



计算机组成与体系结构

复习课1

赵庆行

(qhzhao[at]xidian[dot]edu[dot]cn)

人工智能学院

第五章 指令系统

- 了解计算机的指令格式，掌握指令扩展操作码技术
- 掌握指令设计方法
- 掌握寻址方式
- 了解**CISC**与**RISC**的概念，了解理解精简指令系统计算机**RISC**的特点

基本概念

指令

指令系统

指令系统结构层/指令集体系结构

- 定义了什么/内容（P181）

存储模式：

- 大端、小端（如何存储、地址是多少）
- 堆栈（SP、栈顶栈底关系与入栈/出栈操作） P182
- 冯诺依曼与哈佛结构及其优缺点
- Load/store结构（RISC/CISC分别是哪种？）



基本概念

寄存器组织

- 典型寄存器作用AR/DR/IR/SP/PC（与第六章单总线结构CPU结合）

I/O模式

- 独立编制与统一编制的特点

RISC与CISC的特点（P202）

指令

构成：操作码、地址码；分别什么作用

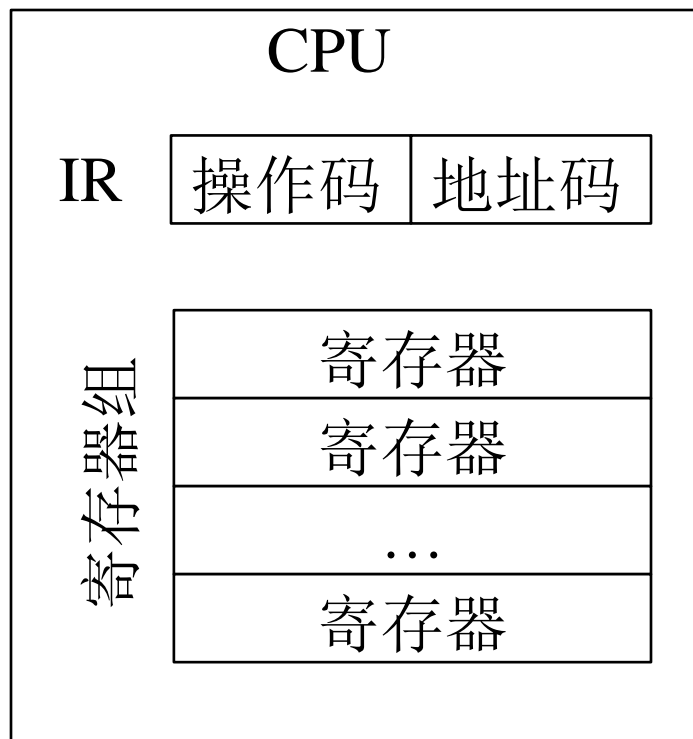
操作码：

- 固定长度
- 可变长度：Huffman编码（每条指令频度已知）
- 可变长度：扩展操作码（指令字长固定），从多地址计算，除零地址外预留1个。（每类指令频度已知，或者根据地址字段确定）

