汇编语言与接口技术

教材及参考资料：

1、微机原理与接口技术（第2版）　　朱晓华　电子工业出版社

2、 微机原理与接口技术（第三版）　彭虎　　电子工业出版社

3、IBM-PC宏汇编语言程序设计　　于春凡　南开大学出版社

4、IBM-PC汇编语言程序设计　　沈美明　清华大学出版社

5、微型计算机技术及应用　　戴梅萼　清华大学出版社

6、汇编语言与PC技术　　朱连章　石油大学出版社

史永宏： shiyh@upc.edu.cn

18953216961　　86981906

第1章　微机基础

目录

1.1 微机概述

1.1.1 微机发展概况

- 1946年第一台电子计算机问世
- 1971年，美国Intel公司研究并制造了I4004微处理器芯片。该芯片能同时处理4位二进制数，集成了2300个晶体管，每秒可进行6万次运算，成本约为200美元。它是世界上第一个微处理器芯片，以它为核心组成的MCS-4计算机，标志了世界第一台微型计算机的诞生。
- 微机概念：以大规模、超大规模构成的微处理器作为核心，配以存储器、输入/输出接口电路及系统总路线所制造出的计算机。
- 划分阶段的标志：以字长和微处理器型号。
- 特点：速度越来越快
  容量越来越大
  功能越来越强

在微机的发展过程中，最为成功也最有影响力的是IBM PC系列微机。

第一台IBM-PC：1982年，采用Intel 8088 CPU的准16位微机IBM PC。

- **微机的体系结构：冯.诺依曼建立的存储程序概念**
  - 计算机的组成
  - 二进制表示指令和数据
  - 程序和数据存放在存储器中

- 微机采用了分层的存储器系统.

  存储器可分为5层：

  0层通常是CPU内部寄存器,离CPU最近,存取速度快,但数量有限.

  1层存储器是高速缓冲存储器Cache

  2层是主存储器,通常由动态RAM(DRAM)组成

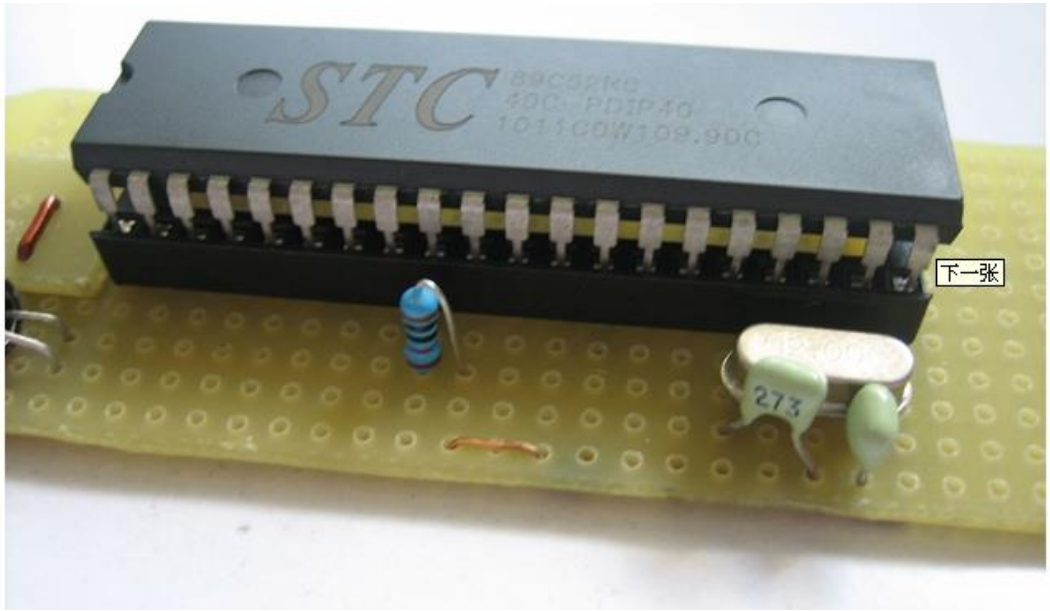
  3层是大容量的虚拟存储器（磁盘存储器）

  4层是外存储器（光存储介质等）

- **软件系统的发展同样发展迅速。操作系统最为常见的5种：DOS，Windows, Linux, UNIX/Xenix, OS/2.**

  本课程是以MS-DOS为操作系统介绍80x86系列微机的原理及接口技术

- 在微机家族中，**单片微机**的发展同样十分引人注目。单片机是把CPU、一定容量的存储器和必要的I/O接口电路集成在一个芯片上构成的具有计算机的完整功能的一种微机.



1.1.2 微机的应用

1.工业控制
2.事物处理
3.计算机辅助设计和辅助制造（CAD/CAM）
4.教学培训
5.家庭娱乐和家政事务管理
6.科学和工程计算
7.人工智能

1.2 计算机中数的表示和编码

主要内容： 1.计算机中的进位计数制

2.计算机中常用的编码

3.带符号数的表示

1.2.1 计算机中的进位计数制

1.进位计数制的表示法

- 十进制数

  十进制数是大家熟悉的，用0，1，2，…，8，9十个不同的符号来表示数值，它采用的是“逢十进一，借一当十”的原则。

- 二进制表示法

  基数为10的记数制叫十进制；基数为2的记数制叫做二进制。

  二进制数的计算规则是“逢二进一，借一当二”。

八进制表示法

八进制数是基数为八的计数制。八进制数主要采用0，1，2，…，7这八个阿拉伯数字。

八进制数的运算规则为“逢八进一，借一当八”。

八进制表示数值方法如下：

例：　$(467.6)_O$=4 * 8^2 + 6 * 8^1 + 7 * 8^0 + 6 * 8^(-1)

十六进制表示法

基数为16，用0 - 9 、A - F 十五个字符来数值，逢十六进一。

各位的权值为 16^I

十六进制表示数值方法如下：

NH= ± Ki * 16^i　　其中：Ki = 0 - 9 、A – F

例：　(56D.3)H = 5 * 16^2 + 6 * 16^1 + 13 * 16^0 + 3 * 16^(-1)

2.进位计数制之间的转换

- **二进制数和十进制数之间的转换**
  - **二进制数转换为十进制数**

方法：按二进制数的位权进行展开相加即可。

例:11101.101

$=1\times2^4+1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$

$=16+8+4+0+1+0.5+0.25+0.125=29.875$

- **十进制数转换为二进制数**

  方法：

  A、将整数部分和小数部分分别进行转换，然后再把转换结果进行相加。

  B、整数转换采用除2取余法：用2不断地去除要转换的数，直到商为0。再将每一步所得的余数，按逆序排列，便可得转换结果。

  C、小数转换采用乘2取整法：每次用2与小数部分相乘，取乘积的整数部分，再取其小数部分乘2直到小部分为0。将所取整数顺序放在小数点后即为转换结果。

- **二进制数和八进制数、十六进制数间的转换**
- **二进制数到八进制数、十六进制数的转换**

A、二进制数到八进制数转换采用“三位化一位”的方法。从小数点开始向两边分别进行每三位分一组，向左不足三位的，从左边补0；向右不足三位的，从右边补0。

B、二进制数到十六进制数的转换采用“四位化一位”的方法。从小数点开始向两边分别进行每四位分一组，向左不足四位的，从左边补0；向右不足四位的，从右边补0。

- **八进制、十六进制数到二进制数的转换**

  方法：采用“一位化三位（四位）”的方法。按顺序写出每位八进制（十六进制）数对应的二进制数，所得结果即为相应的二进制数

1.2.2 计算机中常用的编码

BCD码：用二进制编码表示十进制数

0000：0
0001：1
：
：
1001：9

符号信息的编码

ASCII码 — 美国标准信息交换代码

采用7位二进制代码对字符进行编码

例：
‘A’ 41H
‘a’ 61H
‘1’ 31H
换行 0AH
回车 0DH
空格 20H

ASCII码表

ASCII 字符表

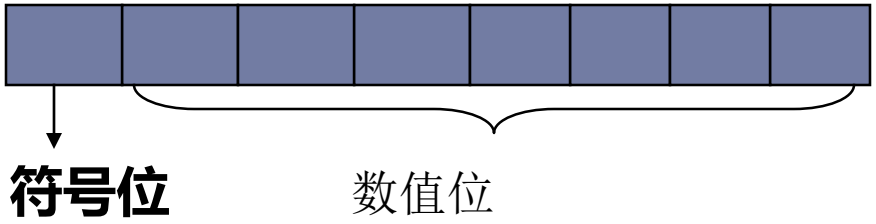
H / L	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENG	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	SI	US	/	?	O	←	o	DEL

