恢复现场的方法；
- 掌握中断屏蔽技术。

# 屏蔽技术可改变处理优先等级

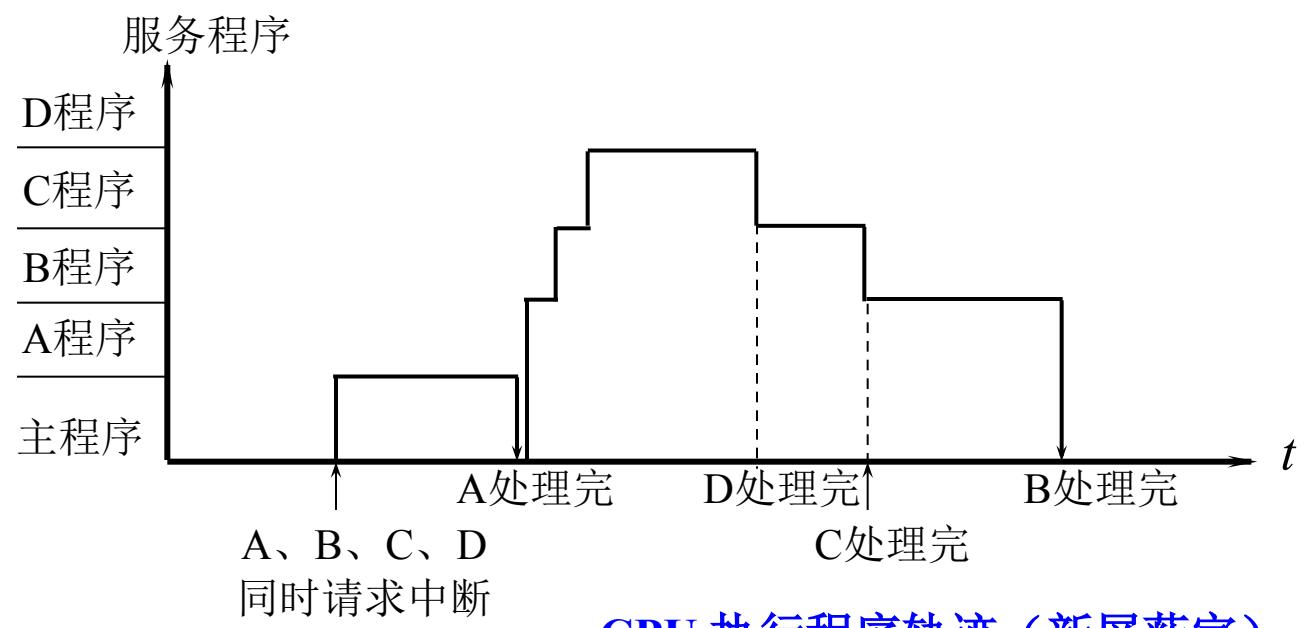


CPU 执行程序轨迹

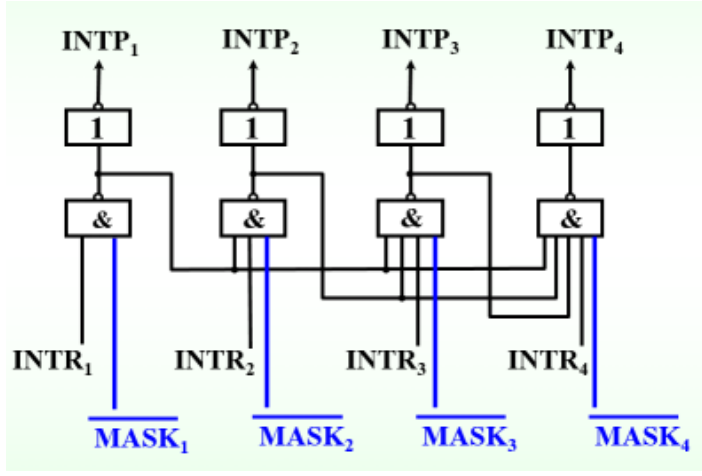


中断源	原屏蔽字
A	1 1 1 1
B	0 1 1 1
C	0 0 1 1
D	0 0 0 1

# 屏蔽技术可改变处理优先等级



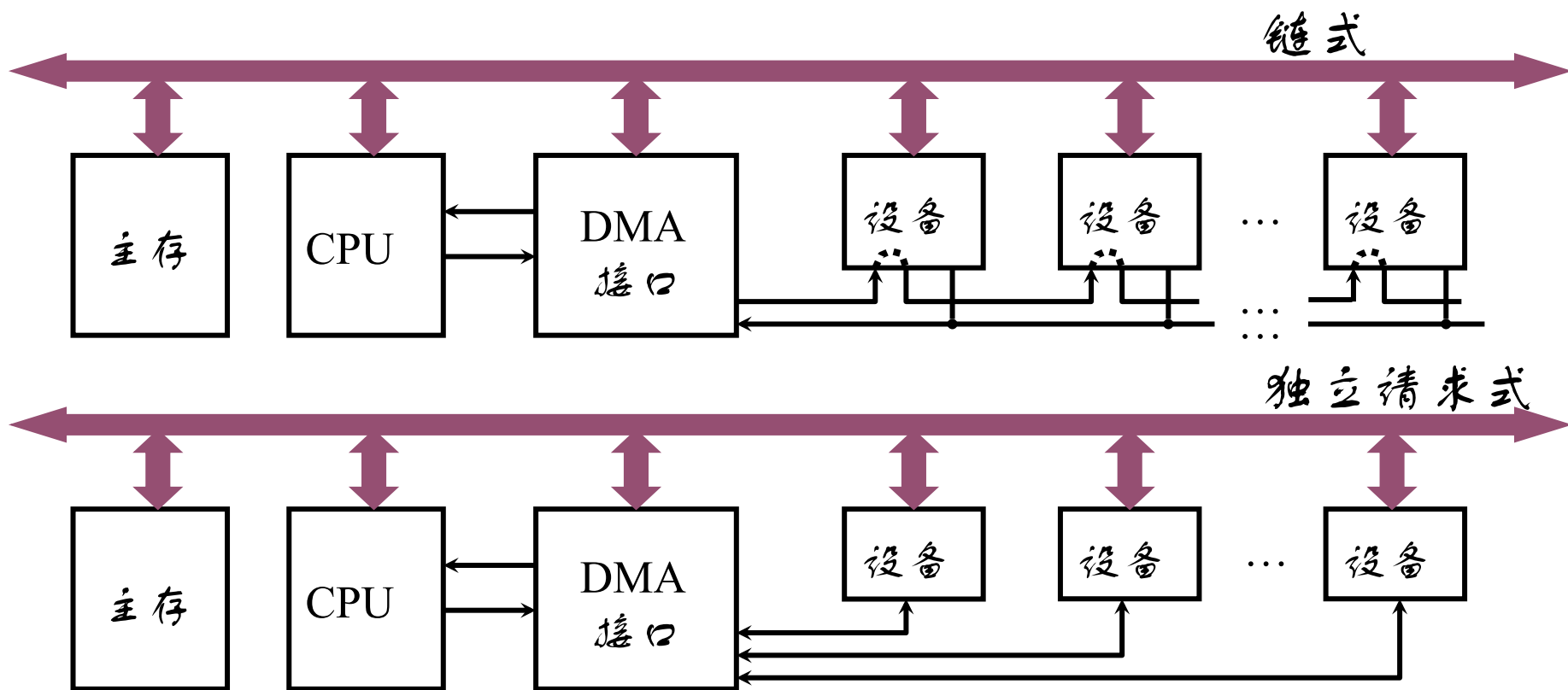
CPU 执行程序轨迹（新屏蔽字）



中断源	新屏蔽字
A	1 1 1 1
B	0 1 0 0
C	0 1 1 0
D	0 1 1 1

# DMA 方式

- DMA 接口 设备优先级 结合总线判优



# 总线

Ø总线的控制方式和总线的通信方式

链式查询方式，计数器定时查询方式，独立请求方式

Ø总线的通信方式：

同步： 由 **统一时标** 控制数据传送

异步： 采用 **应答方式**，没有公共时钟标准

半同步通信： **同步、异步结合**

串行通信： 异步串行数据**帧格式**、**波特率/比特率**

# 总线的性能指标

1. 总线宽度          数据线的根数
2. 总线带宽          每秒传输的最大字节数 (MBps)
3. 时钟同步/异步    同步、不同步
4. 总线复用          地址线 与 数据线 复用
5. 信号线数          地址线、数据线和控制线的 总和
6. 总线控制方式    并发、自动、仲裁、逻辑、计数
7. 其他指标          负载能力