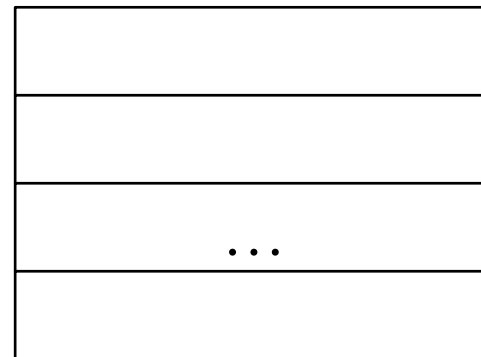
寻址方式

寻址定义，包括操作数和程序
寻址方式

- 理解数据在哪，如何找到数据



存储器

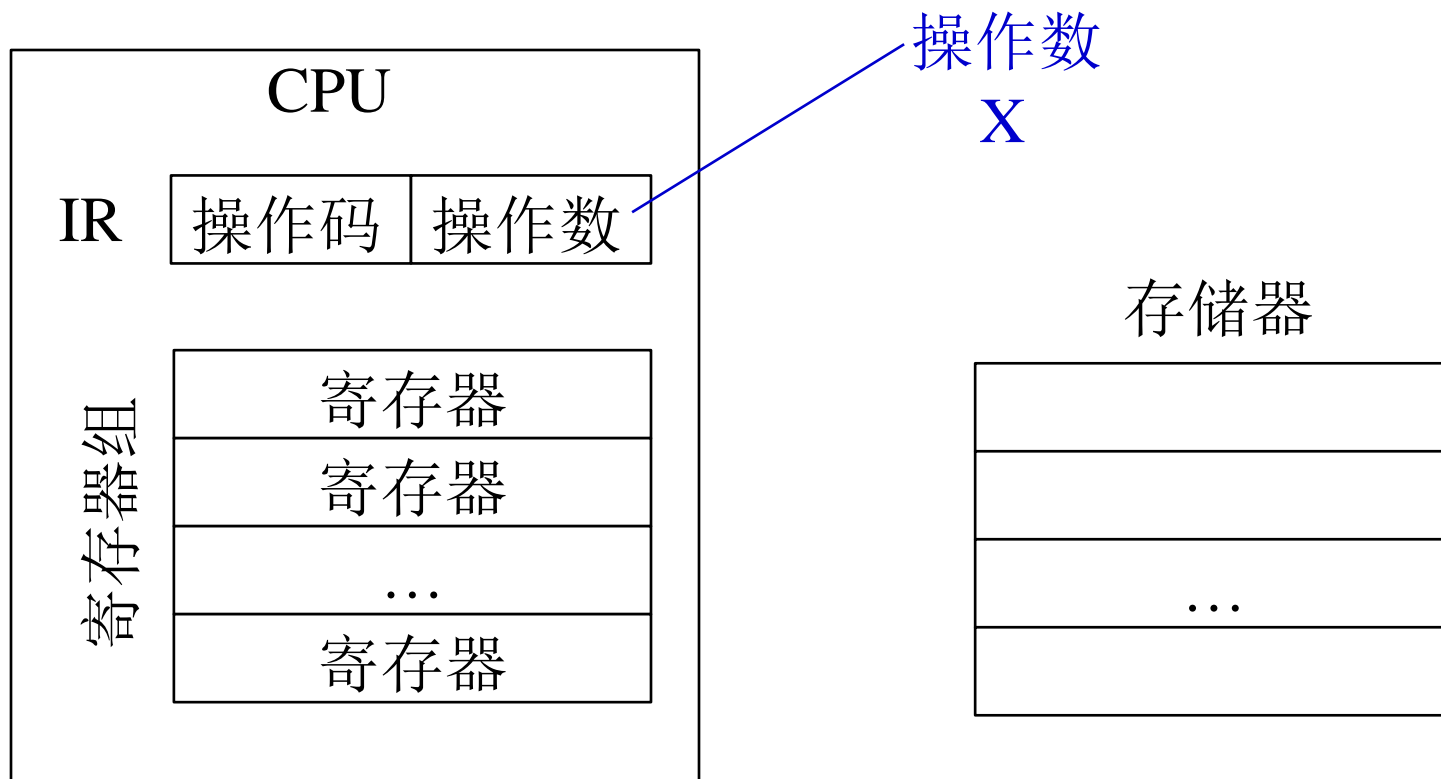


寻址方式

寻址方式

- 理解数据在哪，如何找到数据

立即寻址



寻址方式

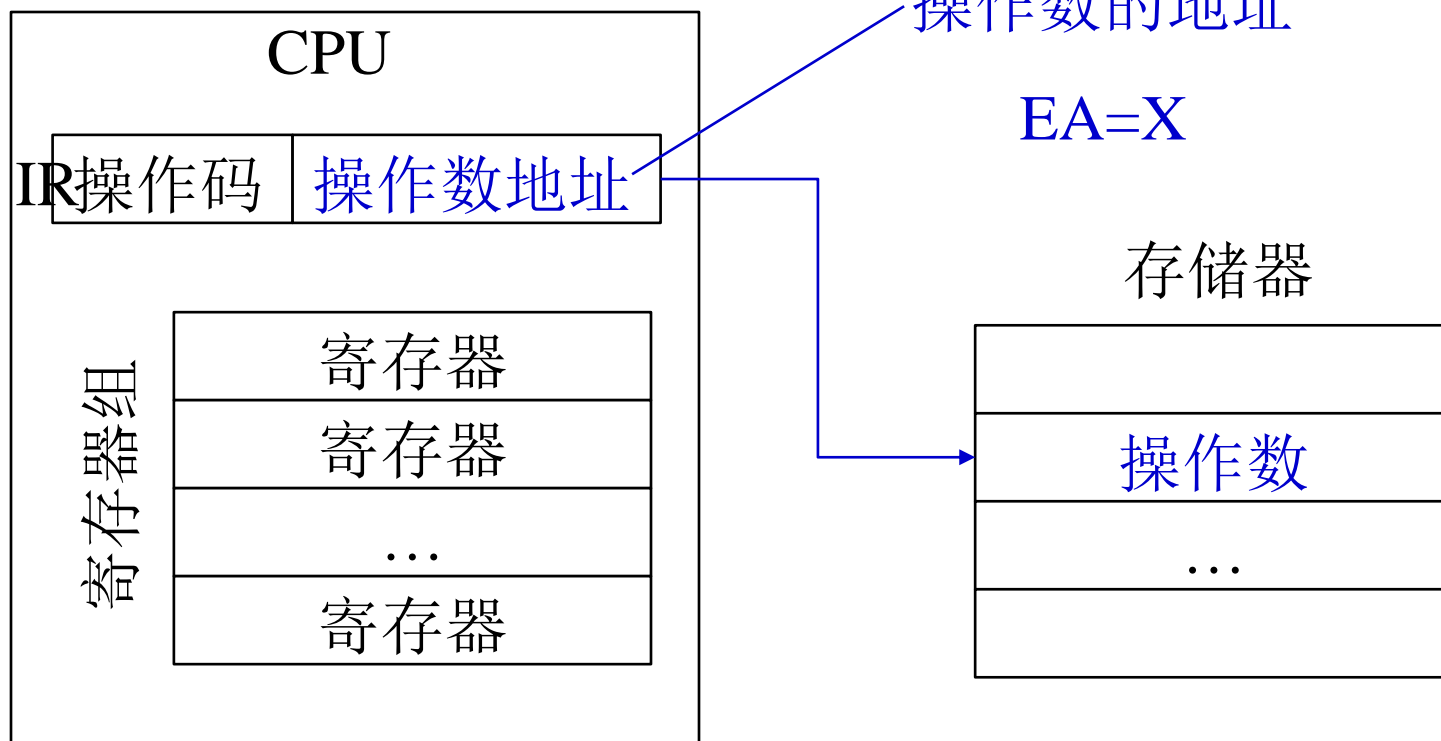
寻址方式

- 理解数据在哪，如何找到数据

直接寻址

操作数的地址

$EA=X$