注：H 表示高 3 位，L 表示低 4 位。

1.2.3 带符号数的表示

1、数的表示：

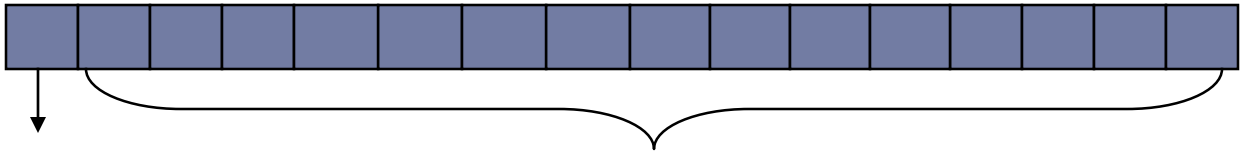
假设机器字长为8位：7 6 5 4 3 2 1 0



符号位　　数值位

假设机器字长为16位：

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



符号位 = 0　正数　　数值位

= 1　负数

2、数的常用表示法：原码 反码 补码

原码表示法：符号 + 绝对值

例：n = 8bit

$[+3]_{原码}$ = 0 000,0011 = 03H

$[-3]_{原码}$ = 1 000,0011 = 83H

$[+0]_{原码}$ = 0 000,0000 = 00H

$[-0]_{原码}$ = 1 000,0000 = 80H　　**∴ 0 的表示不唯一**

反码表示法：正数的反码同原码，负数的反码数值位与原码相反

例：n = 8bit

$[+5]_{反码}$ = 0 000,0101 = 05H

$[-5]_{反码}$ = 1 111,1010 = FAH

$[+0]_{反码}$ = 0 000,0000 = 00H

$[-0]_{反码}$ = 1 111,1111 = FFH　　∴ **0** 的表示不唯一

补码表示法：

正数的补码： 同原码

负数的补码：

（1）写出该负数的原码形式

（2）符号位保持不变，数值位按位求反，末位加一

例： 机器字长8位，$[-46]_{补码}$ = ？

$[-46]_{原码}$ = 1 0 1 0 1 1 1 0

$[-46]_{反码}$ = 1 1 0 1 0 0 0 1

$[-46]_{补码}$ = 1 1 0 1 0 0 1 0 = D2H

机器字长16位，$[-46]_{补码}$ = FFD2H

$[+0]_{补码}$ = 0 0 0 0 0 0 0 0

1 1 1 1 1 1 1 1

0 0 0 0 0 0 0 0 = 00H = $[-0]_{补码}$

∴ 0 的表示唯一

n位二进制补码的表数范围： $-2^{n-1} \le N \le 2^{n-1}-1$

十进制	二进制	十六进制	十进制	十六进制
n=8			n=16	
+127	0111 1111	7F	+32767	7FFF
+126	0111 1110	7E	+32766	7FFE
…	…	…	…	…
+2	0000 0010	02	+2	0002
+1	0000 0001	01	+1	0001
0	0000 0000	00	0	0000
-1	1111 1111	FF	-1	FFFF
-2	1111 1110	FE	-2	FFFE
…	…	…	…	…
-126	1000 0010	82	-32766	8002
-127	1000 0001	81	-32767	8001
-128	1000 0000	80	-32768	8000

补码的加法和减法：

求补运算 ⇒ ：对一个二进制数按位求反、末位加一

$$[X]_{补码} \xrightarrow{求补} [-X]_{补码} \xrightarrow{求补} [X]_{补码}$$

加法规则：$[X+Y]_{补码} = [X]_{补码} + [Y]_{补码}$

减法规则：$[X-Y]_{补码} = [X]_{补码} + [-Y]_{补码}$

例：**补码减法可转换为补码加法**

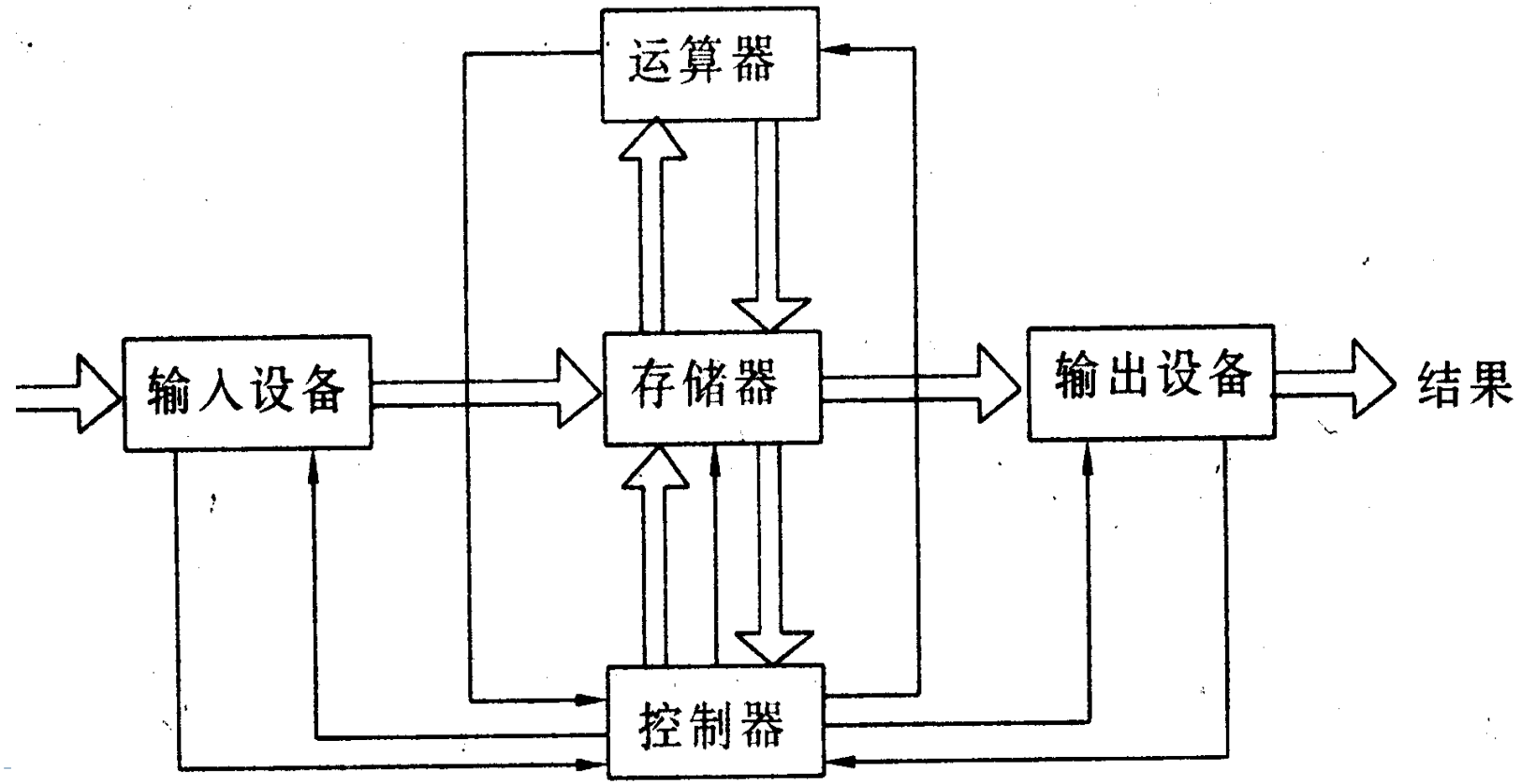
	64	0100 0000
+	(-46)	+ 1101 0010
	18	0001 0010