## 例题 3.2 、 3.3 、 3.4

- 波特率概念 BaudRate

传输信号的速率，也称调制速率，以波形每秒的振荡数来衡量。波特率是单位时间内传送（所有）码元的数目，码元就是要用若个比特表示的最小单位。单位是bps

- 比特率

单位时间内传送**有效数据**比特的数目。

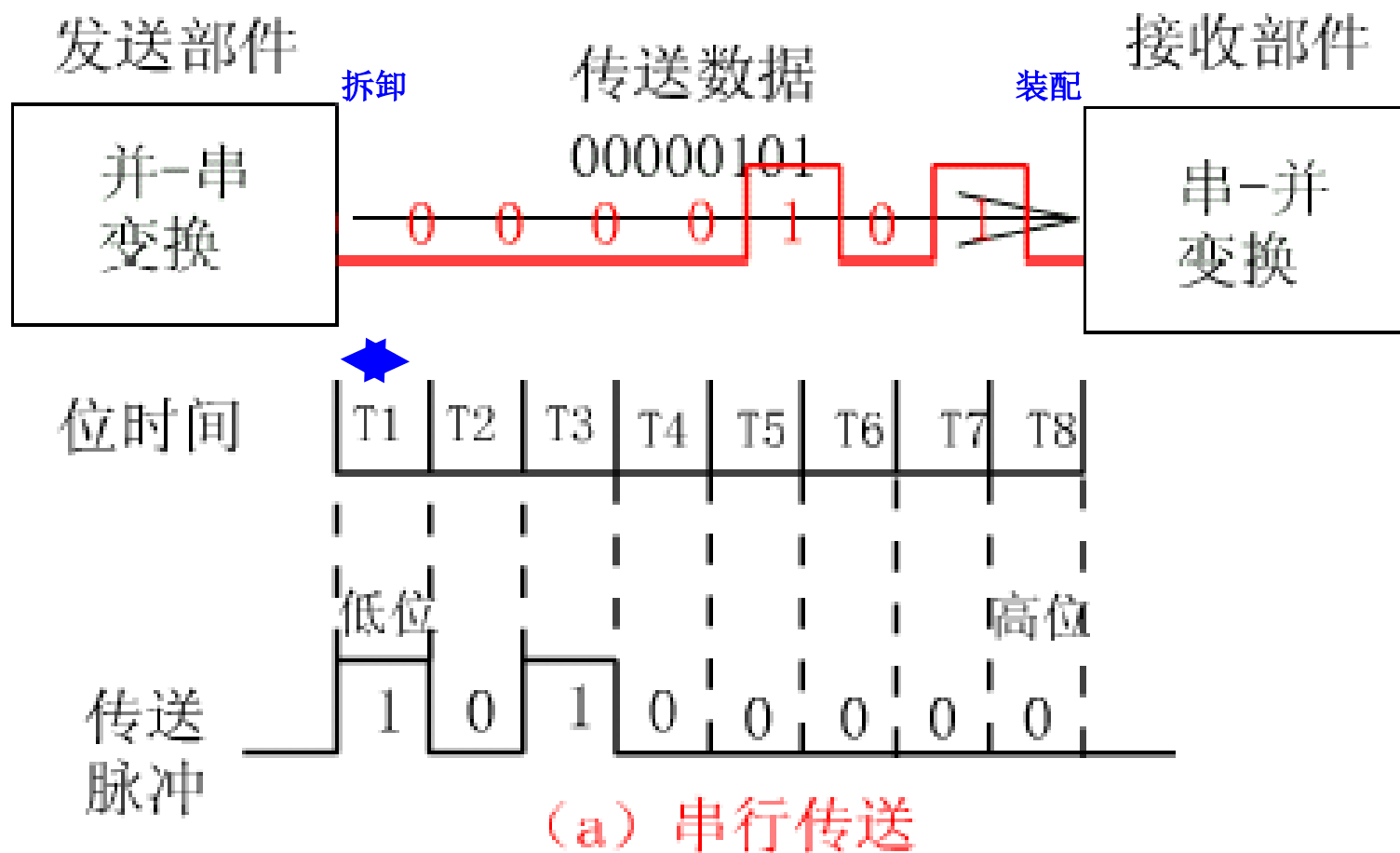
异步串行通信的数据传送速率：

- **波特率**：单位时间内传送二进制数据的位数——bps(位/秒)