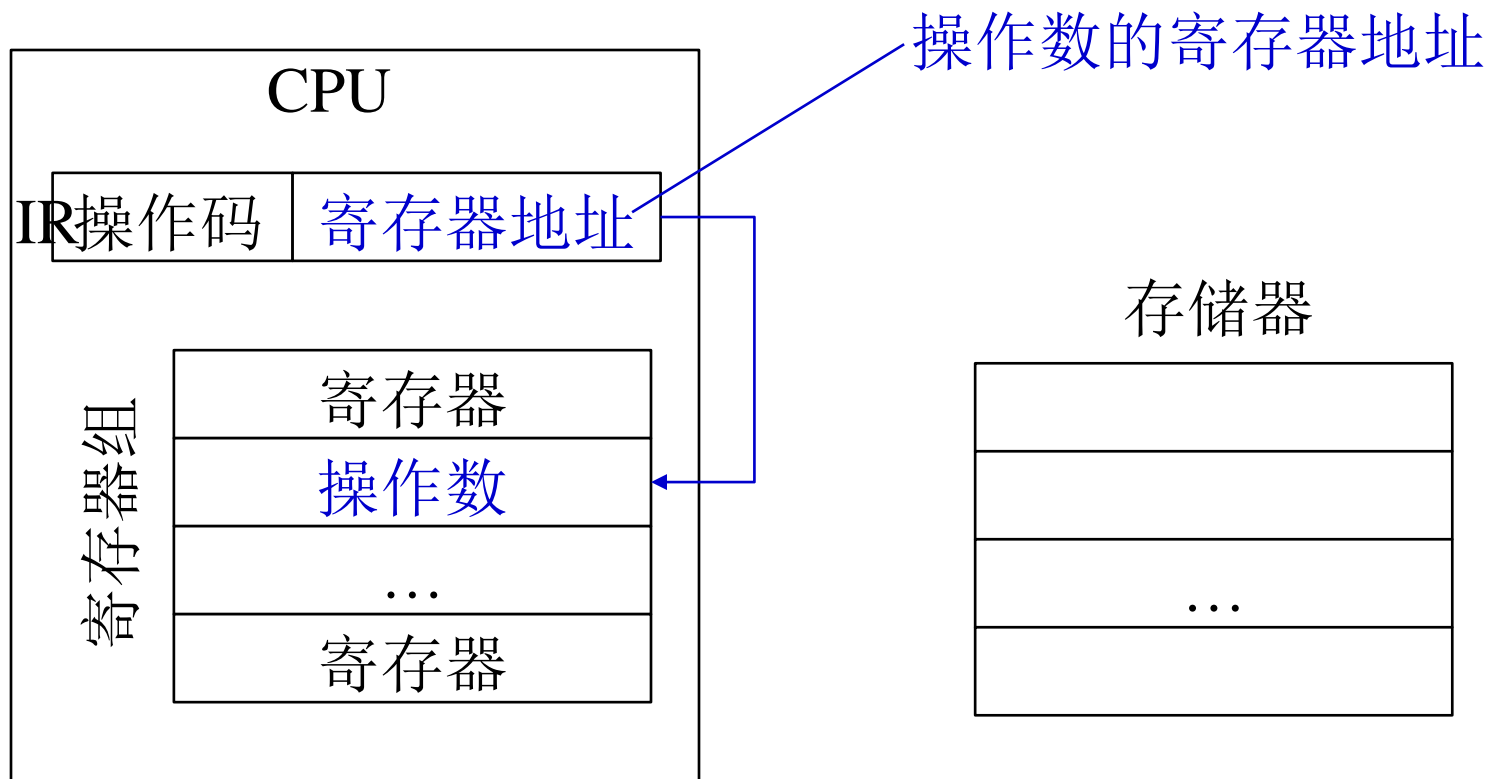


寻址方式

寻址方式

- 理解数据在哪，如何找到数据

寄存器寻址



寻址方式

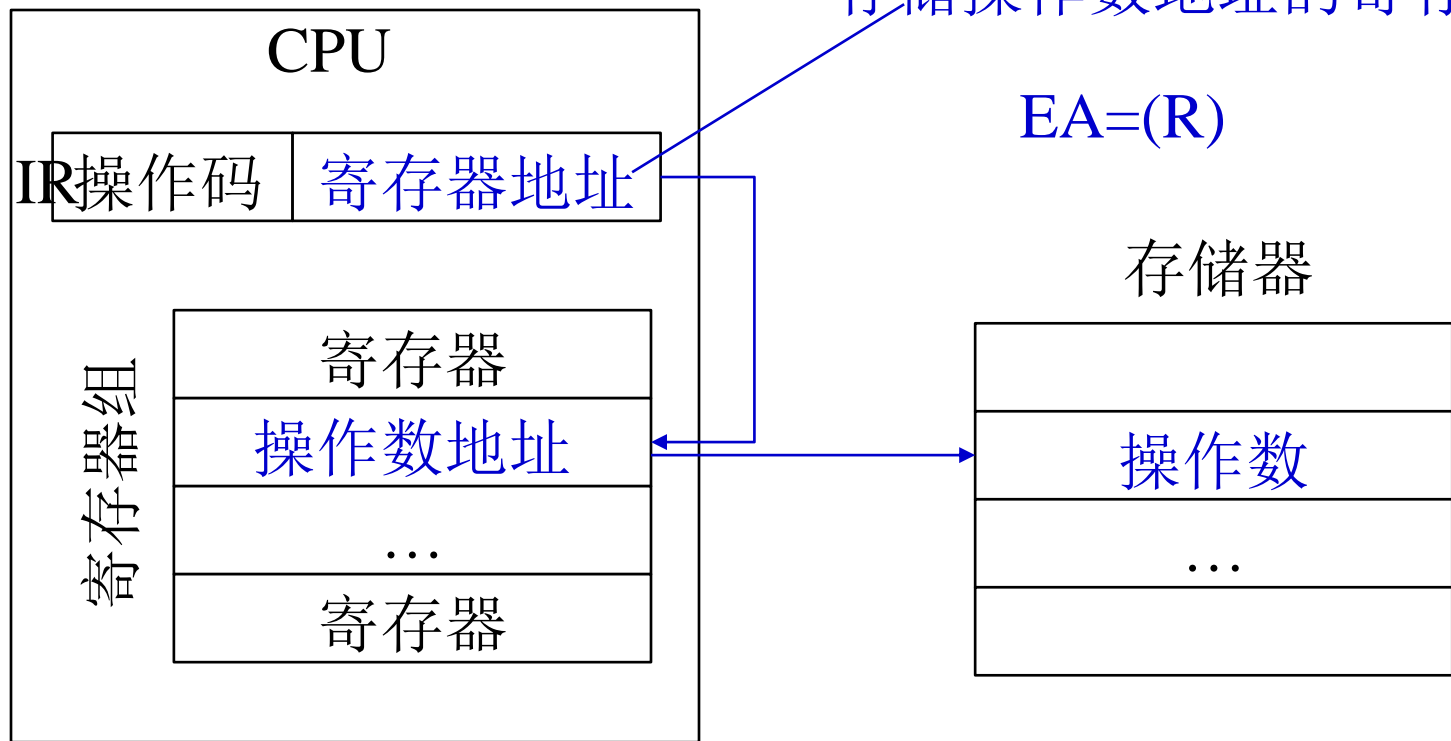
寻址方式

- 理解数据在哪，如何找到数据

寄存器间接寻址

存储操作数地址的寄存器地址

$EA=(R)$



寻址方式

寻址方式

- 理解数据在哪，如何找到数据

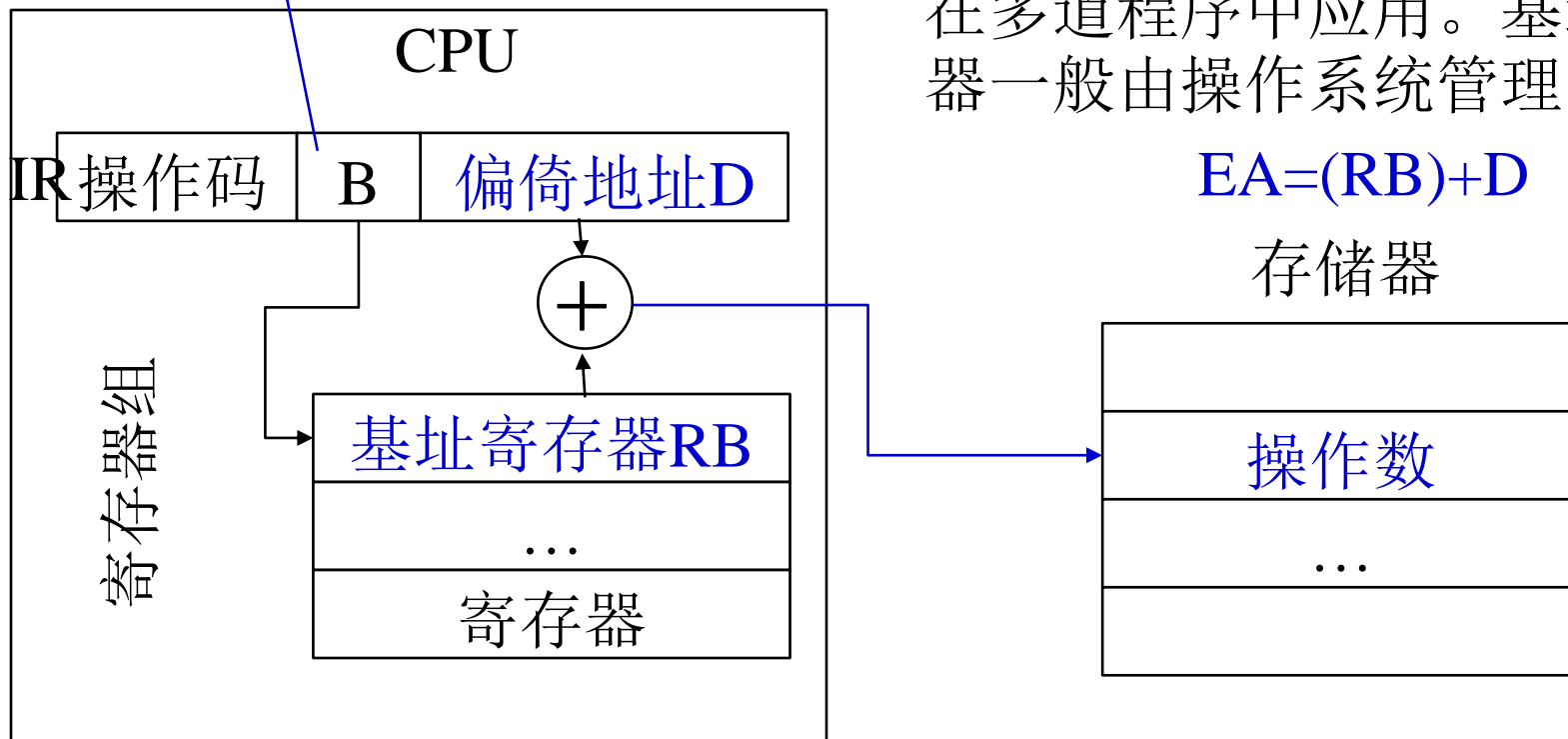
基址寻址

在多道程序中应用。基址寄存器一般由操作系统管理。

$$EA = (RB) + D$$