1.3 微机的一般概念

主要内容： 1.计算机的基本组成和工作原理
2.名词术语
3.微机结构
4. 微机的工作过程
5. 计算机软件系统

1.3.1 计算机的基本组成和工作原理和结构：

1. 计算机的基本组成：运算器、控制器、存储器、以及输入和输出设备。



微处理器

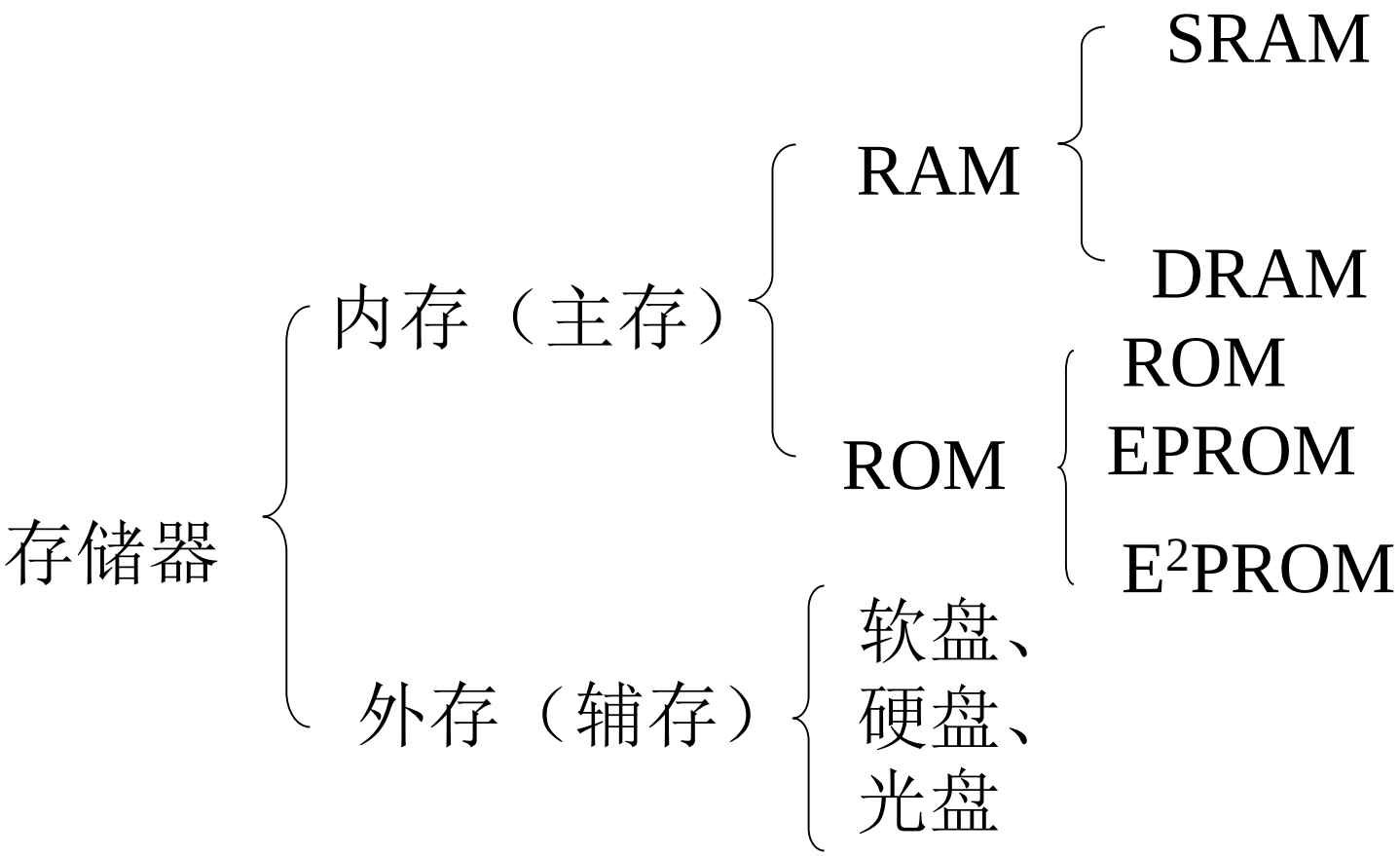
微处理器（CPU）是大规模集成电路技术做成的芯片，芯片内集成有控制器、运算器和寄存器等相关部件，完成对计算机系统内各部件进行统一协调和控制。

控制器：根据程序中的命令发出各种控制信号，使各部分协调工作以完成指令所要求的各种操作。

运算器：对信息进行加工、运算的部件，执行算术运算和逻辑运算。

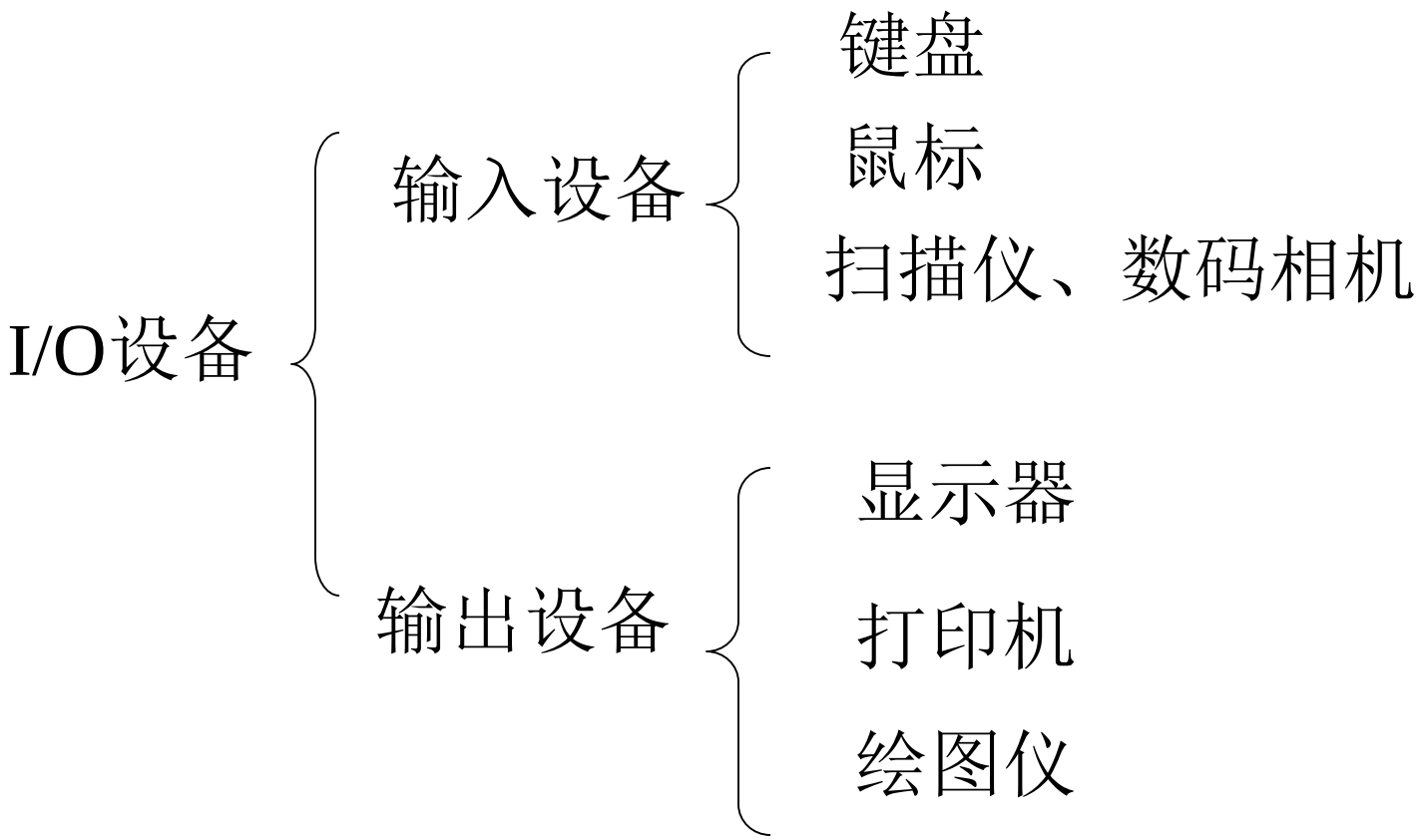
存储器

功能：存放程序和数据。



I/O设备和I/O接口

I/O设备：微机配备的输入/输出设备（外设）。



2.存储程序工作原理：

把编制好的程序和数据一起先送入存储器中保存起来。启动机器运行后，根据给出的程序中第一条指令的存储地址，控制器就可以根据存储程序中的程序周而复始的取出指令、分析指令、执行指令，直至完成全部指令操作，即控制器通过指令流的串行驱动实现程序控制。