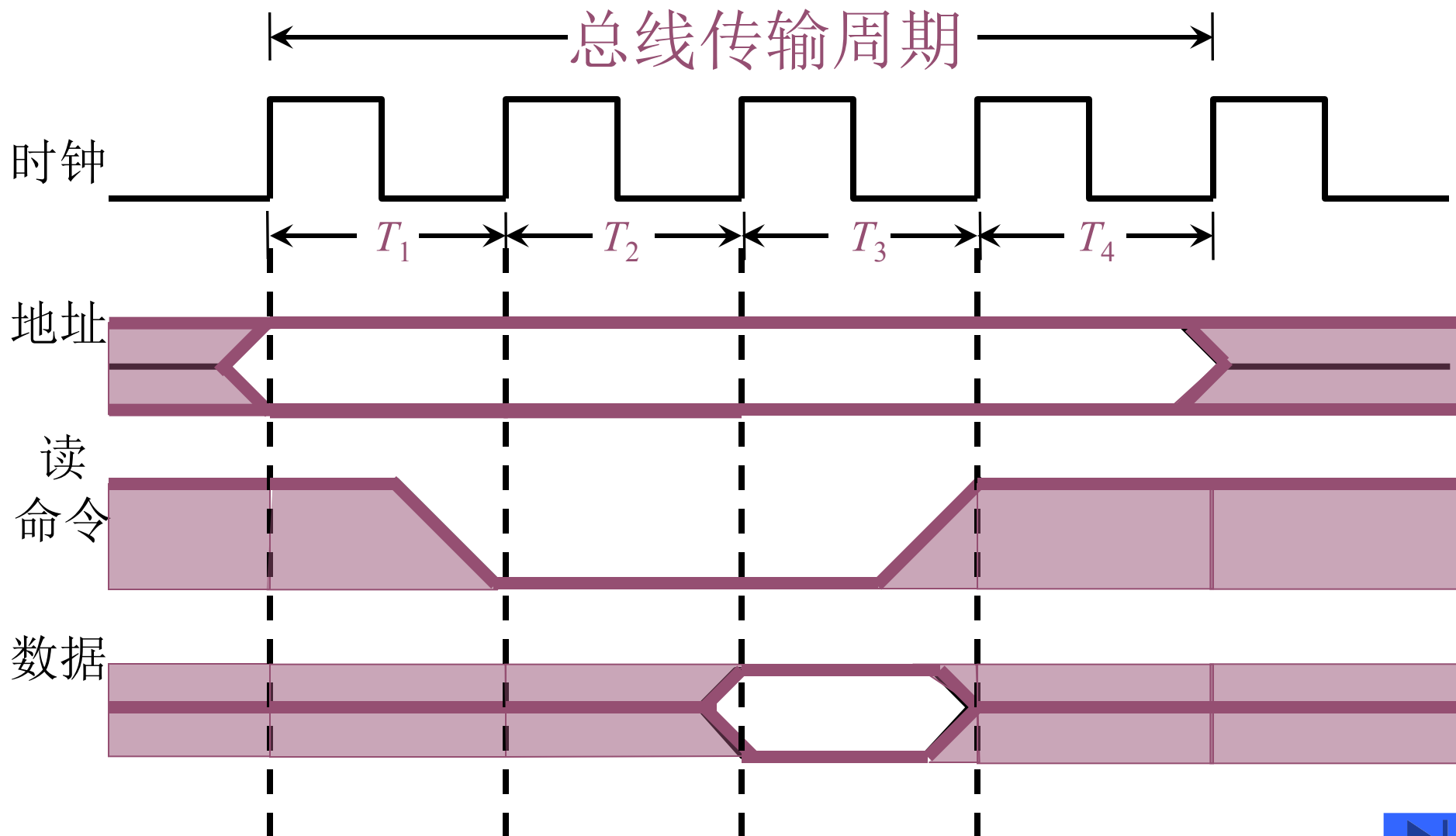
- **比特率**：单位时间内传送二进制**有效数据**的位数——bps(位/秒)



