

《微机系统与接口》课程总结

基本内容

1. 微机系统基础

（数制、系统、IA-16/32 μ P/MPU）

2. 指令和用途—系统硬件相关

3. 汇编语言程序设计（伪指令-数据-程序流）

4. 半导体存储器（概念、MPU接口电路）

5. 接口技术—数字量接口

6. 模拟量接口及应用编程

7. IA-32原理概念

《微机系统与接口》考试

1. 选择题（基本概念）——**20%**
2. 填空题（程序阅读理解、简述）
3道中题—— **30%**
3. 接口电路编程
1道中题—— **15%**（模拟IO）
4. 接口电路设计与应用（连线、计算、编程）
1道大题多个小问—— **35%**（数字IO，含画图）

《微机系统与接口》课程总结

基本内容

1. 微机系统基础

(数制、系统、IA-16/32 μ P/MPU)

2. 指令和用途—系统硬件相关

3. 汇编语言程序设计 (伪指令-数据-程序流)

4. 半导体存储器 (概念、MPU接口电路)

5. 接口技术—数字量接口

6. 模拟量接口及应用编程

7. IA-32原理概念

第一章、微机系统基础

1. 数制、补码的概念与运算

二进制数据

1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0

$$D_n * 2^n + D_{n-1} * 2^{n-1} + \dots + D_1 * 2^1 + D_0 * 2^0$$

(MSB/LSB (Most/Least Significant Bit) D_n / D_0)

字长： 计算机运算器 (ALU)位数 (二进制)

本课程: $n=8/16/32$ 位 也决定通用寄存器的位数

数据定义伪指令-人工输入形式-物理存储(二进制数)

DB -20H (100H-20H=E0H) , -1(0FFH), -20(ECH), 81H(-127)

MYDATA DW 200, -200, 0FFF0H, 1, -200H, 200H

MAX=? Min=? 根据实际物理存储内容判别

(1) 数制、补码的概念与运算

□ 带符号数的表示，第一位：符号位（0=正数；1=负数）
补码——计算机擅长做加法！ $A-B=A+(-B)$

在微机系统中，数值以补码表示和存储

例：汇编指令 **MOV AX, -4** 操作数=？

-4 \longleftrightarrow 八位100H-4=0FCH, 16位1000H-4=0FFFCH

“码和反码是人为定义，补码是机器特性”

□ 原码、反码、补码在特定字长内（ $n=8/16/32$ ）表示范围，
相互转换——溢出：运算结果超出范围 CF/OF

8位有符号数整数表示列表

8位二进制数	十六进制数	无符号数	原码	补码	反码
00000000	00H	0	+0	+0	+0
00000001	01H	1	+1	+1	+1
00000010	02H	2	+2	+2	+2
.....
01111101	7DH	125	+125	+125	+125
01111110	7EH	126	+126	+126	+126
01111111	7FH	127	+127	+127	+127
10000000	80H	128	-0	-128	-127
10000001	81H	129	-1	-127	-126
10000010	82H	130	-2	-126	-125
.....
11111101	FDH	253	-125	-3	-2
11111110	FEH	254	-126	-2	-1
11111111	FFH	255	-127	-1	-0

两种二进制编码

□ 十进制数的二进制编码表示 压缩/非压缩BCD码

——输入输出符合习惯!

□ 非数值（字母与符号）的二进制编码表示

ASCII码: 00-7FH 字符(美国标准信息交换代码)

0DH—— CR 回车

0AH—— LF 换行

字符: ‘0’-‘9’——30-39H;

‘A’——41H, ‘a’——61H

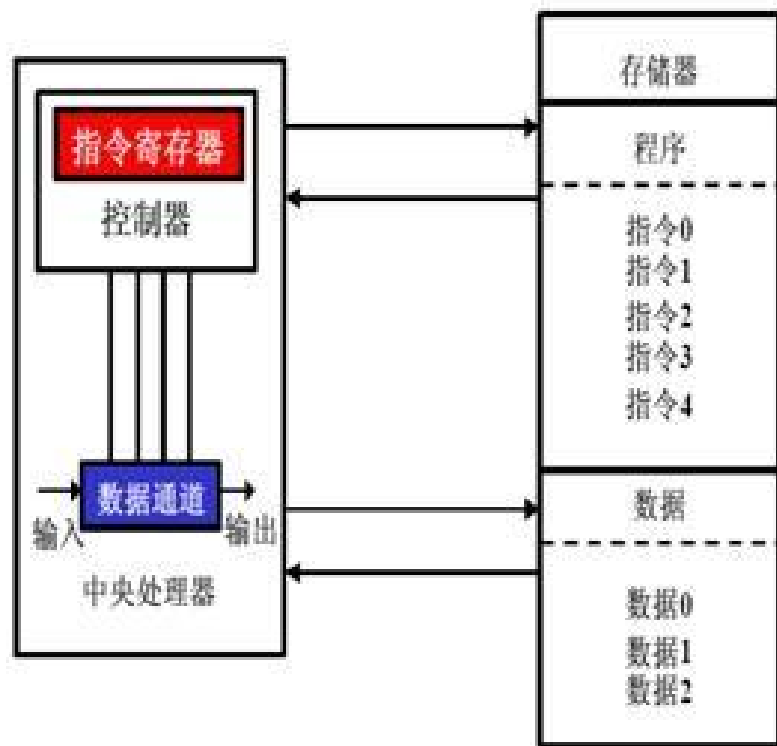
(2) 8086/8088 CPU硬件结构

CPU基本构成:

❑ **执行单元EU**: 执行指令, 含ALU和通用寄存器;

❑ **总线接口单元BIU**: 与存储器或I/O端口之间进行数据传送, 并能形成物理地址, 含段寄存器和指令指针寄存器;

——并行: 预取-指令队列-执行, 流水线



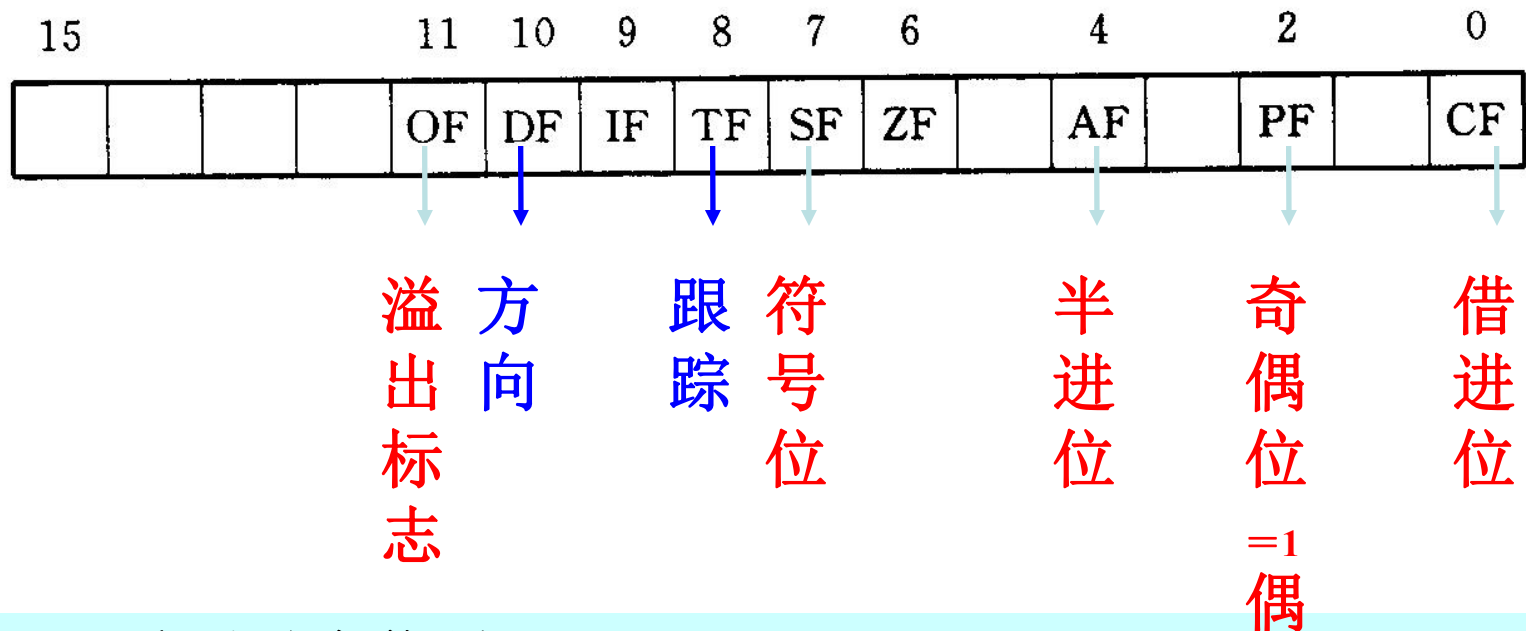
冯·诺依曼结构:

是一种将**程序指令存储器**和**数据存储器**合并在一起的存储器结构。程序指令存储地址和数据存储地址指向同一个存储器的不同物理位置。

——Intel 8086/8088

(2) 8086/8088 CPU硬件结构

❑ **FLAGS各位定义** IF, TF, (CF, DF, ZF, OF/SF, AF)
CLI/STI; CLD/STD; 单步运行 (中断)



❑ **CPU重要引脚信号**: DEN, ALE, /WR, /RD, /IOR/W, AEN (信号有效性)

❑ **最小/最大模式** 单/多处理器(标准总线)

(3) 8086/8088存储器组织*

1.字长和寻址空间1M(00000~FFFFFFH)

8086: AD0~AD15, A16~A19

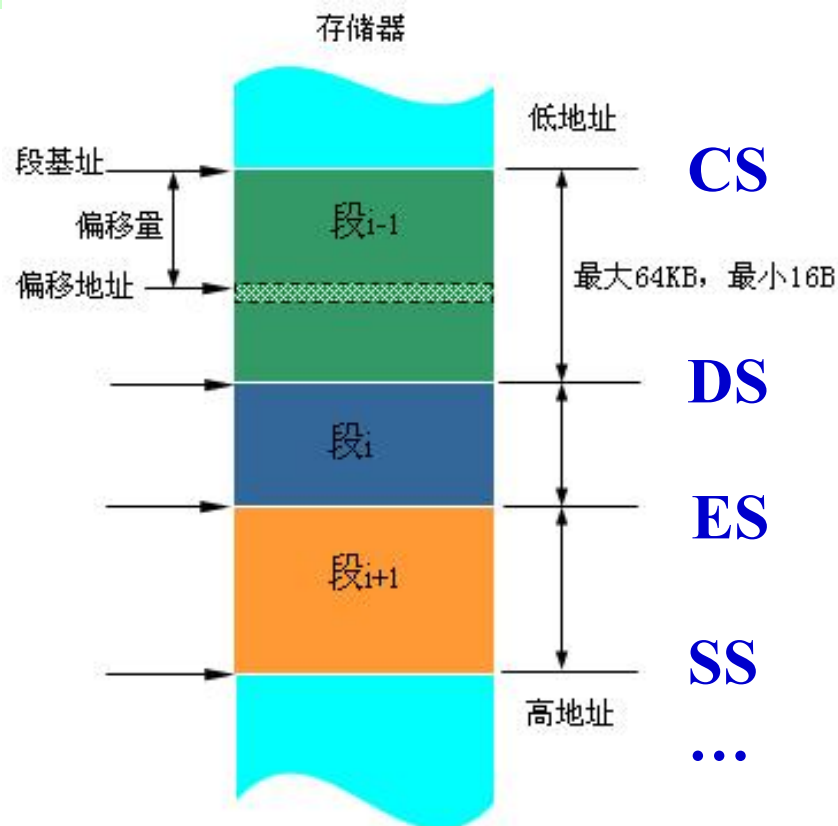
8088: AD0~AD7, A8~A19

2.分段结构

16位寄存器→20位物理地址

将1MB的地址空间分为段，
(4个活动的存储段)一个段最多64K字节的连续空间；

用段基址和段内偏移地址来访问物理存储器。



- 80286/386实地址模式类似

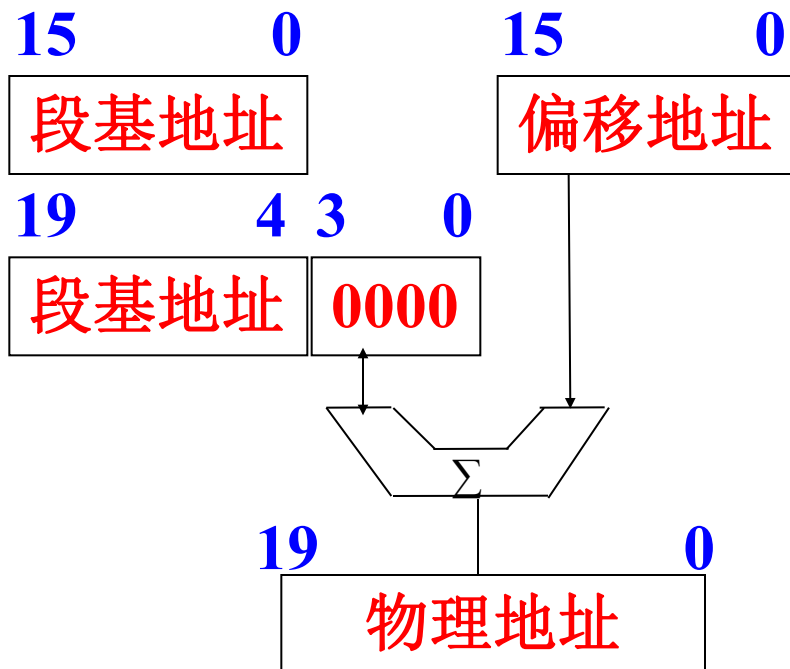
(3) 8086/8088存储器组织*

□ **逻辑地址**=段基址S (16位) 和偏移地址EA (16位)

→ 程序使用

□ **物理地址**= $S*16+EA$ (得到20位物理地址 $A_{19}--A_0$)

→ 呈现在数据总线上



■ **段基地址默认段寄存器**

■ 代码段的段基→CS

■ 堆栈段的段基→SS

■ 数据段的段基→DS

■ 附加段的段基→ES

■ **偏移地址约定寄存器**

→ 寻址方式

(3) 8086/8088存储器组织*

寻址方式——不光是MOV，要理解操作本质

无须访问总线：

立即寻址 **MOV AX, 100**

寄存器 **MOV AX, BX**

通过总线访问存储器：

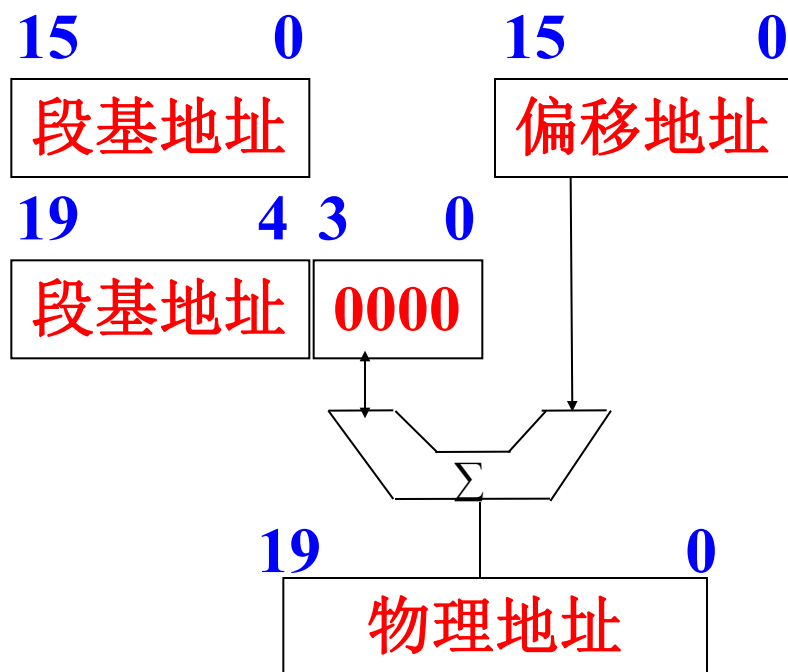
直接 **MOV AX, [200H]**

寄存器间址 **MOV AX, [BX]**
 MOV AX, [BP]

基址 **MOV AX, [BP+TAB]**
 MOV AX, TAB[BX]

变址 **MOV AX, [SI+TAB]**
 MOV AX, TAB[SI]

基+变址
 ADD TAB[BX+SI], AX



(3) 8086/8088存储器组织*

存储单元/堆栈/中断矢量表——会画图!