1.3.2 名词术语

1.微处理器：是将运算器和控制器做在一块集成电路上的一个独立部件。它具有解释指令、执行指令和与外界交换数据的能力。

2.微机：通过总线把I/O、CPU和半导体存储器有机结合在一起。

微机分为：单板机（印制电路板）、单片机（芯片）、多板机。

3.微机系统：微机配上外部设备、系统电源和系统软件就构成微机系统。

。

4.**微机多机系统：**多台微机/微处理器组合而成。

5.**微机开发系统（MDS）：**在研制开发微机应用系统时，从程序调试到样机的系统调试，他都能提供软件和硬件的支持

6.**计算机网络系统：**借助通信网络将一定的域内的众多计算机和外设连接起来构成计算机网络可以实现计算机之间的互相通信和资源共享。

7.**多媒体：**文、图、声、像等单媒体与计算机程序融合在一起形成的信息传播媒体。

1.3.3 微机结构

总线使计算机功能部件的相互关系，变成各个功能部件面向总线的单一关系

计算机各个部件挂在总线上

优点：①结构简单

②维护扩展方便

- 微机的结构是一种总线结构

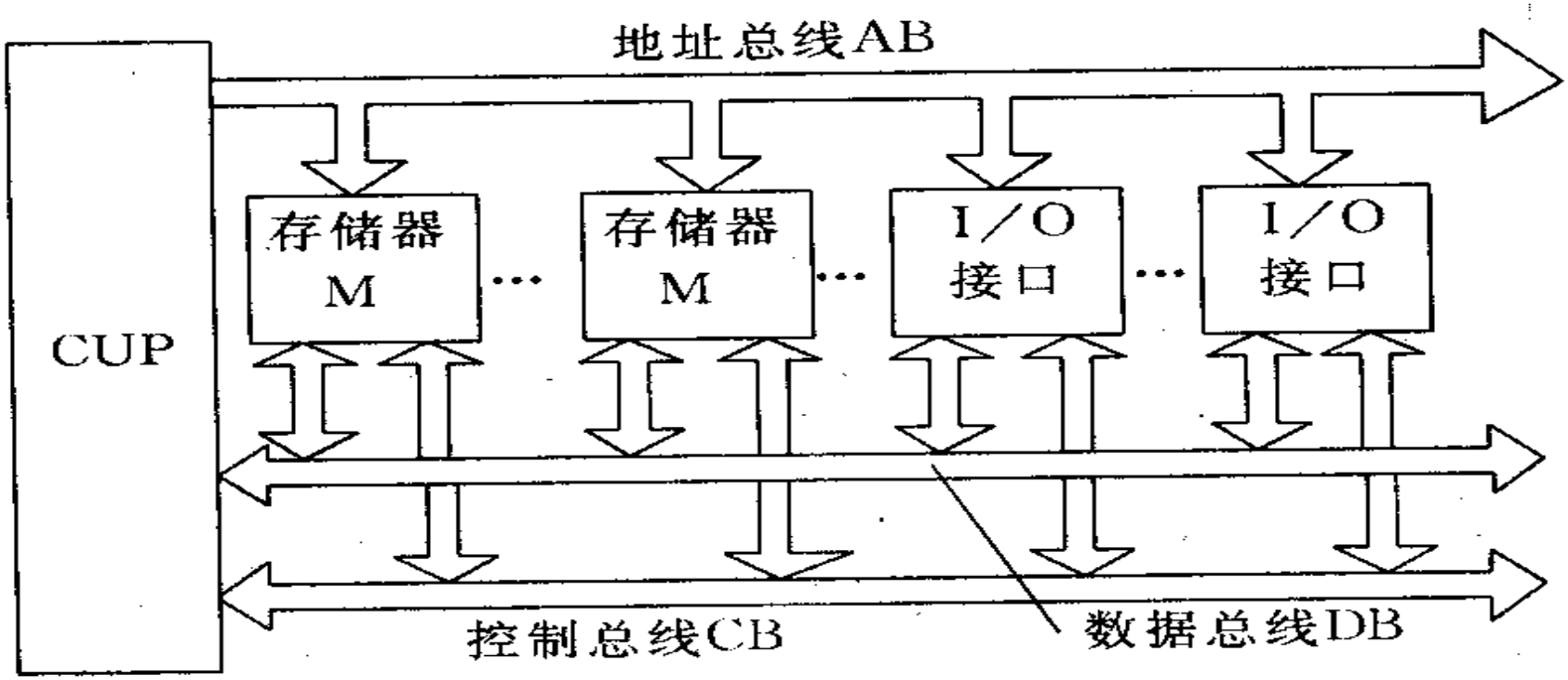


图 1-2　微机的结构

- **总线（BUS）:传递信息的一组公用导线。**

  系统总线：从处理器引出的若干信号线，CPU通过它们与存储器或I/O设备进行信息交换。

  一个部件只要满足总线标准，就可以连接到采用这种总线标准的系统中。

系统总线分为：

地址总线：传递地址信息的总线，即AB。CPU在地址总线上输出将要访问的内存单元或I/O端口的地址，该总线为单向总线。

内存容量的计算：

16条地址线可访问2^16 = 64 KB。

20条地址线可访问 2^20 = 1 MB。

1K = 1024B　1M = 1024 KB　1G = 1024 MB

数据总线：传递数据信息的总线，即DB。

在CPU进行**读**操作时，内存或外设的数据通过数据总线送往CPU；

在CPU进行**写**操作时，CPU数据通过数据总线送往内存或外设，数据总线是双向总线。

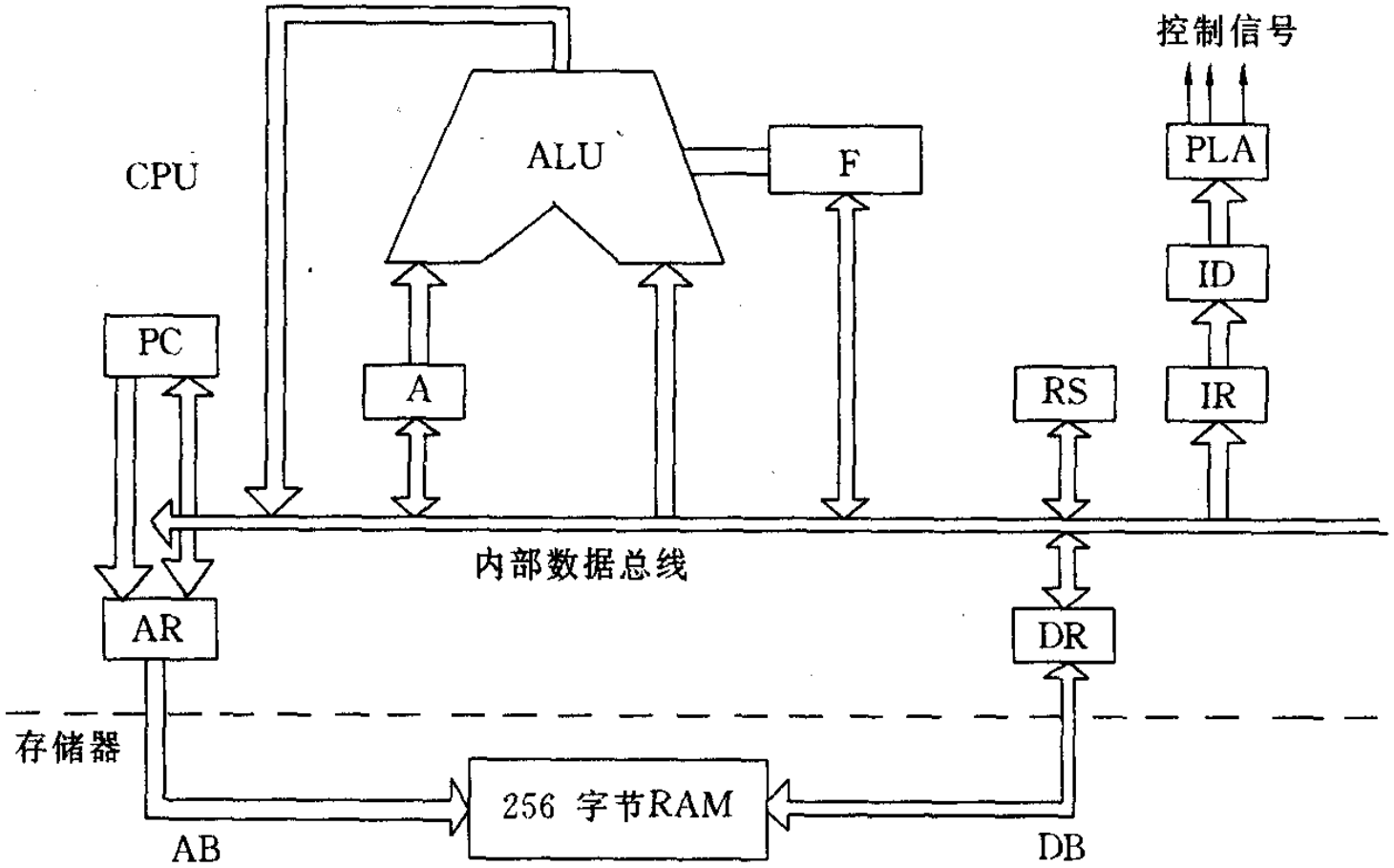
控制总线：传递控制信息的总线，即CB。

一部分是从CPU输出：通过对指令的译码，由CPU内部产生，由CPU送到存储器、输入/输出接口电路和其它部件。如读写控制信号等。

另一部分是由系统中的其他外设产生，送往CPU，如：中断请求信号、总线请求信号、状态信号。

微处理器

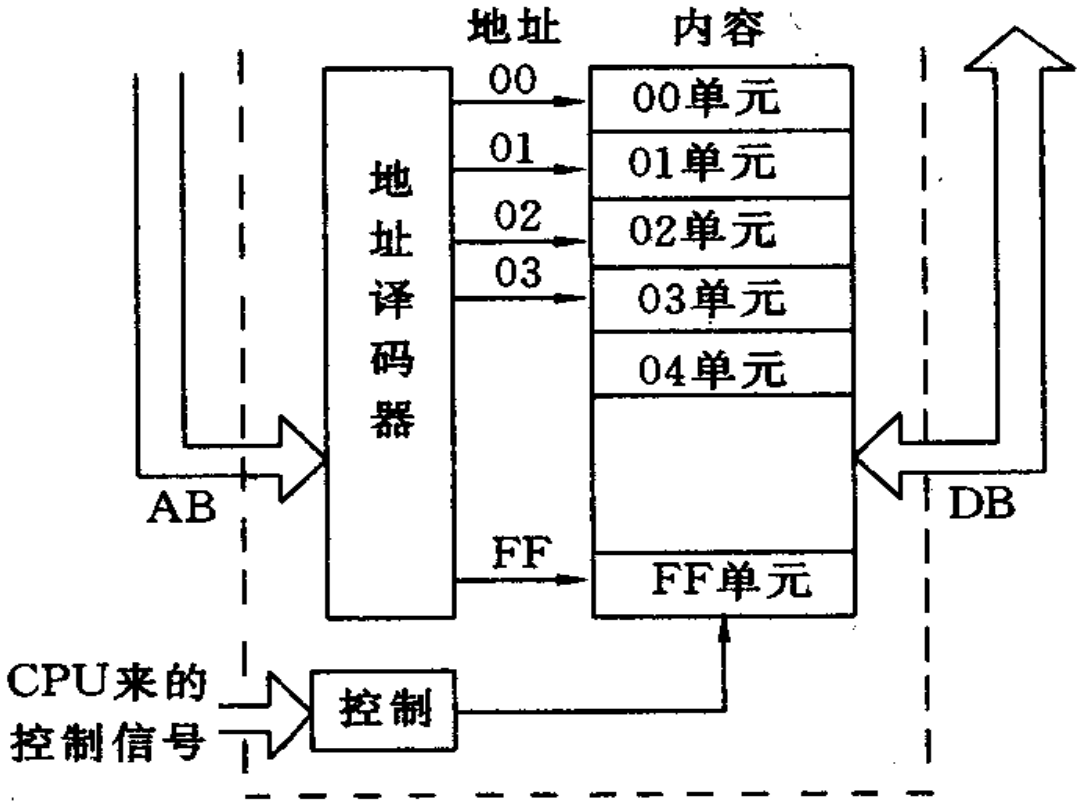
包括：寄存器阵列（RS）、算术和逻辑运算单元（ALU)、控制器、内部总线及缓冲器