存储器

基址寄存器寻址位或地址



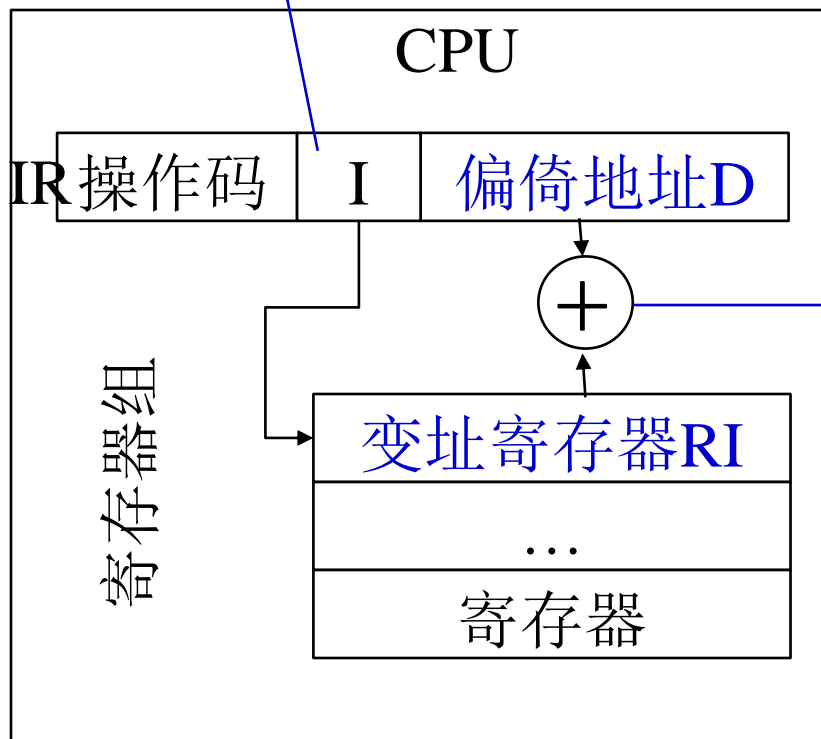
寻址方式

寻址方式

- 理解数据在哪，如何找到数据

变址寻址

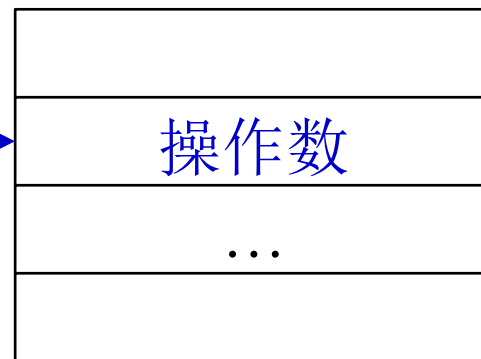
变址寄存器寻址位或地址



RI自动改变，因此偏移地址无需变化就可以处理数组或字符串。

$$EA = (RI) + D$$

存储器



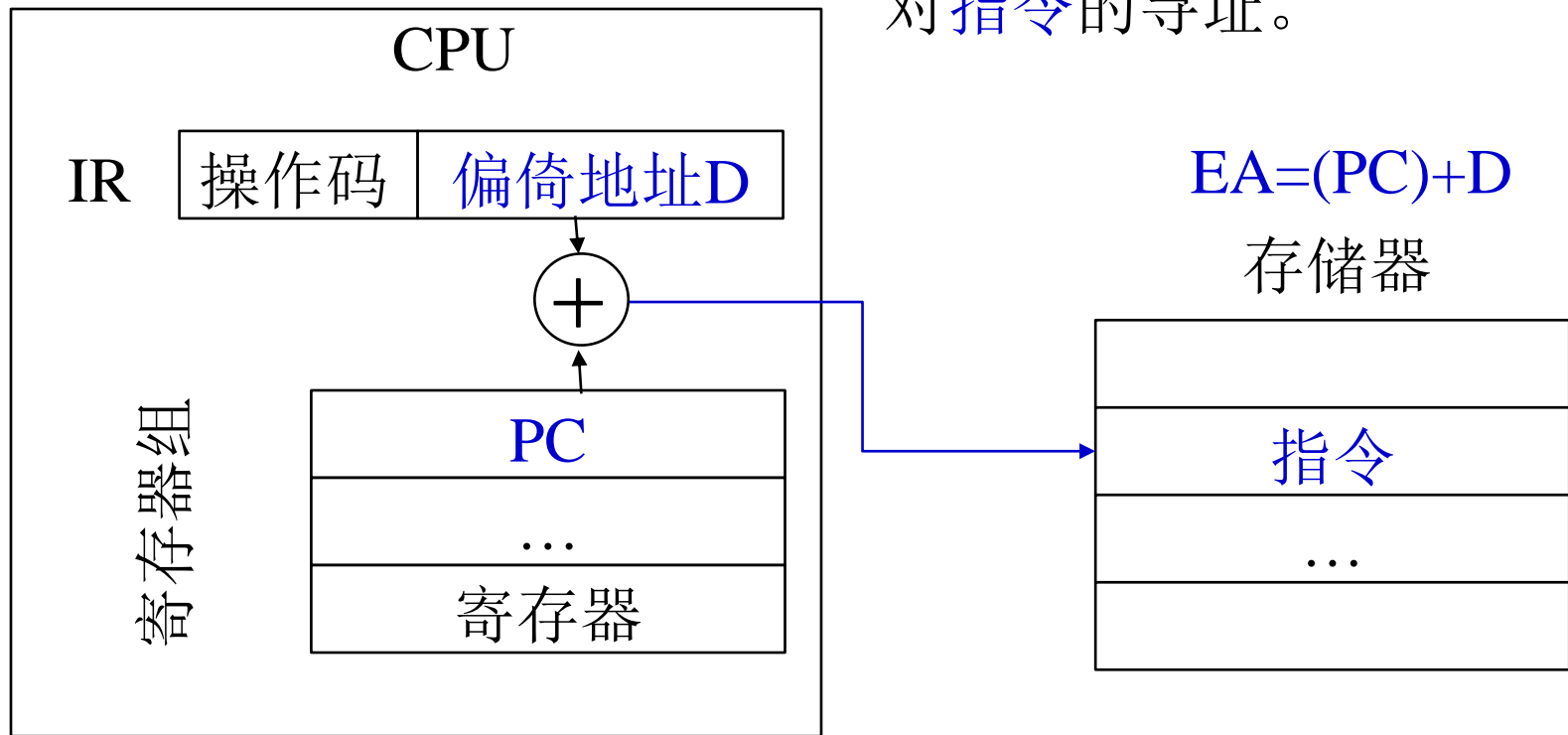
寻址方式

寻址方式

- 理解数据在哪，如何找到数据