❑ 存储器：高字节高地址，低字节低地址。

❑ 堆栈段：SS表示堆栈段的段基，SP指向栈顶

PUSH AX ;(SP) \leftarrow (SP-2);(SP+1) \leftarrow AH;(SP) \leftarrow AL

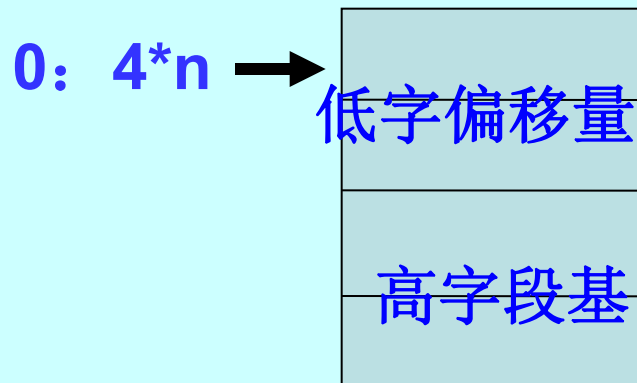
POP AX;AL \leftarrow (SP); AH \leftarrow (SP+1); (SP) \leftarrow (SP+2)

POP并不改变堆栈里单元内容；堆栈以字为单位操作，**SP**值总是偶数，先进后出（**FILO**）。

❑ 中断矢量表：中断类型号 $n=0\sim 255$ ，256个中断源*4字节=1K (00000-003FFH)，连续存放；

一个中断矢量

——类型为 n 的中断对应的
中断服务程序入口地址。



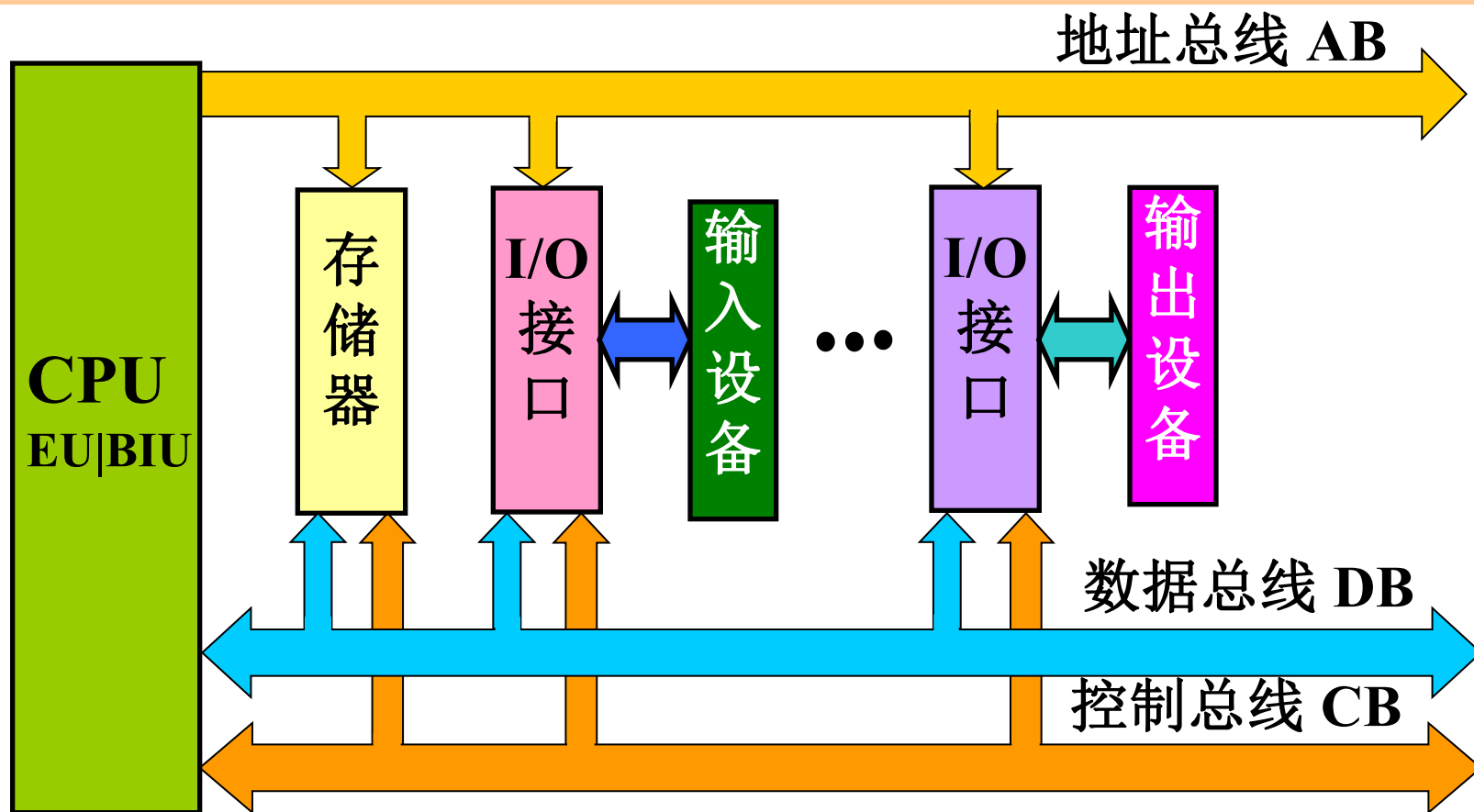
(3) 8086/8088存储器组织*

12年考题

2. 8086/8088CPU 地址总线为 20 位，用 CPU 指令进行存储器寻址时一般通过_____实现。

- A. 2 个 16 位物理地址
- B. 1 个 4 位物理地址和 1 个 16 位物理地址
- C. 2 个 16 位逻辑地址
- D. 1 个 4 位逻辑地址和 1 个 16 位逻辑地址

(4) 微机系统组成及各部分之间关系



□ 总线：地址,数据,控制——根据传送相关内容区分

□ 独立编址，不同指令区分8086访问存储器和I/O：

A19—A0 $2^{20}=1\text{M}$ 存储空间； A15-A0 $2^{16}=\text{I/O}$ 空间

100H以上需用DX间址：**MOV DX,2F8H** **IN AL,DX** **OUT DX,AL**

(5)8086/8088工作过程

□ CPU收到RESET信号后:

各寄存器的状态=0000H, CS=0FFFFH → 启动地址FFFF0H
(CS: IP=FFFFH: 0H)

□ 理解指令与硬件对应关系:

- 基本时钟T=系统时钟, 最小的单位;
- 机器周期: 把一条指令的执行过程划分为若干个阶段, 每一阶段完成一项基本操作: 取指、运算、存储器读写、I/O读写、中断响应、等待Tw、空闲周期;
- 指令周期: 指令从取出到执行完毕时间, 包括若干机器周期;