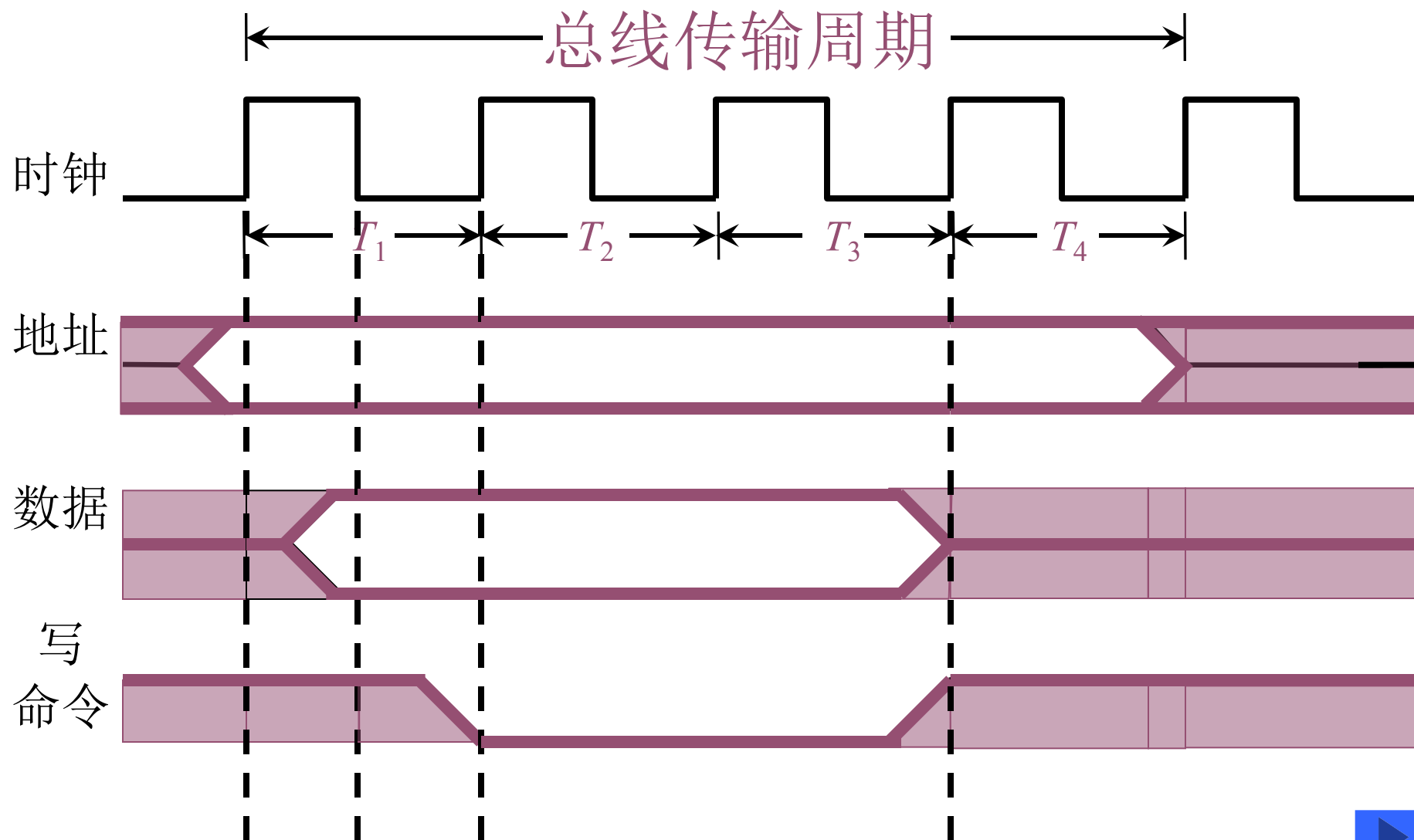
# 串行传送示意图



# (1) 同步式数据输入



## (2) 同步式数据输出



## 小贴士

- 读题 细心 仔细
- 四则运算 最后验算一遍
- 题干多读几遍 别丢信息 几个周期
- 画图 先用铅笔
- ABH 长的16进制的数 最好四位四位的写
- 书后例题要全做



**谢谢！  
祝大家考试顺利！**

**考的全会**



**考试锦鲤**



**认真学习你就是最棒的！**