相对寻址

对指令的寻址。

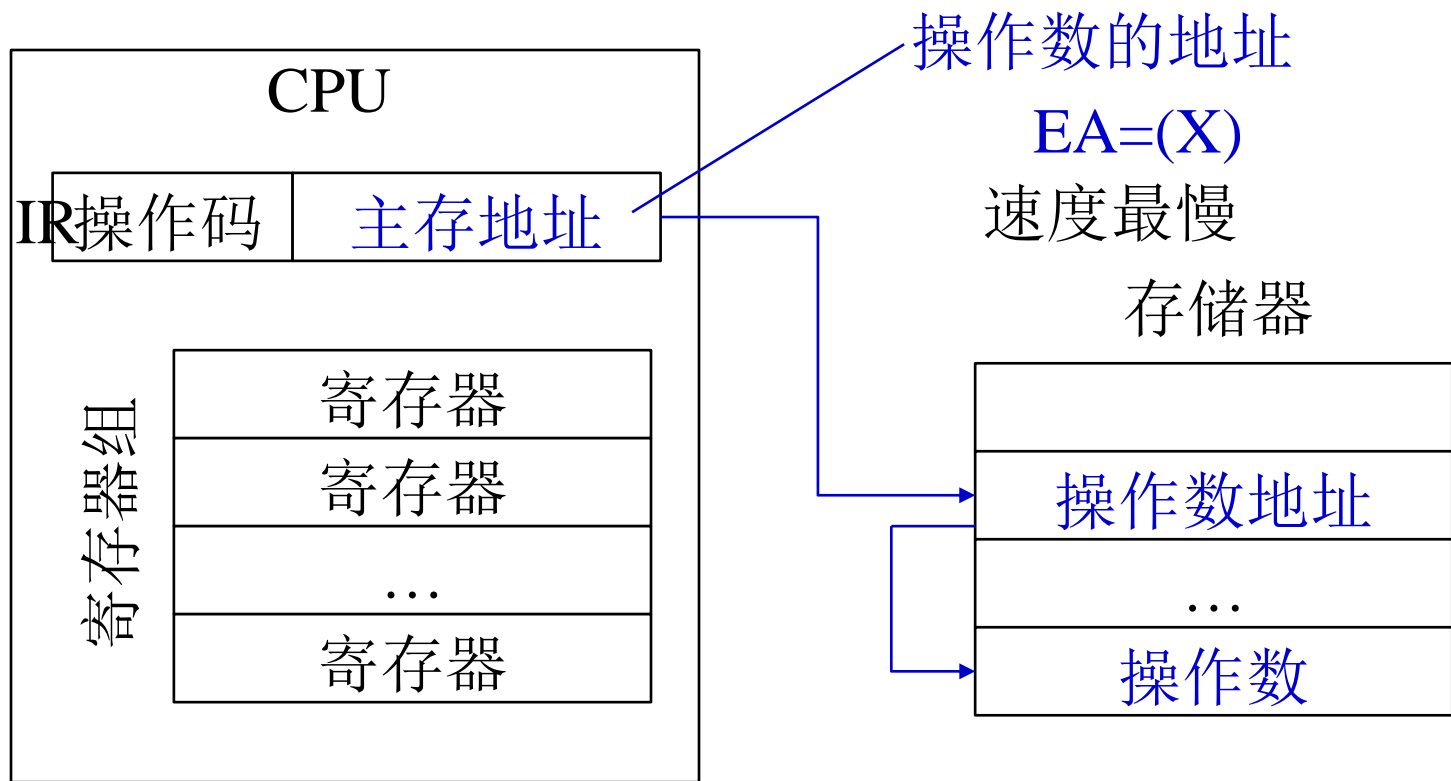


寻址方式


寻址方式

- 理解数据在哪，如何找到数据

存储器间接寻址



寻址方式	数据(指令)存储位置	是否访问寄存器	是否访问存储器	是否需要计算地址	速度
立即寻址	指令的地址字段	否	否	否	1
直接寻址	存储器	否	是	否	3
寄存器寻址	寄存器	是	否	否	2
寄存器间接寻址	存储器	是	是	否	4
基址寻址/ 变址寻址/ 相对寻址	存储器	是	是	是	5
存储器间接寻址	存储器	否	是, 2次	否	6