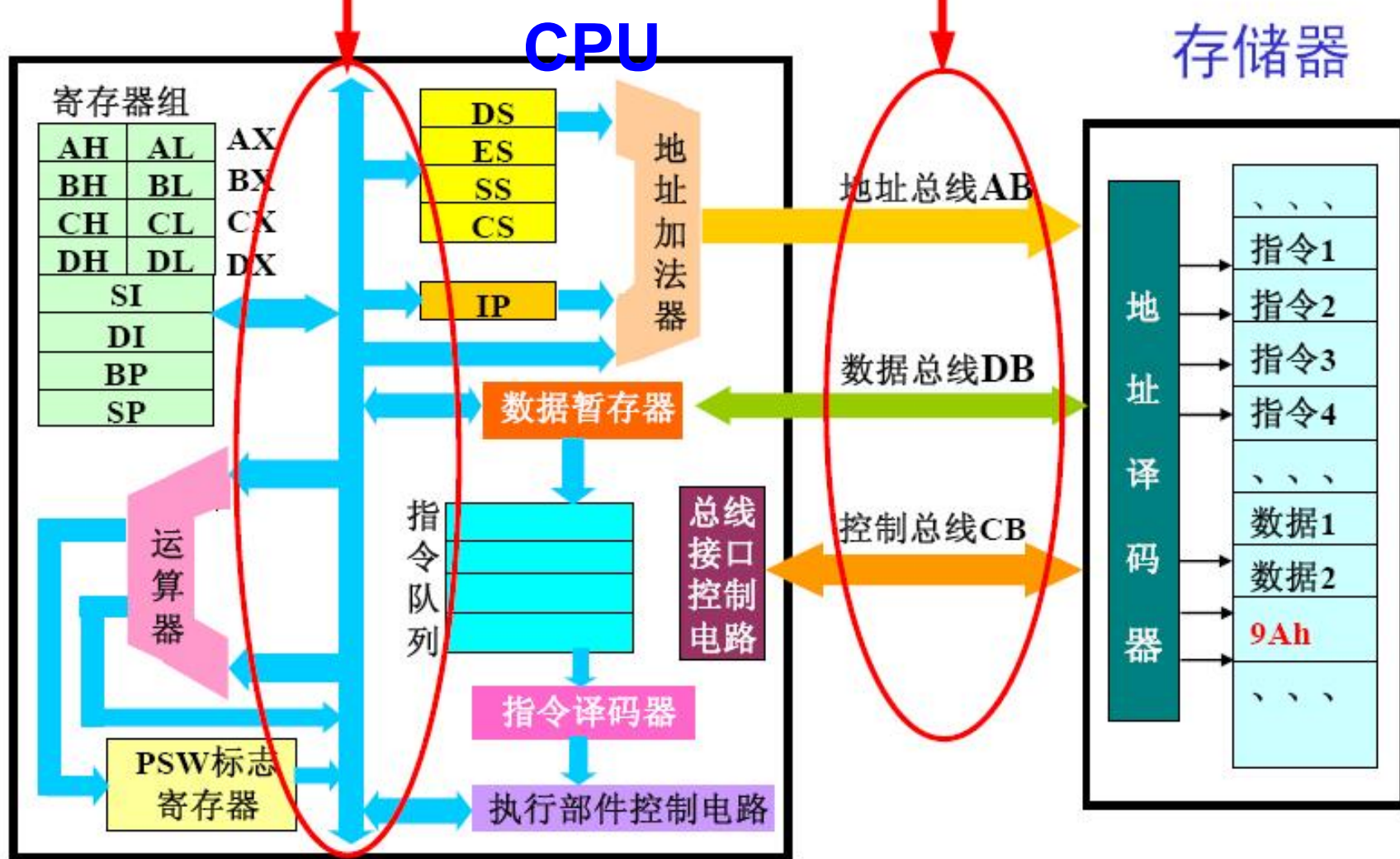
P238图5.3~图5.6

□ 含总线操作的机器周期——总线周期

BIU完成总线接口操作, 理解总线时序图→与指令相关

CPU

存储器

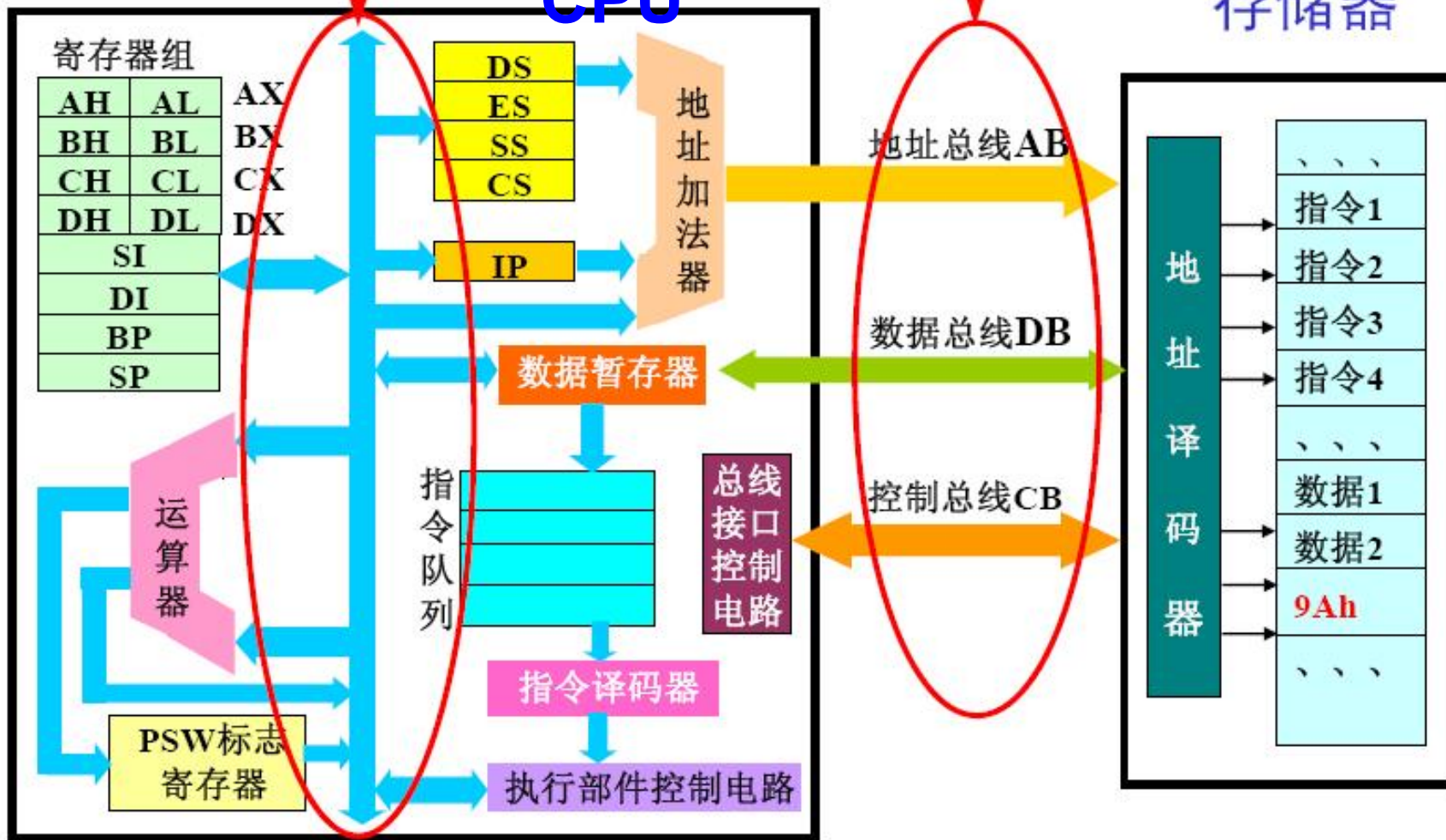


例:

(2)直接、间接、立即三种寻址方式的执行速度由快至慢依次为: 立即、直接、间接

CPU

存储器



```

OUT DX, AL
MOV DS, AX
SUB [BX], AL
MOV [BX+20], BX
ADD AL, [BX]
    
```

; DX间址, 产生 IOWR, 总线 I/O 写操作
 ; DS 为默认数据寄存器, 补发总线周期
 ; 总线读, 总线写, 总线读, 总线写
 ; → CPU 总线信号, 总线读操作

(续)

POP DS ; 1个存储器总线读周期

CALL DWORD PTR CS:[8000H]; 2个存储器总线写周期
(CS和IP)、再2个存储器总线读周期 (段基和偏移量)

INT 80H ; ?

历年考题

7. 8088/8086CPU访问存储器时，在地址总线上送出的地址信号称为_____地址。

- A. 逻辑 B. 偏移量 C有效 D. 物理

2. 8086/8088系统中复位信号RESET的作用是使_____

- A. 处理器总线休眠 B.处理器总线清零
C. 处理器和协处理器工作同步 D. MPU恢复到机器的起始状态并重新启动

（2016B）关于8086/8088 CPU中的两大部件EU和BIU，以下说法错误的是：

- A. EU内部含有算术逻辑单元ALU
B. BIU的功能是负责与存储器、I/O接口传送信息
C. BIU内部含有负责生成20位物理地址的加法器
D. EU和BIU之间通过地址总线、数据总线、控制总线这三类CPU总线交换信息

关于8086和8088 CPU的内部结构，以下说法错误的是_____。

- ☐ A EU与BIU协同工作，形成指令级流水线，提高了CPU工作效率
- ☒ B 8086的ALU是16位的，但8088的ALU是8位的
- ☐ C 8086和8088内部的BIU就能计算20位物理地址，并负责与片外存储器、I/O接口交换数据
- ☐ D 8086和8088的通用寄存器都在EU中

提交

8086/8088 CPU最大模式的特点是_____。

- ☒ A 最大模式是多处理机模式，需要协调主处理器和协处理器
- ☐ B 最大模式下的控制总线信号是由CPU产生的
- ☐ C 最大模式中不需要地址锁存器
- ☐ D 最大模式只适合8086，不适合8088

提交

下列 8086 指令中，操作数访问需要 2 个总线周期的是
_____。

- ☐ A DEC BYTE PTR[SI+200]
- ☐ B SUB SI, DX
- ☒ C CALL 2000: 3000H
- ☐ D JMP 2000: 3000H

提交

第二章、指令系统

指令——实现微机算术、逻辑运算和控制功能

重点：

1、寻址方式

2、掌握重要指令/伪指令

转移控制（转移指令、过程调用、中断）

3、结合指令/伪指令，分析内存/堆栈内容变化、
指针变化（程序阅读题）

(1) 寻址方式

□ 寻址方式（寻找操作数地址的方式），立即数/直接/寄存器/寄存器间址/变址/基+变址，掌握默认寄存器，段超越ES: CS: DS: →指令前缀

MOV BP, ES:[BP]

ADD WORD PTR CS:[SI+200],100

□ 汇编器如何找到操作数——寻址方式：

(1)在数据存储器中，指令中如何提供操作数或操作数地址的方式。

(2)在程序存储器中，程序转移时需提供转移地址，这也称为寻址。

(2) 汇编指令

□ 指令=操作码+目标，源操作数

CMP AX, [BP]