存储器

每一个存储单元有一个确定的**地址**。

注意： 存储单元的地址和该地址单元中存放的内容是两个不同的概念。



1.3.4 微机的工作过程：

- **在进行计算前，应做如下工作：**

(1)　编写程序（源程序）；

(2)、将源程序汇编或编译成计算机能识别的机器语言程序；

(3)、将数据和程序放入存储器中存放。

例：完成5+9=？的程序：

```
MOV A，05H    /B0H  05H    ；把05送入累加器A
ADD A，09H    /04H   09H    ；06与A中内容相加，
                             结果存入累加器A
HLT                /F4H          ；停止所有操作。
```

1、取指令阶段的执行过程：（设程序从00H开始存放）

（1）、将程序计数器（PC或IP）的内容送地址寄存器AR。

（2）、程序计数器PC的内容自动加1变为01H，为取下一条指令作好准备。

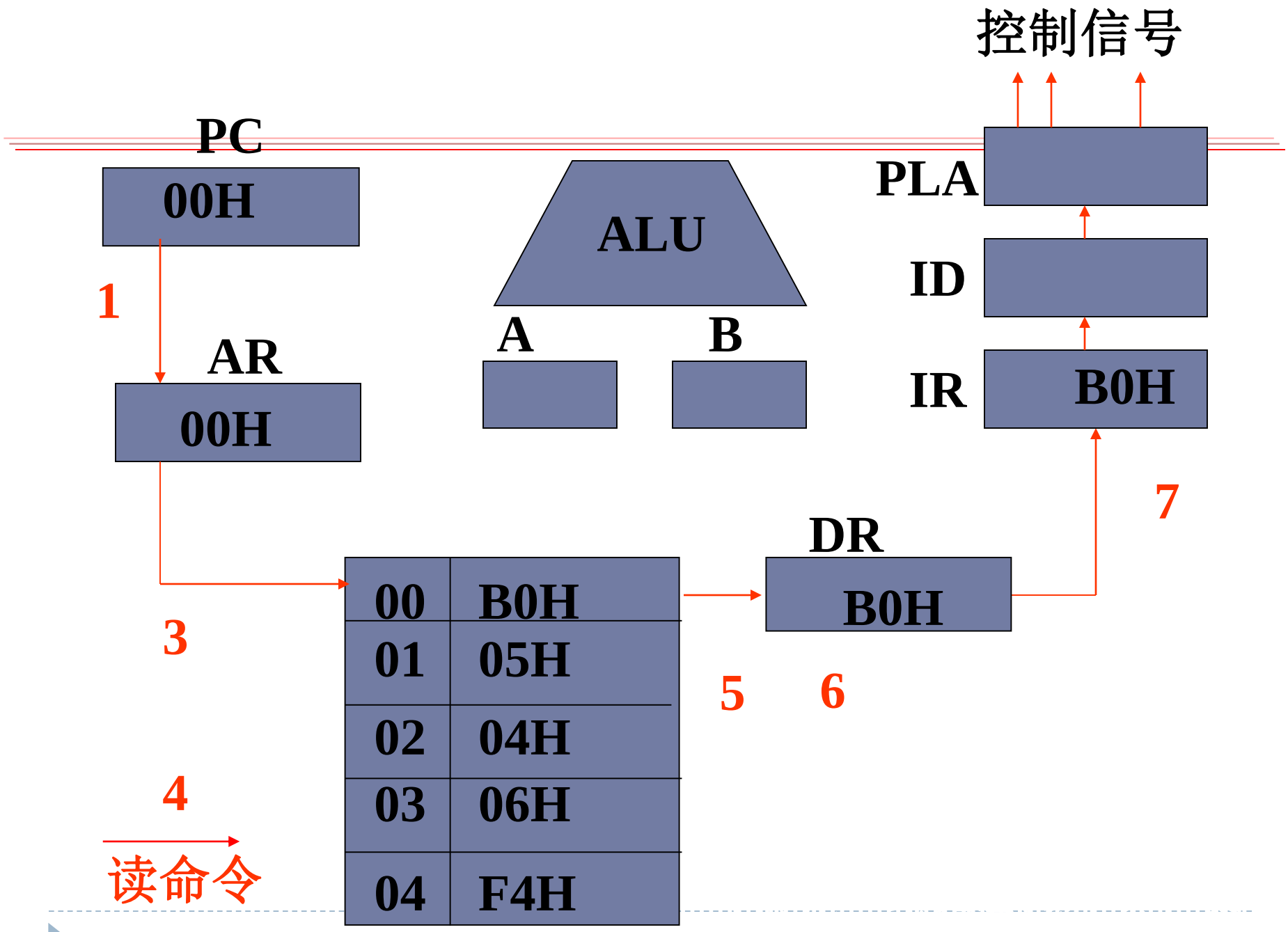
（3）、地址寄存器AR将00H通过地址总线送至存储器地址译码器译码，选中00H单元。

（4）、CPU发出“读”命令。

（5）、所选中的00单元的内容B0H读至数据总线DB上。

（6）、经数据总线DB，读出的B0H送至数据寄存器DR。

（7）、数据寄存器DR将其内容送至指令寄存器IR中，经过译码CPU“识别”出这个操作码为“MOV A，05H”指令，于是控制器发出执行这条指令的各种控制命令。


控制信号
PC
00H
1
AR
00H
ALU
A
B
PLA
ID
IR
B0H
7
DR
B0H
00 B0H
01 05H
02 04H
03 06H
04 F4H
3
5
6
4
读命令

2、执行指令阶段的执行过程：

（1）、将程序计数器（PC或IP）的内容送地址寄存器AR。

（2）、程序计数器PC的内容自动加1变为02H，为取下一条指令作好准备。

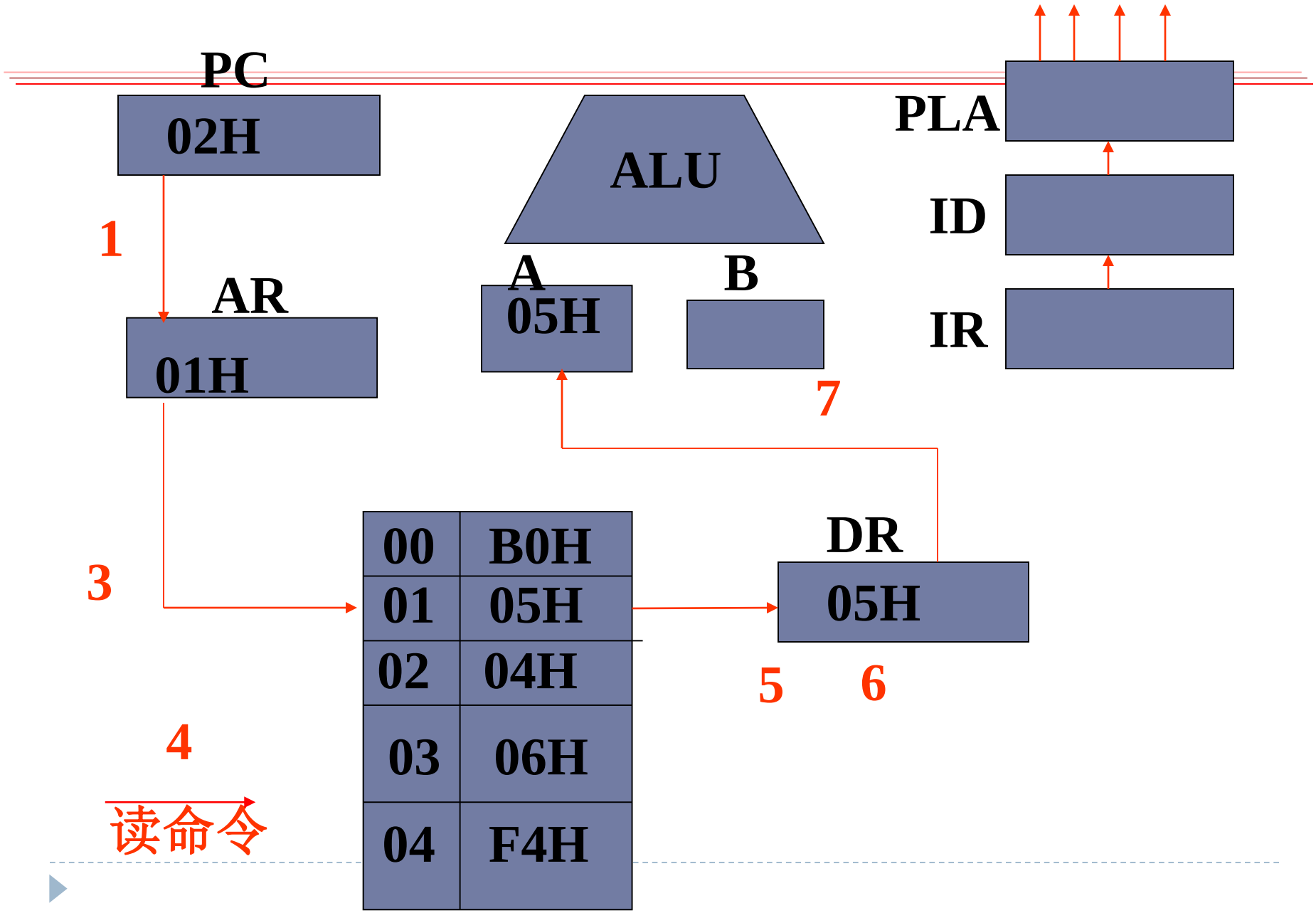
（3）、地址寄存器AR将01H通过地址总线送至存储器地址译码器译码，选中01H单元。