5.12 某指令系统指令字长16位，有零地址、一地址、二地址、三地址指令格式，每个地址字段均为4位，且三地址指令有L条，二地址指令有M条，零地址有N条。若采用定长操作码设计方法，一地址指令最多可以有多少条？若采用扩展操作码设计方法，一地址指令最多可以有多少条？

答：16-3*4=4位，操作码有4位可用，因此可以编码的指令条数：2⁴=16，因此，若采用定长操作码设计方法，一地址指令最多可以有（16-L-M-N）条。

若采用扩展操作码设计方法，一地址指令最多可以有：

$$\{[(2^4-L)*2^4-M]*2^{4-x}\}*2^4=N, \quad x=[(2^4-L)*2^4-M]*2^4-N/2^4$$

5.12 某指令系统指令字长16位，有零地址、一地址、二地址、三地址指令格式，每个地址字段均为4位，且三地址指令有L条，二地址指令有M条，零地址有N条。若采用定长操作码设计方法，一地址指令最多可以有多少条？若采用扩展操作码设计方法，一地址指令最多可以有多少条？

指令格式	操作码	地址1 (4位)	地址2 (4位)	地址3 (4位)
三地址指令	操作码 (4)			
二地址指令	操作码 (8)			
一地址指令	操作码 (12)			
零地址指令	操作码 (16)			

若采用扩展操作码设计方法，一地址指令最多可以有：

$$\{[(2^4-L)*2^4-M]*2^4-x\}*2^4=N, \quad x=[(2^4-L)*2^4-M]*2^4-N/2^4$$



5.15 根据操作数所在位置，指出寻址方式：

- (1) 操作数在寄存器中，为（**寄存器**）寻址方式。
- (2) 操作数地址在寄存器中，为（**寄存器间接**）寻址方式。
- (3) 操作数在指令中，为（**立即**）寻址方式。
- (4) 操作数地址在指令中，为（**直接**）寻址方式。
- (5) 操作数的地址为某一寄存器内容与位移量之和，可以是（**相对/基址/变址**）寻址方式。

5.16 某单地址指令格式如下图所示

OP	I	X	D
----	---	---	---

上图中，I为间接特征，X为寻址模式，D为形式地址。I、X、D组成该指令的操作数有效地址EA。设R为变址寄存器，R1为基址寄存器，PC为程序计数器，具体寻址方式特征如表5.11所示，请在表中填入寻址方式名称。

I	X	EA	寻址方式
0	00	$EA = D$	直接寻址
0	01	$EA = (PC) + D$	相对寻址
0	10	$EA = (R) + D$	变址寻址
0	11	$EA = (R1) + D$	基址寻址
1	00	$EA = (D)$	存储器间接寻址
1	11	$EA = ((R1) + D)$	基址间接寻址

5.17 某计算机字长为32位，采用单字长单地址指令格式，共有40条指令。若采用直接、立即、变址、相对四种寻址方式获取操作数，试设计指令格式。该指令格式可直接寻址的地址范围是多大？可间接寻址地址范围又是多大？

答： $2^6=64>40$,40条指令需要6位进行编码，寻址方式4种需要2位进行编码，因此地址码为24位。

31	26	25	24	23	0
Opcode		X		D	


X=00 直接寻址，直接寻址的地址范围是 $2^{24}=16M$;

X=01 立即寻址;

X=10 变址寻址,有效地址 $E = (IR) + D$ （可寻址4G个存储单元）;

X=11 相对寻址,有效地址 $E = (PC) + D$ （可寻址4G个存储单元）.

试题库指令系统2的20题



10、设机器字长为16位，存储器按字节编址，设PC当前值为1000H，当读取一条双字长指令后，PC值为（ ）

- A、 1001H
- B、 1002H
- C、 1004H

11、设机器字长为16 位，存储器按字节编址，CPU 读取一条单字长指令后，PC值自动加（ ）

- A、 1
- B、 2
- C、 4



22、某计算机按字节编址,采用大端方式存储信息。其中,某指令的一个操作数的机器数为ABCD 00FFH,该操作数采用基址寻址方式,指令中形式地址(用补码表示)为FF00H,当前基址寄存器的内容为C000 0000H,则该操作数的 LSB(即FFH)存放的地址是 ()

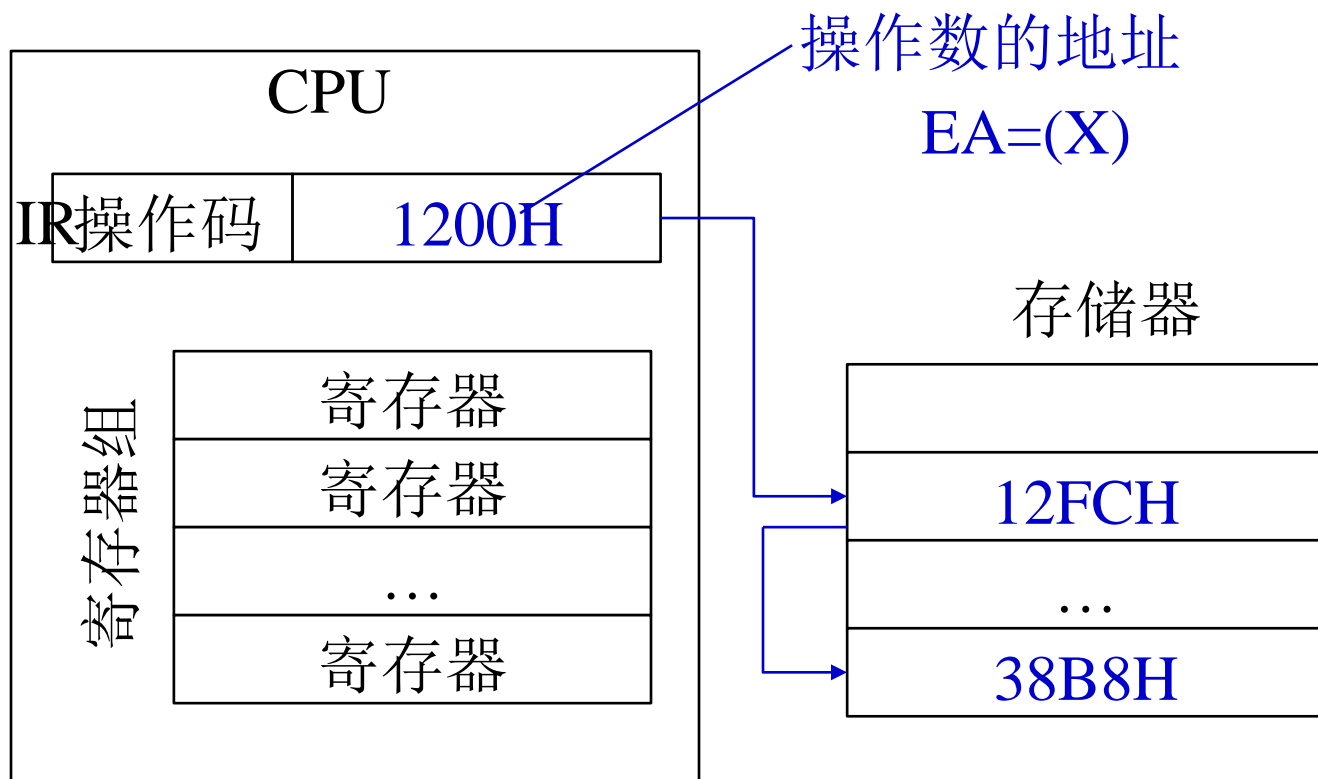
- A、 C000 FF00H
- B、 C000 FF03H
- C、 BFFF FF00H
- D、 BFFF FF03H