；源操作数在堆栈段中

ADD AX, WORD PTR [BX+SI]

；源操作数在数据段中

习题二. 1.(7) **PUSH [1234H]**，源操作数寻址方式：
直接寻址；目标操作数：堆栈栈顶单元

基本指令集

□ 掌握基本指令*

MOV, LEA, PUSH, POP,
ADD, SUB, INC, DEC, CMP, MUL
AND, OR, TEST, XOR
SHL, ROL (等)
JMP, JZ, JNZ, JC, JNC, LOOP
CALL, RET, INT n, IRET
CLD, STD, STI, CLI
REP MOVSB(W)
IN, OUT

□ 掌握伪指令*

DB, DW, DD; ORG, OFFSET/ SEG, \$,
(WORD/BYTE/FAR/NEAR) PTR

转移类指令JMP/CALL

□ CALL/RET改变程序流

CALL near_proc ;sp←sp-2, (SP+1:SP) ←IP
;IP←IP+disp

CALL far_proc ;sp←sp-2, (SP+1:SP) ←CS
;CS←SEG far_proc
;sp←sp-2, (SP+1:SP) ←IP
;IP←OFFSET far_proc

CALL reg16/mem16 ;例如CALL BX,
;CALL WORD PTR[BX]

CALL mem32 ;例如CALL DWORD PTR [BX]

RET ;近、远 隐含弹出

RET pop_value ;弹出后, 再(SP)←(SP)+pop_value
; pop_value为偶数

条件控制转移指令

所有条件转移指令（**JE/JZ/JC/JA/JB...**）、循环控制指令（**LOOP**）的**操作数都是一个短标号，即位移量在-128~127的范围内。**

注意：在编制大的循环程序时要注意跳转范围的限制问题，否则会出现以下的编译问题。

```
error A2053: Jump out of range by 236 byte(s)
```

因此要求在编制程序时要做到短小精悍，简洁易读

提示：条件转移指令往往与逻辑指令，移位指令，**CMP**等指令相配合。

中断指令

中断指令 **INT n** ; n中断类型号

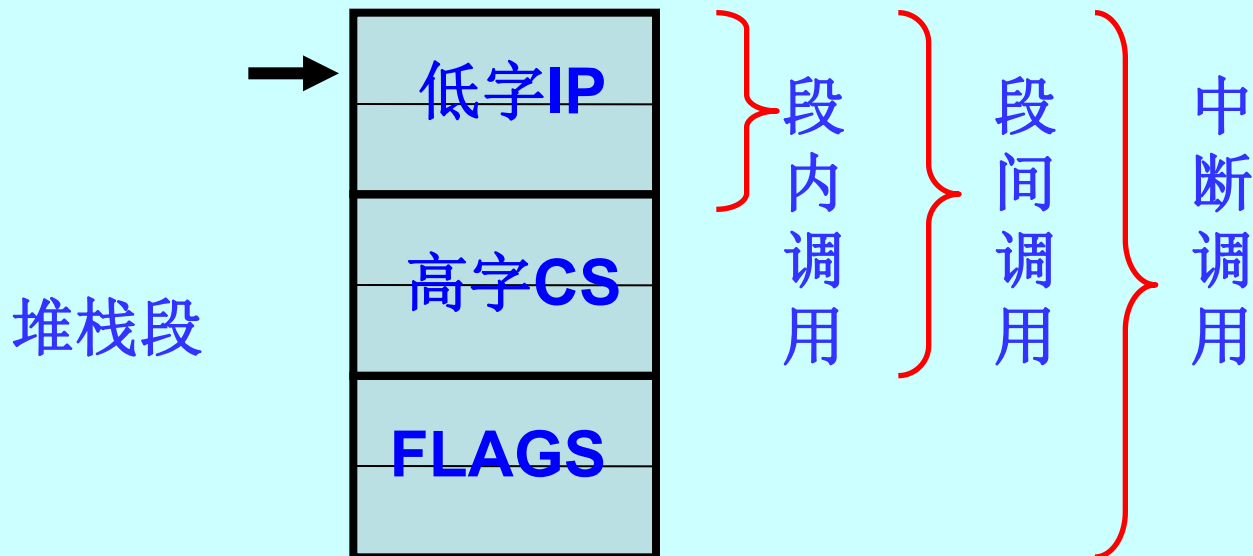
INT n ;中断指令

$SP \leftarrow SP-2, (SP+1, SP) \leftarrow FLAGS;$

$IF \leftarrow 0 \quad TF \leftarrow 0$

$SP \leftarrow SP-2, (SP+1, SP) \leftarrow CS; \text{新}CS \leftarrow [0:n*4+2]$

$SP \leftarrow SP-2, (SP+1, SP) \leftarrow IP; \text{新}IP \leftarrow [0:n*4]$



串操作指令

注意“**先期的准备工作**”：

- ①对相应的寻址寄存器进行（**DS:SI**、**ES:DI**）设置，注意段超越的问题。
- ②根据自己的程序决定对存储区的操作**方向**，即对**DF**的设置。
- ③若要进行重复操作时，一定要设置**CX**。
- ④若要进行ZF的判别，注意判别条件的设置。

串操作功能与默认操作数

例：将数据段中首地址为**BUFFER1**的**200**个字节传送到附加段中首地址为**BUFFER2**的内存中，使用串操作指令：

LEA SI, BUFFER1

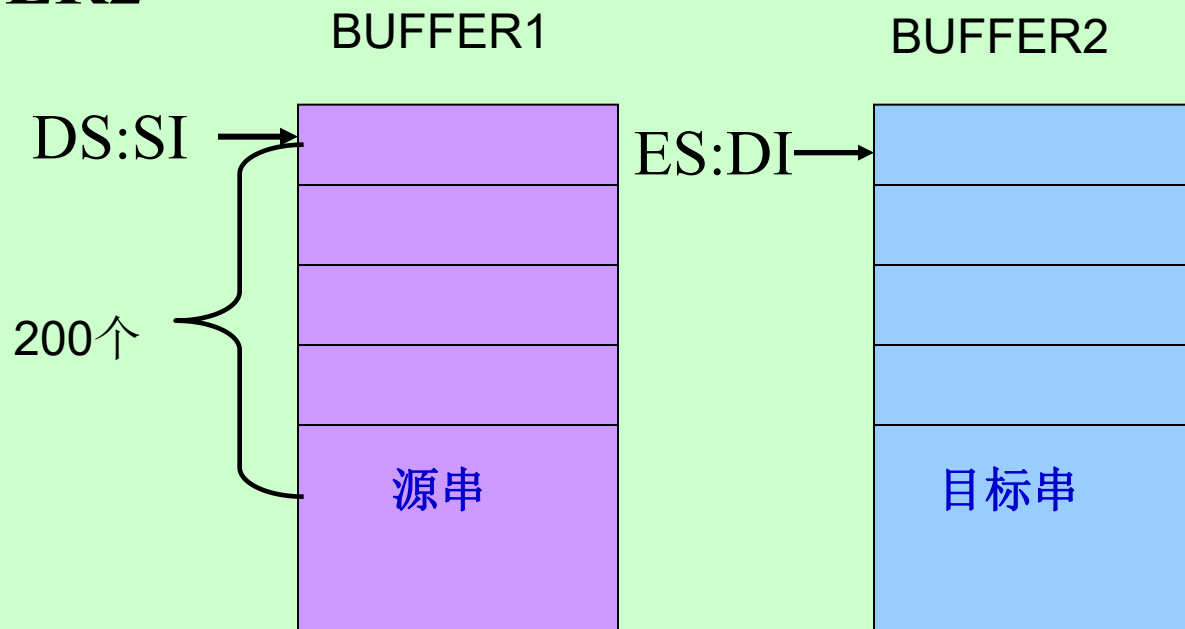
LEA DI, BUFFER2

MOV CX, 200

CLD

REP MOVSB

(按地址增方向)

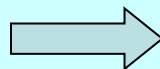


第三章、汇编语言程序设计

- 常量
- 寄存器
- 存储器操作数: 具有段属性、偏移量属性和类属性。
 - 标号: 在代码段中。
 - 变量: 存放数据的存储单元的名字, 在数据段或堆栈段中。
- 表达式: (由运算符和被操作数组成)汇编时获得。
 - 数值表达式: 常量和运算符组成, 汇编时产生立即数。
 - 地址表达式: 由常量、变量、标号、寄存器、**运算符**组成, 表示存储器的段内偏移地址。

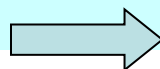
标号名和变量名——反汇编看不到

MOV DATA1, AX



MOV [0000], AX

JMP NEXT



JMP 0036

(1) 操作数、运算符

要点:

注意区分逻辑运算符和逻辑指令。

运算符

算术(+,-)逻辑(AND,OR)关系(EQ,LT)分析(OFFSET, SIZE)合成(PTR), 优先级*

运算符出现在操作数部分,在汇编时完成运算;指令出现在操作码部分,运行exe时由CPU执行。例如:

AND AL,12H AND 0FH ;等价于AND AL,02H


数值表达式作为操作数

伪指令

(1)符号定义 EQU:

定义的是**符号名**，不能重新定义，常用于定义端口。

EQU可以写在程序的任何位置。

(2)数据定义 DB,DW,DD:

定义的是变量，定义字时要遵循低位低地址，高位高地址的原则。**DB/DW/DD**只能定义在**DATA**段中。

(3)段定义 SEGMENT/ENDS ASSUME AT

(4)过程定义 PROG/ENDP

(5)模块定义 (6)宏处理

例题：

6. 在汇编语言程序中，对**END** 语句的叙述正确的是

_____。

- (A) **END** 语句是一可执行语句
- (B) **END** 语句执行与**HALT** 指令相同的功能
- (C) **END** 语句表示源程序到此结束
- (D) **END** 语句在汇编后要产生机器码

6. 在汇编语言程序中，伪指令与**CPU**指令的特点差别以下哪个是错误的说法__。

- A. **CPU**指令是在程序运行时**CPU**执行的命令，每条指令对应特定的操作；
- B. 伪指令是在编译阶段发挥作用的，由汇编器（**MASM**、**TASM**等）来解释；
- C. 伪指令既不控制机器的操作也不被汇编成机器代码；
- D. 使用**DEBUG**反汇编功能，可以看到每一条伪指令对应的机器码。

(2) 汇编语言程序设计 --数据定义

DATA SEGMENT AT A800H (段基地址)

ORG 1000H ;起始偏移量地址

DB 'ABC' ;存放在1000H→ 41H, 42H, 43H

3 DUP (5, 3 DUP (5))

\$:当前地址 (可移动)

NEXT: JMP \$; 死循环

ARRAY DB d1,d2,d3,...,dn

COUNT DB \$-ARRAY

(2) 汇编语言程序设计 --数据定义

用变量/过程名来定义新的变量

➔ 指针变量(近地址指针2字节/远地址指针4字节)

DATA1 DB 12H, 34H

DATA2 DW DATA1

DATA3 DD DATA2

MOV BX, SEG DATA1

MOV DX, OFFSET DATA1

(3) 汇编语言程序设计-存储器组织

1、汇编程序中，以下数据段段地址为 2000H，数据定义为：

DATA SEGMENT

SRT DB 'CD'

data1 DB 100, 0FFH

data2 DB 2 DUP (6, 3 DUP (2))

WV1 DW data1

DATA ENDS

(1) 画出内存数据段组织结构。

(2) 设当前 CS=1000H，执行

MOV BX, OFFSET data2

JMP DWORD PTR [BX] 指令后 CS= 0202H ✓, IP= 0206H ✓

(3) 执行 MOV AX, WV1 指令后 AX= 02H ✓。

2000H:0000	43H	SRT
2000H:0001	44H	
2000H:0002	64H	data1
2000H:0003	FFH	
2000H:0004	06H	data2
	02H	
	02H	
	02H	
	06H	
	02H	
	02H	
	02H	
	02H	
	02H	WV1
	00H	

2、采用并行接口 i8255A 每秒钟定时检测微型导线对地短路状态并构成报警系统如图 PA 口连接

1. 阅读分析程序,按要求填空回答问题(共15 分)

(1)(8 分) 汇编程序中,一数据段段基址为 (DS)=9000H, 偏移量地址5600H, 需要定义变量VSX (X=1, 2, 3), 使它们均顺序存放相同的数据: 00H, 11H, 22H, 33H, 44H, 55H, 66H, 77H, 88H, 99H, AAH, BBH。

1) 按汇编语言格式要求, 分别补充写出按字变量和双字变量定义变量VS2 和 VS3 的伪指令:

ORG 5600H

VS1 DB 00H, 11H, 22H, 33H, 44H, 55H, 66H, 77H, 88H, 99H, 0AAH, 0BBH

VS2 DW _____

VS3 DD _____

2) 执行指令MOV AX, VS2+4 后, AX=_____

3) 当前 (CS)=1000H, 执行

MOV BX,OFFSET VS2+2

JMP BX 指令后

(CS) = _____, (IP) = _____

4) 当前 (CS)=1000H, 执行

MOV BX,OFFSET VS2+2

JMP WORD PTR [BX+2] 指令后

(CS) = _____, (IP) = _____

5) JMP DWORD PTR VS2 后

(CS) = _____, (IP) = _____

答案: 1) **VS2 DW 1100H,3322H,5544H,7766H,9988H,0BBAAH (2 分)**
VS3 DD 33221100H,77665544H,0BBAA9988H(2 分)

2) **5544H (1 分)**

3) **1000H,560EH (1 分)**

4) **1000H, 5544H (1 分)**

5) **3322H, 1100H (1 分)**

(4) 汇编语言程序设计

考点：

数据块操作：字符查找、比较、最大最小值

结构：

分枝（比较--散转）/循环（条件）

要求：

流程框图、必要注释、模块化、过程调用

忠实理解题意-要求-建模分析

(4) 程序设计- (a) 检测、判断

TEST AL,80H ;或者AND
JZ NEXT ;AL的最高位=0时满足条件

OR AL,80H ;强制把AL最高位置1
AND AL, 0FH ;只取AL的低4位，而高4位清零

指令名称		助记符	测试条件
无/有符号数	相等	JZ	ZF=1
无符号数	高于	JA	CF=0 且 ZF=0
	低于	JB	CF=1 且 ZF=0
	不考虑 ZF 时，常用 JC 和 JNC 判断		
有符号数	大于	JG/JNLE	SF \oplus OF=0 且 ZF=0
	小于	JL/JNGE	SF \oplus OF=1 且 ZF=0

常用**JZ/JE**等于，**JNZ/JNE**不等，**JC**小于(无符号)，**JNC**大于(无符号)

(4) 程序设计- (b) 循环

两种常见的循环取数方式

□ 循环

LOOP计数循环 (CX次)

条件循环 (计数) **JCXZ**, **LOOPE/Z**

循环之前:

- 指针初始化

MOV BX, OFFSET CMPDAT

- 设置计数器初值

MOV CX, n

循环内部:

- 寻址

MOV AX, [BX] ;寄存器间

- 指针加减**DEC /INC**

INC BX (字节±1, 字±2, 双字±4)

或者:

MOV SI, 0

MOV AX, CMPDAT[SI]

;变址寻址

INC SI

程序编制——例子——找关键字

2006年考题

2. （20分） 设已知缓冲区bDATA存放着不超过Nmax（定义为300）个字节^{（1）}的非空8位数，以0FFH结束。现要求编写一预处理程序，将其中所有连续两个字节数值等于回车/换行字符（即0DH，0AH或0AH，0DH）的偏移量地址指针^{（2）}顺序存放在以pDATANP为首的内存字单元中，其总数存入字单元wNPSUM；如果在Nmax字节中未发现结束符0FFH,则将0FFH写入字节单元bFULL,否则将该单元清零。写出程序片断，并加必要的注释，要求用伪指令定义各存储单元和常数。

思路

2006年考题

个数Nmax，用CX存放

pDATA

12H,0DH,0AH,11H,21H,35H,0AH, 0DH,.....0FFH

↑ SI

pDATANP

0001,0006H

↑ DI

wNPSUM，保存统计个数

bFULL= 0FFH，找到结束符则赋00H

```
Nmax EQU 300
CRLF1 EQU 0D0AH
CRLF2 EQU 0A0DH
pDATA db Nmax DUP(?)
pDATANP DW 2*Nmax dup(?)
wNPSUM DW 0
bFULL DW 0FFH
```

```
MOV CX, Nmax
MOV DI, offset pDATANP
MOV SI, offset pDATA
LP1: MOV AX, [SI]
CMP AX, CRLF1
JZ FOUND
CMP AX, CRLF2
JZ FOUND
CMP [SI], 0FFH
JZ FINISH
```

```
INC SI
LOOP LP1
JMP EXIT
FOUND: MOV [DI],SI
INC DI
INC DI
INC SI
INC SI
DEC CX
JZ FINISH
INC wNPSUM
LOOP LP1
FINISH: XOR AL,AL
MOV bFULL,AL
EXIT:
```

;没有找到尾符标志，不修改bFULL

程序编制——例子——找关键字

2012年考题

2. （20 分）8086/8088 系统数据段中首地址为RBUF 的数据区中存放了一批数据，总数不超过400 字节，其中包括了一个通信报文块，以AAH，55H 开头，55H，AAH 结束，要求：