（4）、CPU发出“读”命令。

（5）、所选中的01H单元的内容05H读至数据总线DB上。

（6）、经数据总线DB，读出的05H送至数据寄存器DR。

（7）、由控制码计算机已知到读出的是立即数，并要求将它送入累加器A中，所以数据寄存器DR通过内部总线将05H送入累加器A中。


PC
02H
1
AR
01H
ALU
A
05H
B
PLA
ID
IR
7
3
00 B0H
01 05H
02 04H
03 06H
04 F4H
DR
05H
5
6
4
读命令

1.3.4 计算机软件系统

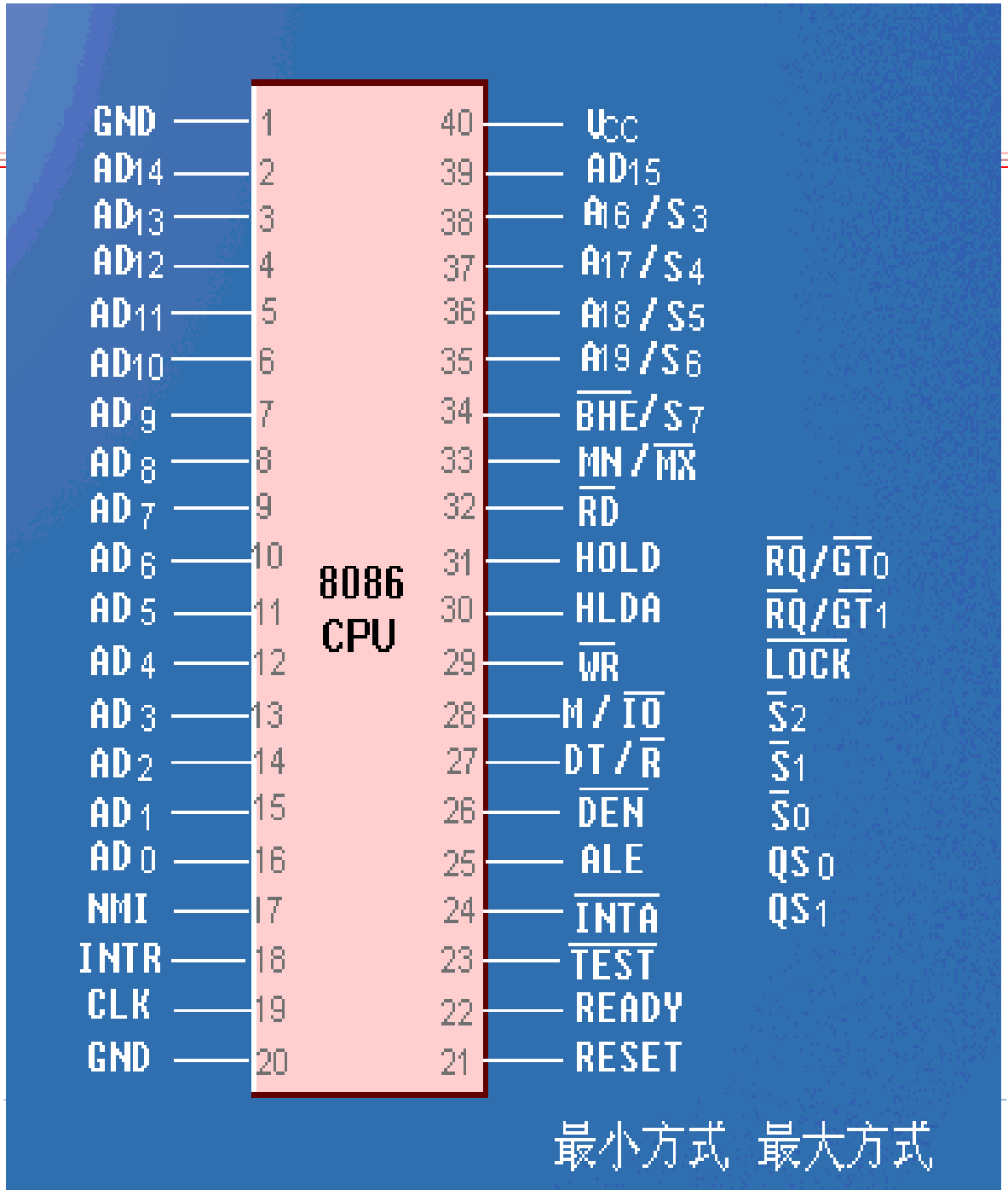
- 系统软件和应用软件
- 系统软件包括操作系统、各种高级语言处理程序、编译系统和其他服务程序、数据库管理系统等软件。这些软件不是用来解决具体应用问题的，而是利用计算机自身的功能，合理的组织解题流程，管理计算机软、硬件各种资源，提供人-机间的接口，从而简化或代替各环节中人所承担的工作。还可以为用户使用机器提供方便，扩大机器功能，提高工作效率。
- 应用软件是由用户利用计算机及其系统软件编制的解决实际应用问题的程序。