解析:基址寻址方式下,操作数的有效地址为基址寄存器内容加上形式地址,因此,操作数ABCD 00FFH存放在
 $C000\ 0000H + FF00H = C000\ 0000 + FFFF\ FF00 = BFFF\ FF00H$
因为是大端方式并按字节编址,所以LSB(FFH)存放的地址为BFFF FF03H。答案为选项D。

知识点:大小端存储、有效地址计算

24、假设某条指令的一个操作数采用一次间接寻址方式，指令中给出的地址码为1200H，地址1200H中的内容为12FCH，地址12FCH中的内容为38B8H，地址38B8H中的内容为88F9H，则该操作数的有效地址为()。

- A、 1200H
- B、 12FCH
- C、 38B8H
- D、 88F9H



第六章 中央处理器

- 了解中央处理器的微体系结构、**CPU性能参数（CPU时间、CPI、MIPS、FLOPS）**
- 掌握指令流程、指令执行的全过程的分析与设计（微操作流程分析）
- 掌握微程序控制器的基本工作原理，理解微程序设计技术
- 了解组合逻辑控制器及其设计思想

基本概念

- CPU基本功能
- CPU执行指令过程→执行程序过程
- CPU基本组成及其作用
 - 控制器
 - 数据通路
- 指令周期：包含取指和执行
- 微操作与微命令：概念、表示，两者关系