1) 编写一段程序，找到数据区中的该报文块，并把报文块（包括AAH，55H和55H，AAH）转存到数据区TSBUF（在同一数据段中，已分配空间定义为总数为404 个字节的存储变量区）。

2) 如果数据区未发现该报文块，则按空报文格式（只有AAH，55H 和55H，AAH），存入TSBUF；

3) 如果数据区只发现报文块开始字节（AAH，55H），则转存数据并自动添加结束字符（55H，AAH）。

写出程序片断，并加必要的注释。


```
MOV SI,OFFSET RBUF
MOV DI,OFFSET TSBUF
MOV CX,399
L1:  MOV AX,[SI]
      CMP AX,55AAH
      JZ HEADFOUND
      INC SI
      LOOP L1
T1:  MOV AX,55AAH
      MOV [DI],AX
      XCHG AH,AL
      MOV [DI+2],AX
      JMP EXIT
```

```
HEADFOUND:  CMP CX,2
              JZ T1
              DEC CX
              DEC CX
              MOV [DI],AX
              INC DI
              INC DI
              INC SI
              INC SI
L2:  MOV AX,[SI]
      CMP AX,0AA55H
      JZ TAILFOUND
      MOV [DI],AL
      INC SI
      INC DI
      LOOP L2
      MOV [DI],AH
      INC DI
      MOV AX,0AA55H
TAILFOUND:  MOV [DI], AX
EXIT:
```

习题3.7

在内存数据段从 **DATA1** 开始的存储单元中存放了**M**个字，试编一程序求其中最大的数。如需求绝对值最大的数，程序应如何修改？如为无符号数，求最大数的程序如何修改？

```
MOV SI, OFFSET DATA1
MOV CX, M-1
MOV AX, [SI]
INC SI
INC SI
LP1:  CMP AX, [SI];
      JGE NEXT1   ;无符号数改为JNC
      MOV AX, [SI] ;[SI]更大
NEXT1: INC SI
      INC SI
      LOOP LP1; AX=最大数
```

绝对值最大

```
        MOV SI, OFFSET DATA1;
        MOV CX,M-1
        MOV AX,[SI]
        INC SI
        INC SI
LP1:     MOV BX,[SI]    ;下一个数
        MOV DX,AX      ;当前（最大的数）
        CMP AX,0
        JGE LP2        ;AX>=0
        NEG DX          ;当前数由负变正
LP2:     CMP BX,0
        JGE LP3
        NEG BX          ;下一个数由负变正
LP3:     CMP DX,BX      ;正数比较
        JGE NEXT1      ;无符号数改为JNC
        MOV AX,[SI]     ;小于[SI]
NEXT1:  INC SI
        INC SI
        LOOP LP1        ;AX=绝对值最大数
```

习题3.11

试编写一程序，要求比较两个字符串**STRING1**和**STRING2**所含字符是否完全相同，如相同则显示‘**MATCH**’，如不相同则显示‘**NOT MATCH**’。

MATCH\$
NOMATCH\$

DB 'MATCH\$'
DB 'NOMATCH\$'

MOV AX,SEG STRING1
MOV DS, AX
MOV SI, OFFSET STRING1
MOV AX, SEG STRING2
MOV ES, AX
MOV DI, OFFSET STRING2

MOV CX, LENGTH
REP CMPS ;不强调用串指令
JNE NOMATCH

MOV DX, OFFSET MATCH\$
JMP SHORT DISPLAY

NOMATCH:
DISPLAY:

MOV DX, OFFSET NOMATCH\$
MOV AX,SEG MATCH\$
MOV DS,AX ;DS:DX指向字符串
MOV AH,9
INT 21H
RET

复习课（4-6章）

第4部分：半导体存储器

第5部分：数字量输入输出

总线、接口概念

*接口电路（芯片）、端口地址

* 数据传送方式

* 中断电路及其处理

*定时/计数器电路与应用

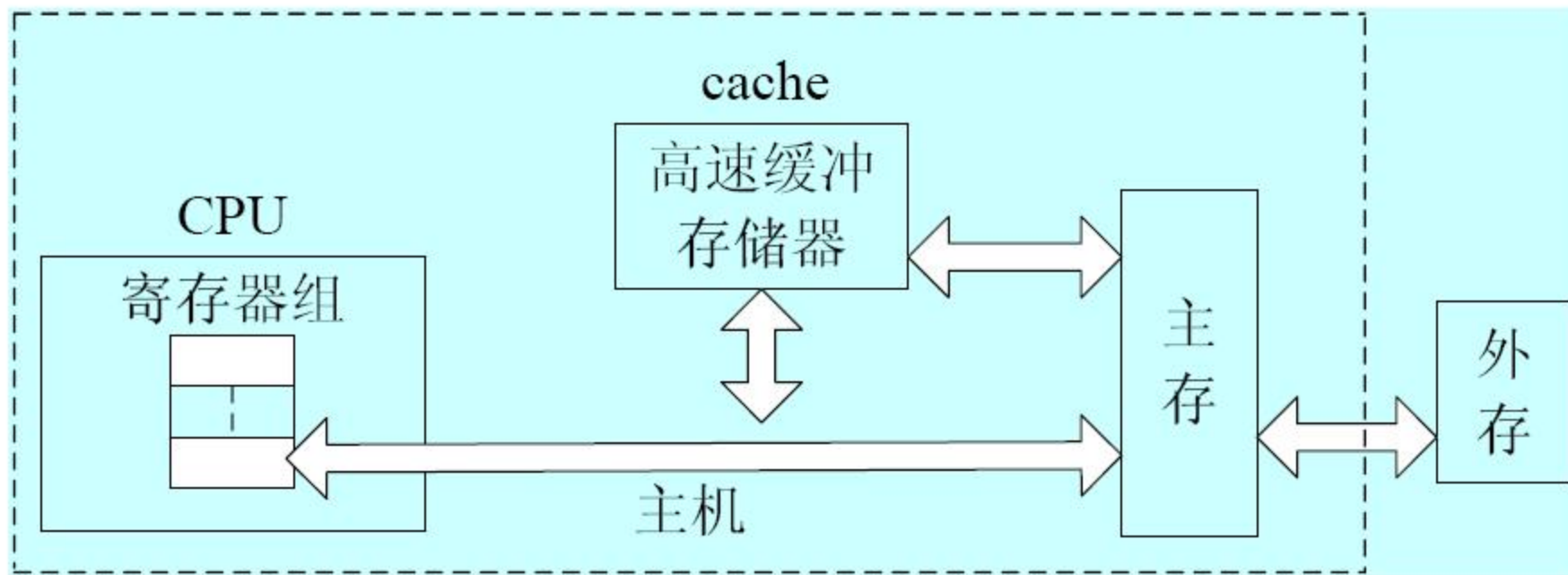
* 并行接口电路与应用

* 串行接口电路与应用

DMA电路与应用

第6部分：模拟量输入输出

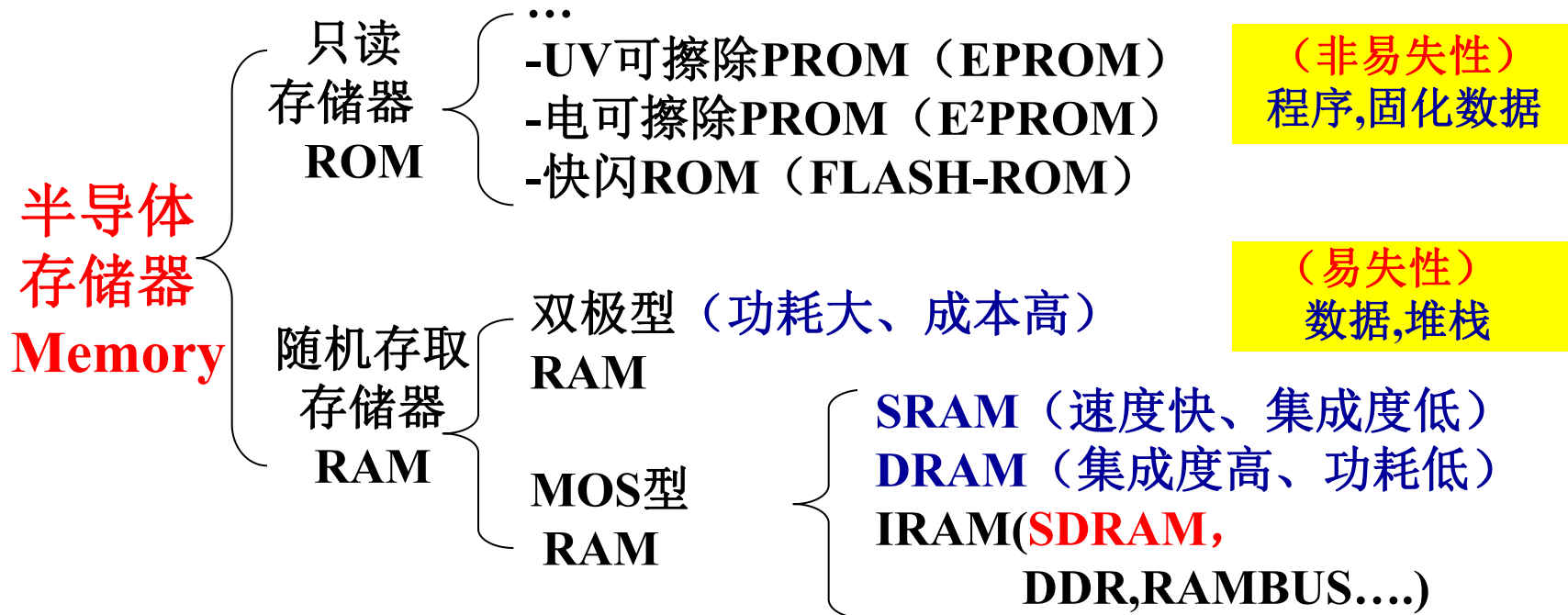
Cache—主存—外存（辅存） 三级存储系统结构



内/外存

半导体存储器分类

根据运行时存取（读写）过程的不同分类



	组成单元	速度	集成度	应用
静态RAM (SRAM)	双稳态触发器	快	低	嵌入式系统板载 (小容量)
动态RAM (DRAM)	电容	慢	高	内存条 (大容量)

半导体存储器

- *关键:与处理器的接口—正确选中、使能、读写单元
- *掌握: 存储器扩展、简单互连, 读图/设计

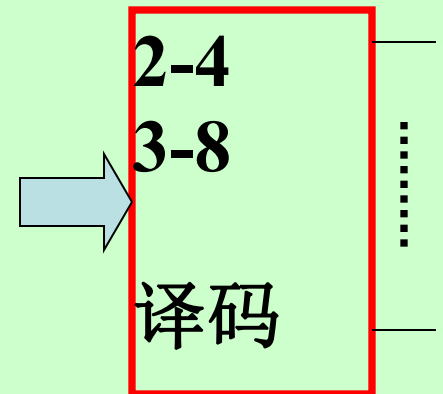
$\overline{\text{WE}}$ ($\overline{\text{MEMW}}$)、 $\overline{\text{OE}}$ ($\overline{\text{MEMR}}$)、 $\overline{\text{CE}}$ 或 $\overline{\text{CS}}$ (译码器输出)

- *常见SRAM存储器芯片容量计算: 例6116 ($2\text{K} \times 8\text{bit}$) , 数据线8根, 地址线11根

n 根地址线 = 2^n 个单元 $A_0 \sim A_{n-1}$

*译码

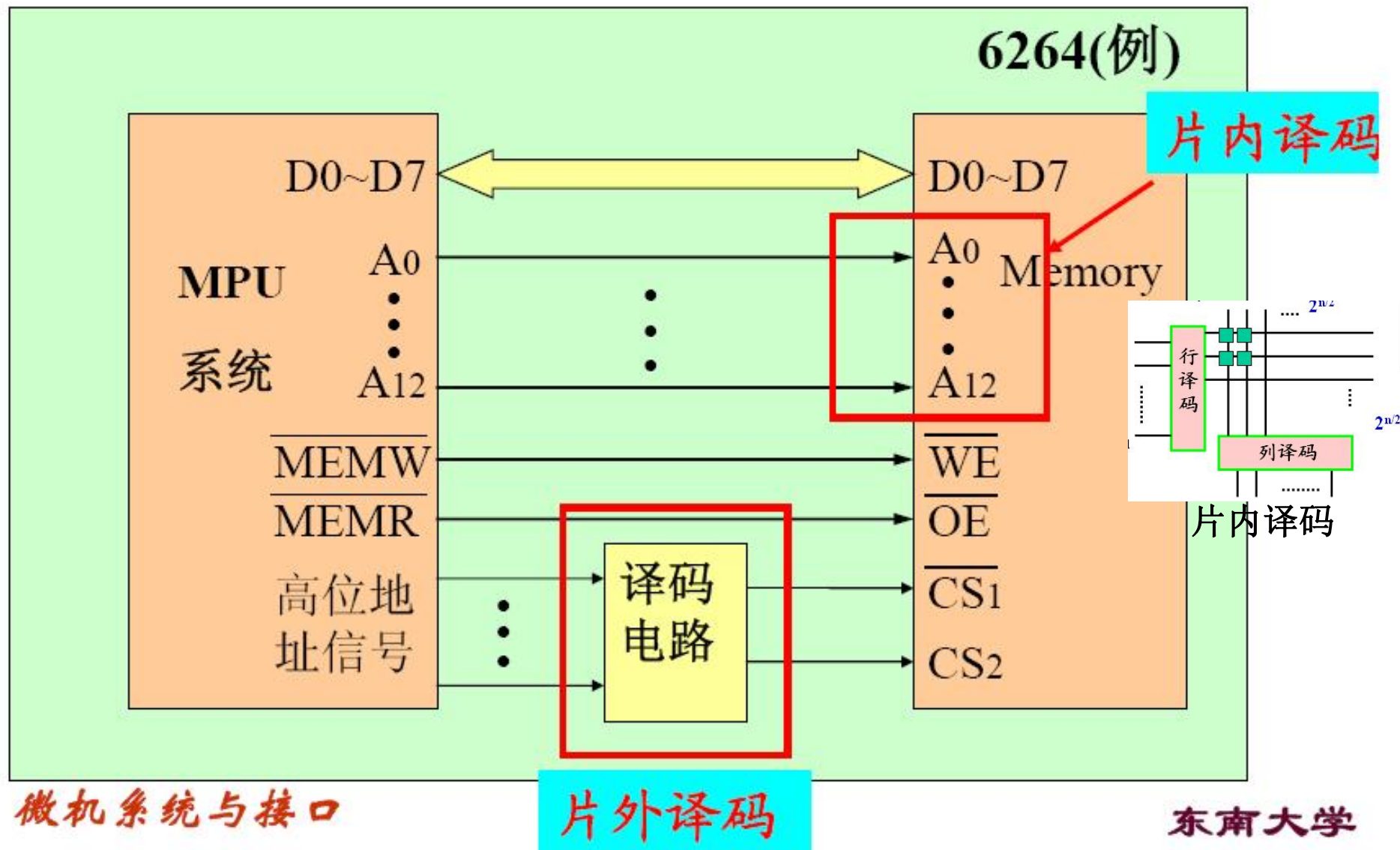
- 片选-片内两级译码
- 全译码、部分译码、线选译码 (重叠)
- 常用译码器 (74LS138) 真值表 (给出)



存储器与CPU接口

8086/8088有20根地址线

8K×8bit



存储器扩展

掌握：存储器扩展、简单互连，读图/设计

例题：用 1024×1 位的RAM芯片组成 $16K \times 8$ 位的存储器，需要多少芯片？

答：构成 $1K \times 8$ 位的存储器需要8片，因此组成 $16K \times 8$ 位的存储器需要 $16 \times 8 = 128$ 片（16组，每组8片）。

- 每片内有1024个单元，需要10根地址线。
- 每组共用一个片选，故需要16根片选信号，至少再需要4根地址线经译码器输出。

例题

(08考题 基本概念)

7. 下面四类存储器中, Cache 类型的存储器速度最快。

A. SRAM

B. ROM

C. EPROM

D. DRAM

答案： A

8. 某 DRAM 芯片, 其存储容量为 512K*8 位, 该芯片的地址线 and 数据线数目分别为 。答案: D

A. 8, 512

B. 512, 8

C. 18, 8

D. 19, 8

(06考题)

7. 设计 128M 字节的系统存储器, 需要选用 () 片 8M*4Bit 的存储器。

A. 4

B. 8

C. 16

D. 32

E.128

答案：D

例题

(12年考题, 本题可忽略)

7. 27C256是32KB的8位EPROM, 当用两片构成16位存储器时, 高8位27C256的地址 A_0 应该_____, 低8位27C256的地址 A_0 应该_____。

- A、直接连接 CPU 的 A_0 , 也直接连接 CPU 的 A_0
- B、通过反相器连接 CPU 的 A_0 , 也通过反相器连接 CPU 的 A_0
- C、直接连接 CPU 的 A_0 , 通过反相器连接 CPU 的 A_0
- D、通过反相器连接 CPU 的 A_0 , 直接连接 CPU 的 A_0

答案: D