1.4 Intel微处理器结构

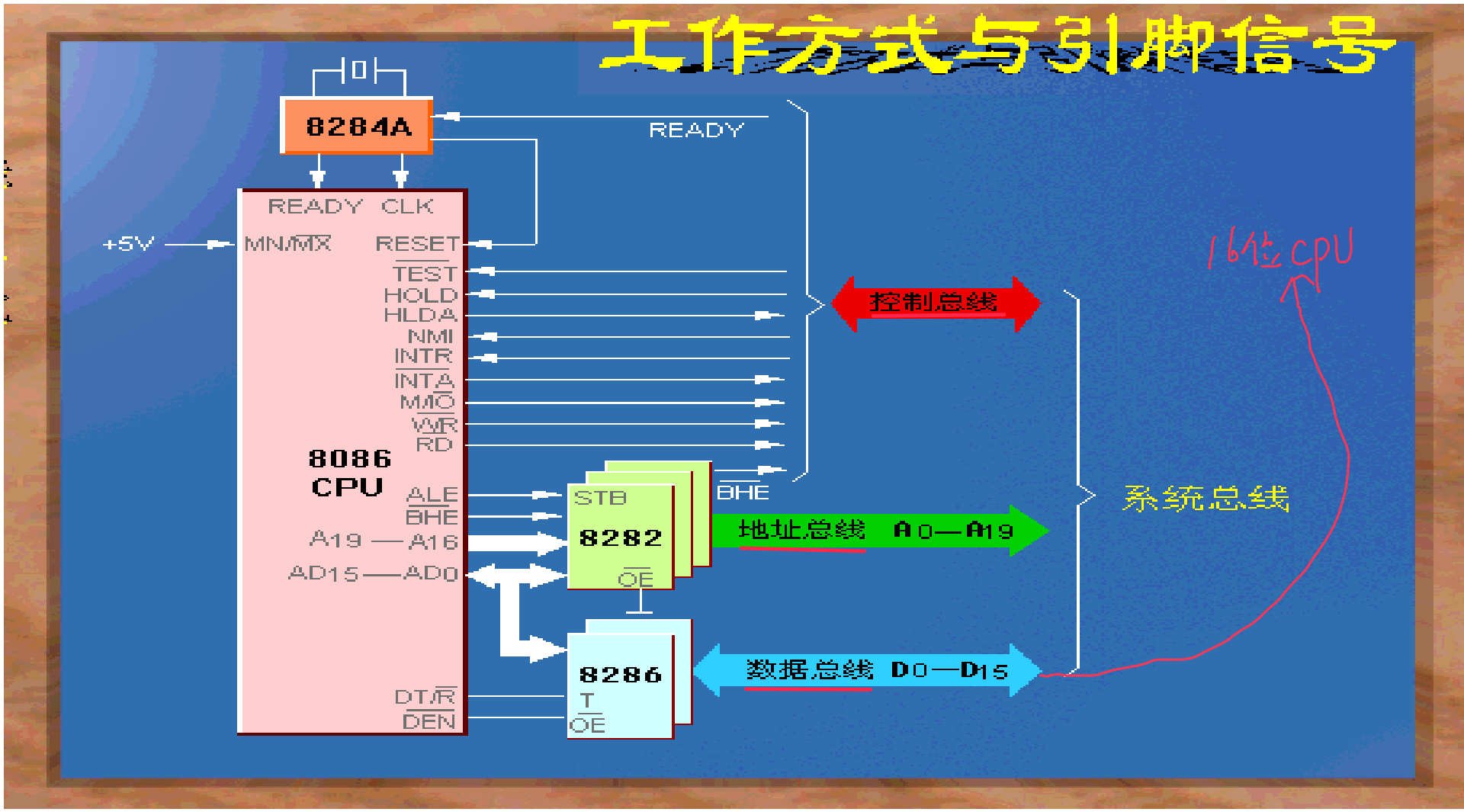
Intel 8086/8088微处理器结构

主要内容：

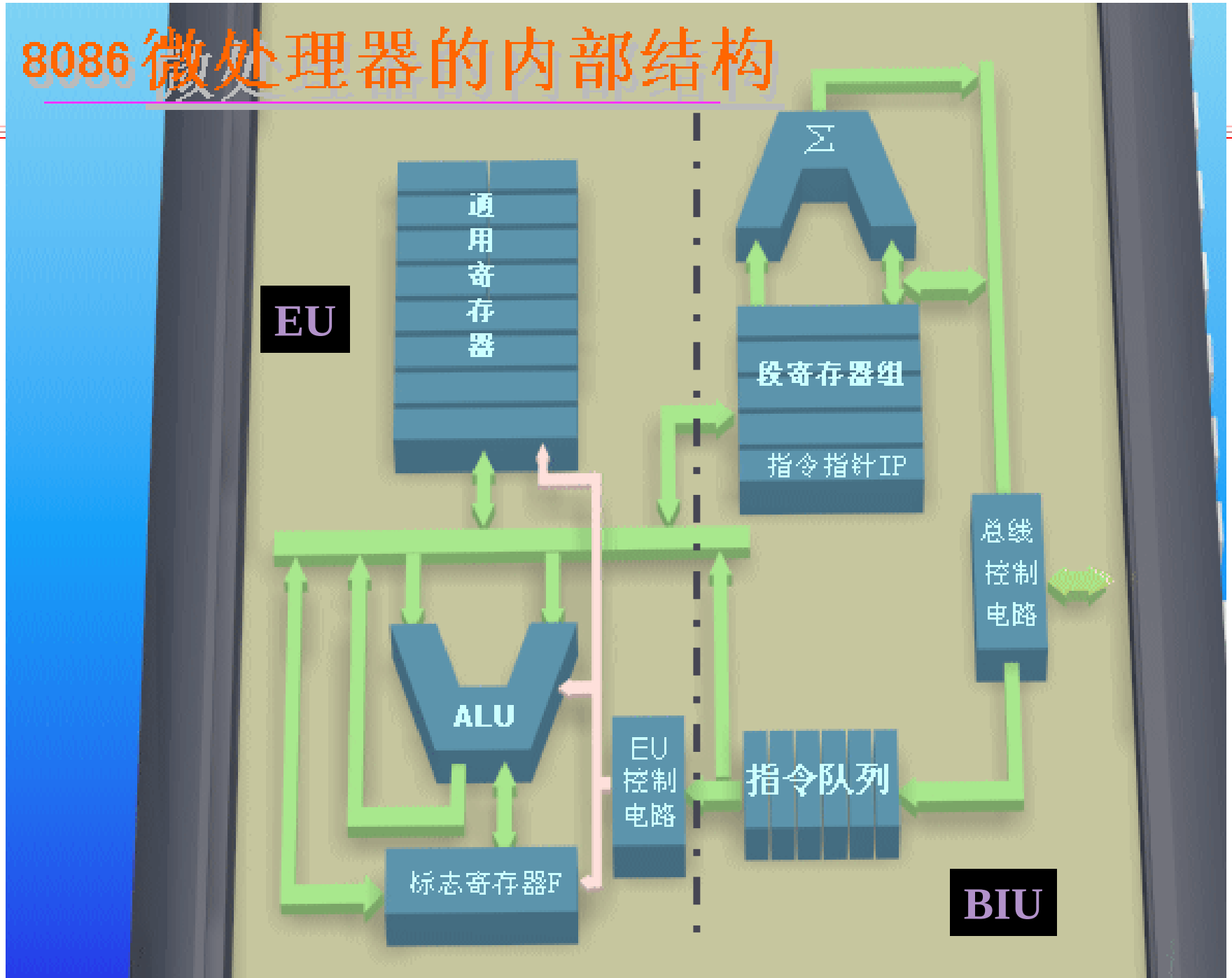
1. **8088的功能结构/编程结构**
2. **8088的寄存器结构**
3. **存储器组织**
4. **标志寄存器**


GND 1
AD14 2
AD13 3
AD12 4
AD11 5
AD10 6
AD9 7
AD8 8
AD7 9
AD6 10
AD5 11
AD4 12
AD3 13
AD2 14
AD1 15
AD0 16
NMI 17
INTR 18
CLK 19
GND 20
8086
CPU
40 VCC
39 AD15
38 A16/S3
37 A17/S4
36 A18/S5
35 A19/S6
34 BHE/S7
33 MN/MX
32 RD
31 HOLD
30 HLDA
29 WR
28 M/IO
27 DT/R
26 DEN
25 ALE
24 INTA
23 TEST
22 READY
21 RESET
RQ/GT0
RQ/GT1
LOCK
S2
S1
S0
QS0
QS1
最小方式 最大方式

工作方式与引脚信号


8284A
READY
READY CLK
+5V
MN/MX
RESET
TEST
HOLD
HLDA
NMI
INTR
INTA
M/IO
W/R
RD
8086
CPU
ALE
BHE
A19 — A16
AD15 — AD0
DT/R
DEN
STB
8282
OE
8286
T
OE
BHE
控制总线
地址总线 A0—A19
数据总线 D0—D15
系统总线
16位CPU

8086微处理器的内部结构



1、8086的功能结构/编程结构

编程结构：是指从程序员和使用者的角度看到的结构。

从功能上划分为总线接口部件(BIU)和执行部件(EU)两部分。

1、 BIU的功能：

预取指令到指令队列。

负责CPU和存储器/外设之间的数据传送。

2、 EU的功能：

负责指令的执行。

一、总线接口部件（BIU—Bus Interface Unit）的结构

- 4个16位的段地址寄存器
  - CS——代码段寄存器
  - DS——数据段寄存器
  - SS——堆栈段寄存器
  - ES——扩展段寄存器
- 16位的指令指针寄存器 IP
- 20位的地址加法器Σ

  段地址*16 （左移4位）+ 偏移量 → 20位的实际物理地址
- 6个字节的指令队列缓冲器： 提高CPU的效率
- 总线控制逻辑

二、执行部件（EU——Execution Unit）的结构

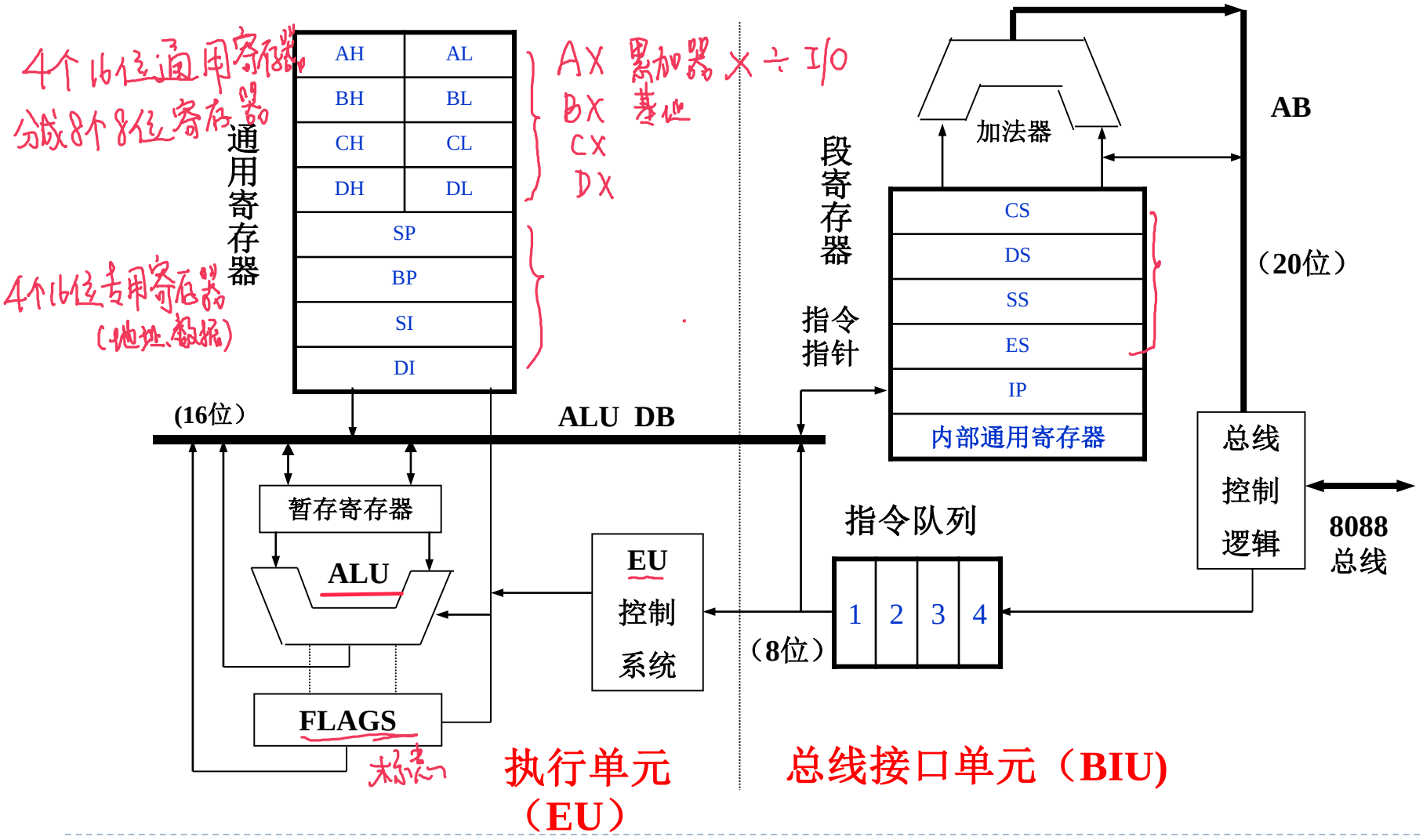
- **16位的算术逻辑单元ALU**
  - 完成算术/逻辑运算和指令要求寻址的单元地址的位移量
- **4个16位的通用寄存器**
  - AX ——累加器
  - BX —— 基址寄存器
  - CX —— 计数器
  - DX —— 数据寄存器