AR ← PC ; PC_{out} , AR_{in}
微操作 微命令

- 指令周期、CPU周期、节拍周期

单总线CPU结构

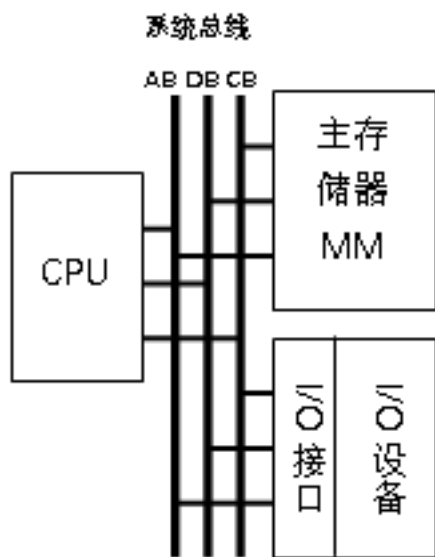


图 6.4 早期 x86 系统模型

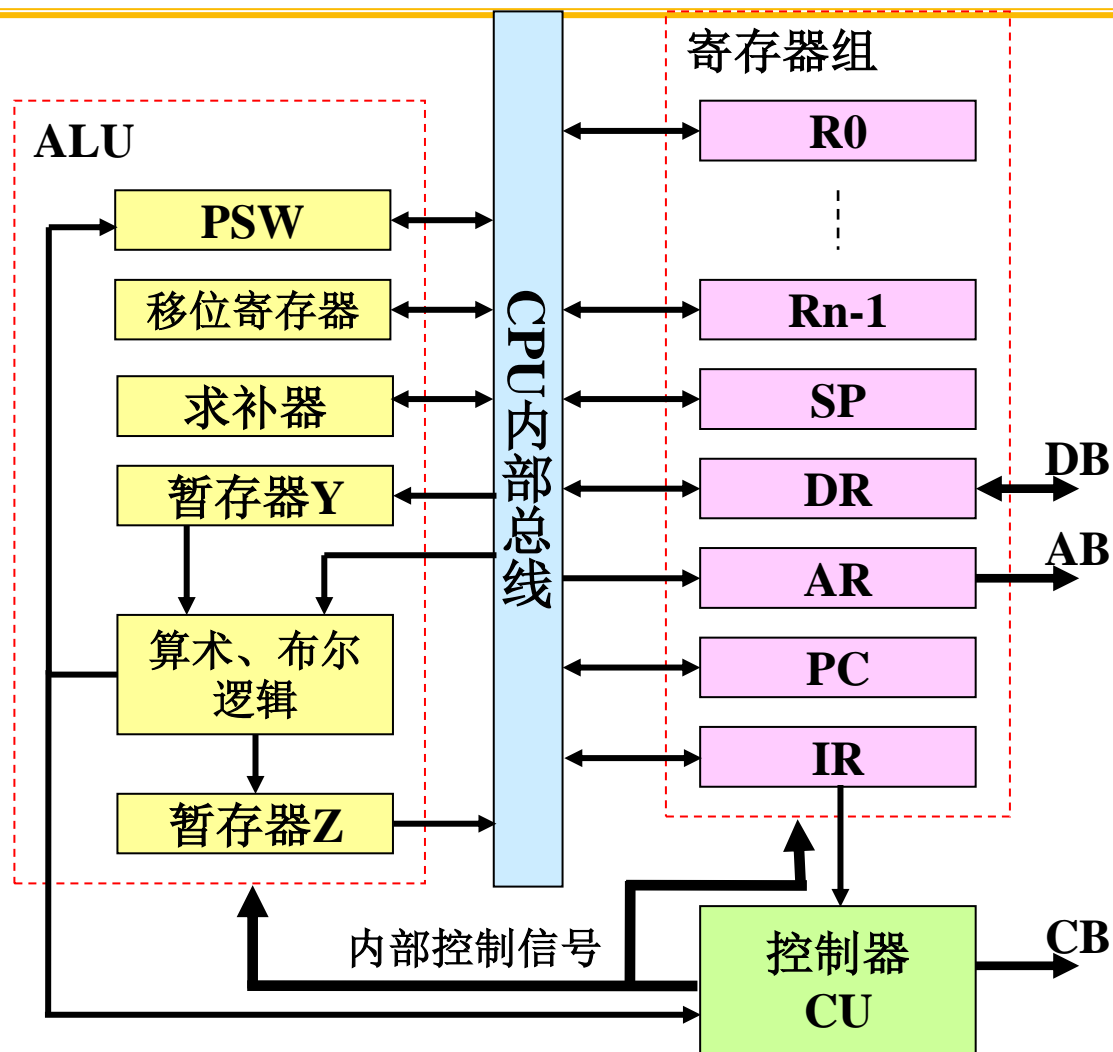


图 6.3 单总线数据通路 CPU 内部结构图

微命令

➤ 控制信号：

PC_{in} 为程序计数器的锁存输入控制信号；

PC_{out} 为程序计数器的输出允许控制信号；

PC+1 为程序计数器的自动增量（如自动加1）控制信号；

IR_{in} 为指令寄存器的锁存输入控制信号；

IR_{out} 为指令寄存器的输出允许控制信号；

SP_{in} 为指令寄存器的锁存输入控制信号；

SP_{out} 为指令寄存器的输出允许控制信号；

SP+1 为堆栈指示器的自动增量（如自动加n）控制信号；

SP-1 为堆栈指示器的自动减量（如自动减n）控制信号；

Ri_{in} 为通用寄存器Ri ($0 \leq i \leq n-1$) 的锁存输入控制信号；

Ri_{out} 为通用寄存器Ri ($0 \leq i \leq n-1$) 的输出允许控制信号；

➤ 控制信号：

Y_{in} 为暂存器Y的锁存输入控制信号；

Z_{out} 为暂存器Z的输出允许控制信号；

AR_{in} 为地址寄存器向CPU内部总线的锁存输入控制信号；

AR_{out} 为地址寄存器面向系统总线的输出允许控制信号；

DRI_{in} 为双端口数据寄存器面向CPU内部总线的锁存输入控制信号；

DRI_{out} 为双端口数据寄存器面向CPU内部总线的输出允许控制信号；

DRS_{in} 为双端口数据寄存器面向系统总线的锁存输入控制信号；

DRS_{out} 为双端口数据寄存器面向系统总线的输出允许控制信号；

Mread 为从主存储器读出信息的读控制信号；

Mwrite 为将信息写入到主存储器的写控制信号；

IOrread 为从I/O设备输入信息的读控制信号；

IOWrite 为将信息写入到I/O设备的写控制信号；

微命令

➤ 控制信号：

ADD为加载至ALU的加法运算控制信号；

SUB为加载至ALU的减法运算控制信号；

AND为加载至ALU的逻辑与运算控制信号；

OR为加载至ALU的逻辑或运算控制信号；

SHL为加载至ALU的逻辑左移控制信号；

SHR为加载至ALU的逻辑右移控制信号；

ROL为加载至ALU的循环左移控制信号；

ROR为加载至ALU的循环右移控制信号；

.....

微操作

➤ SUB R0, (X)

一级时序

T1: $AR \leftarrow PC$

;取指令阶段

T2: $DR \leftarrow \text{Memory}[AR], \text{Mread}$

T3: $PC \leftarrow PC + I$

$IR \leftarrow DR$

T4: $AR \leftarrow IR(\text{地址字段})$

;执行指令阶段

T5: $DR \leftarrow \text{Memory}[AR], \text{Mread}$

T6: $AR \leftarrow DR$

T7: $DR \leftarrow \text{Memory}[AR], \text{Mread}$

T8: $Y \leftarrow R0$

T9: $Z \leftarrow Y - DR$

T10: $R0 \leftarrow Z$

微操作

➤ SUB R0, (X)

两级级时序

M1:

;取指CPU周期

T1: $AR \leftarrow PC$

T2: $DR \leftarrow \text{Memory}[AR], \text{Mread}$

T3: $PC \leftarrow PC + I$

$IR \leftarrow DR$

M2:

;执行CPU周期

T1: $AR \leftarrow IR$ (地址字段)

T2: $DR \leftarrow \text{Memory}[AR], \text{Mread}$

T3: $AR \leftarrow DR$

T4: $DR \leftarrow \text{Memory}[AR], \text{Mread}$

T5: $Y \leftarrow R0$

T6: $Z \leftarrow Y - DR$

T7: $R0 \leftarrow Z$

微操作

➤ SUB R0, (X)

两级级时序

M1:

;取指CPU周期

T1: $AR \leftarrow PC$

T2: $DR \leftarrow \text{Memory}[AR], \text{Mread}$

T3: $PC \leftarrow PC + I$

$IR \leftarrow DR$

;取数CPU周期

M2:

T1: $AR \leftarrow IR(\text{地址字段})$

T2: $DR \leftarrow \text{Memory}[AR], \text{Mread}$

T3: $AR \leftarrow DR$

T4: $DR \leftarrow \text{Memory}[AR], \text{Mread}$

M3:

;执行CPU周期

T5: $Y \leftarrow R0$

T6: $Z \leftarrow Y - DR$

T7: $R0 \leftarrow Z$

对应的微命令?

微程序控制器

➤ 微指令与微程序：定义，关系

➤ 微指令格式

地址域	控制域
-----	-----

➤ 微程序控制器一般结构

— 控制存储器CM：与主存区别

➤ 微指令地址格式：与指令格式设计区别

— 双地址、单地址、可变格式：与一般格式区别

— 不同地址格式特点：指令长度、控存大小、速度

— 各自对应的控制器结构

— 控存容量与微指令各字段的关系计算

➤ 微程序控制器与硬布线控制器比较

微程序控制器

- 水平型微指令与垂直型微指令
- 水平型微指令控制域编码
 - 直接表示法
 - 译码法
 - 字段译码法：互斥的信号放在同一字段，相容的信号放在不同字段，每个字段中要设计一个无效控制信号的编码

CPU性能指标

- CPU时间
- CPI
- IPC
- MIPS