AX,BX,CX,DX，它们又可以分成8个8位的寄存器使用：AH,AL,BH,BL,CH,CL,DH,DL

- **4个16位的专用寄存器**
  - SP —— 堆栈指针寄存器
  - BP —— 基址指针寄存器
  - SI —— 源变址寄存器
  - DI —— 目的变址寄存器

AH	AL
BH	BL
CH	CL
DH	DL
SP	
BP	
SI	
DI	

- **EU控制单元： 译码、产生控制信号。**
- **16位的标志寄存器**


4个16位通用寄存器
分成8个8位寄存器
AX 累加器 × ÷ I/O
BX 基址
CX
DX
4个16位专用寄存器
(地址、数据)
通用寄存器
AH
AL
BH
BL
CH
CL
DH
DL
SP
BP
SI
DI
段寄存器
CS
DS
SS
ES
IP
内部通用寄存器
指令指针
加法器
AB
(20位)
(16位)
ALU DB
暂存寄存器
ALU
FLAGS
标志
EU
控制
系统
指令队列
1
2
3
4
(8位)
总线
控制
逻辑
8088
总线
执行单元
(EU)
总线接口单元(BIU)

8088的寄存器结构

AH	AL	AX 累加器
BH	BL	BX 基址
CH	CL	CX 计数
DH	DL	DX 数据
SP		堆栈指针
BP		基址
SI		源地址
DI		目的地址

IP	指令指针
PSW	状态标志

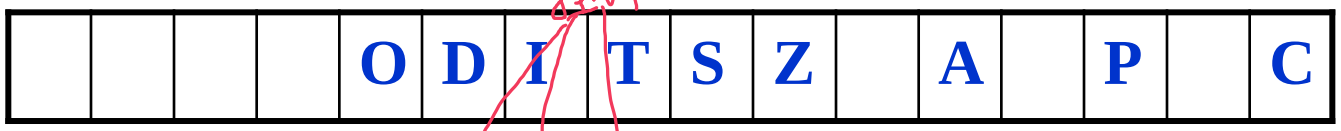
AX 累加器 字乘法，字除法，字I/O
AL 字节乘、除，字节I/O，十进制算术运算、查表
AH 字节乘、除
BX 基数寄存器 查表转换、间接寻址
CX 计数器 串操作，循环计数
CL 变量移位或循环
DX 数据寄存器 字乘法、除法，间接I./O寻址
SP 堆栈指针；BP 基址指针；SI 源变址指针；DI 目的变址指针

8088的寄存器结构

16位标志寄存器

CS
DS
SS
ES

代码段
数据段
堆栈段
附加段



控制标志

无符号数进位即溢出

溢出标志位

方向标志位

地址增量减

中断标志位

中断请求

跟踪标志位

符号标志位

零标志位

辅助进位

奇偶校验

进位

状态标志

二进制补码

对于有符号数,进位不一定就表溢出

-2,-1

无进位也可能溢出

4 1H

4 1H

8088的功能结构

微处理器 8086, 8088结构类似。从程序员和使用角度来看的结构即编程结构从功能上分为两部分：总线接口部分**BIU**（**Bus Interface Unit**），执行部分**EU**（**Execution Unit**）。

由于指令队列的存在，两部分各自执行自己的功能并行工作，这种工作方式与传统的计算机在执行指令时的串行工作相比极大的提高了工作效率。

计算机执行程序时，CPU的工作顺序是：

取指令 ⟶ 执行指令 ⟶ 再取指令 ⟶ 再执行指令．．． ＣＰＵ串行工作。

８０８６ＣＰＵ工作顺序是：取指令，执行指令同时进行。并行工作。

存储器组织

用段来组织逻辑空间

1、总的存储空间为1M字节，每段最长可达 64K字节。

2、各段起始地址能被 16 整除。（低 4 位为 0 ）

3、各段之间可分开、部分或完全重叠、可首尾相接。

4、根据各段的用途将其定义为CS、DS、ES、SS段。并用偏移地址（距段起址的字节距离）表示被访问单元。

常在CS中用 IP 表示偏移量，SS中用 SP、BP，DS中用 BX、SI、DI、数值 。

实际地址的形成

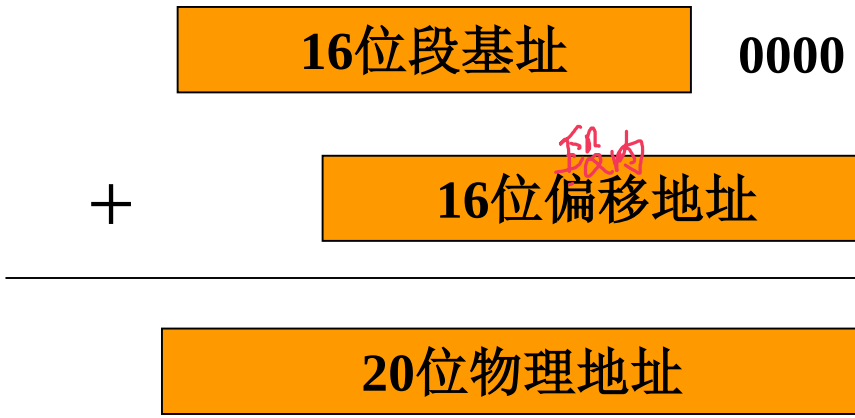
•物理地址：20 位

•逻辑地址：　段基址　（段寄存器的内容）16位

偏移地址（字节距离）16位

一个实际地址可用多个逻辑地址表示。

实际地址的形成（BIU完成）



2000H:1000H表示的物理地址为：

20000H

\+ 1000H

= 21000H

标志寄存器

- **CF**：进位标志位。当执行一个加法（或减法）运算使最高位产生进位（或借位）时，CF为1，否则为0。
- **PF**：奇偶标志位。该标志位反映运算结果中1的个数是偶数个还是奇数个。当指令执行结果的低8位中含偶数个1时，PF为1，否则为0.
- **AF**：辅助进位标志位。当执行一个加法（或减法）运算使结果的低4位向高4位有进位（或借位）时，AF为1，否则为0.
- **ZF**：零标志位。若当前的运算结果为零，ZF为1，否则为0.
- **SF**：符号标志位。他与运算结果的最高位相同。

- **OF**：溢出标志位。当补码运算有溢出时，OF为1，否则为0.
- **DF**：方向标志位。用以指定字符串处理的方向，当DF=1，字符串以递减顺序处理，即地址以从高到低顺序递减。反之则以递增处理。
- **IF**：中断允许标志位。它用来控制8086是否允许接收外部中断请求。若IF=1，8086能响应外部中断。反之则不响应。注意：IF的状态不影响非屏蔽中断请求（NMI）和CPU内部中断请求。
- **TF**：跟踪标志位。为调试程序而设定的陷阱控制位。当TF=1，8086CPU处于单步状态，此时CPU每执行完一条指令就自动产生一次内部中断。当该复位后，CPU恢复正常。

思考题

运算器、控制器、寄存器

- 1、CPU在内部结构上由哪几部分组成?
- 2、8088/8086 20位物理地址是怎样形成的？当CS=2000H，IP=0100H，其指向的物理地址等于多少？ 20000 0100 20100
- 3、8088/8086微处理器有哪些寄存器？寄存器中哪些可以做地址指针使用？ 通用 专用 段地址 指针 偏移
- 4、有两个16位字1EE5H和2A3CH分别存放在8086微机的存储器的000B0H和000B3H单元中，请用图表示出它们在存储器里的存放情况。