

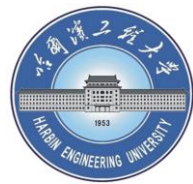
第 2 章 8086 微处理器





第 2 章 8086微处理器

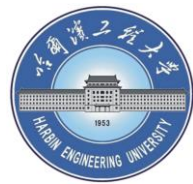
- **2-1 8086 CPU结构**
- **2-2 8086 CPU的引脚及其功能**
- **2-3 8086 存储器组织**
- **2-4 8086 系统配置**
- **2-5 8086 CPU时序**



学习目的

通过对本章的学习，应该能够达到下列要求：

- 掌握8086的编程结构
- 说明8086CPU的工作特点
- 了解8086CPU的引线
- 分析8086CPU基本总线周期时序



学习目的

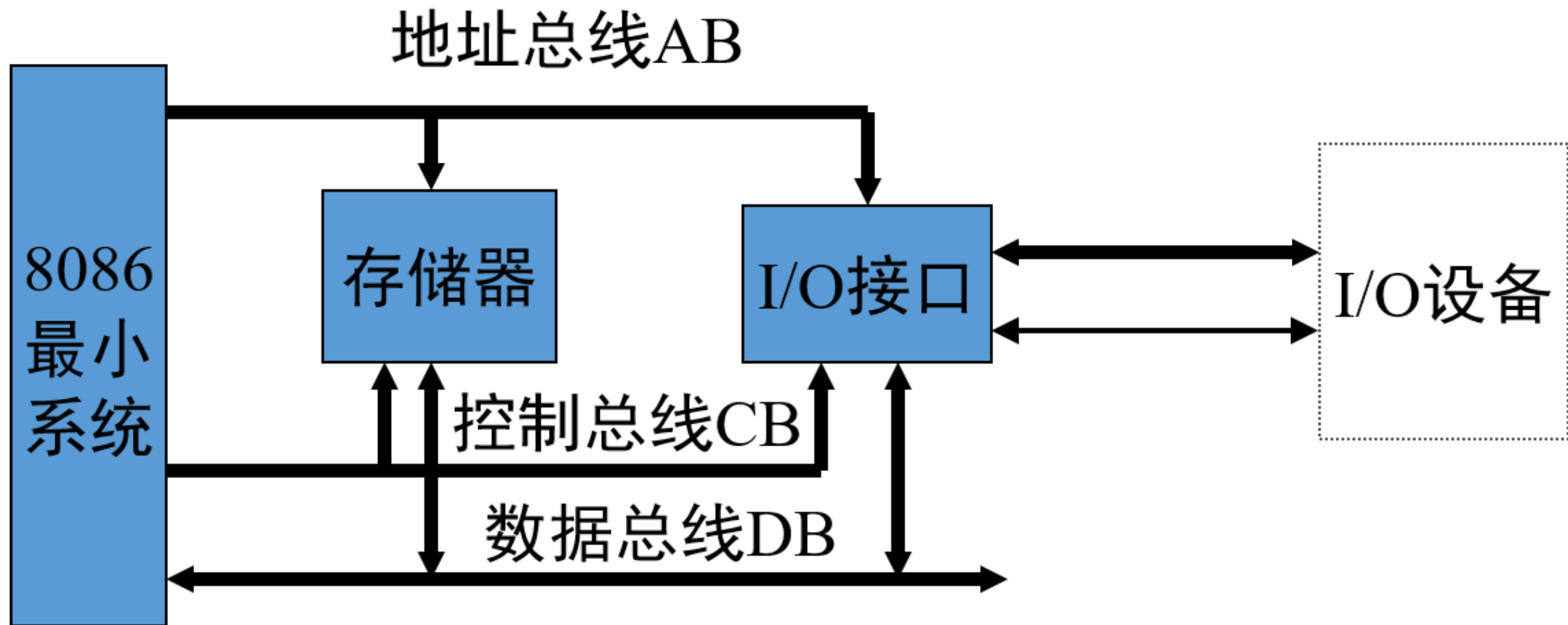
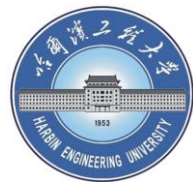
重点

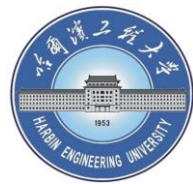
- 8086CPU的编程结构，总线接口单元和执行单元的动作管理
- 8086CPU的引线
- 8086CPU基本总线周期各个T状态的作用
- 存储器与I/O接口的读写时序

难点

- 8086的寄存器结构以及时序

单总线微型计算机结构

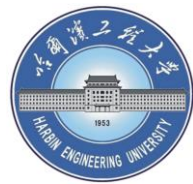




2-1 8086 CPU结构

8086内部寄存器，为16位。

- 1、8086有**16位数据线**，与地址线 A_0-A_{15} 兼用；**20位地址线**，寻址空间达到**1MB**。
- 2、8088有**8位数据线**，与地址线 A_0-A_7 兼用；引脚28、34信号功能与8086稍有不同。



8086的内部结构

8086CPU内部结构分成两部分:

- **总线接口部件BIU:**

总线接口单元BIU，负责控制CPU与**存储器**和**I/O接口**读写。

- **执行部件EU:**

执行单元EU从指令队列中取出指令并执行。

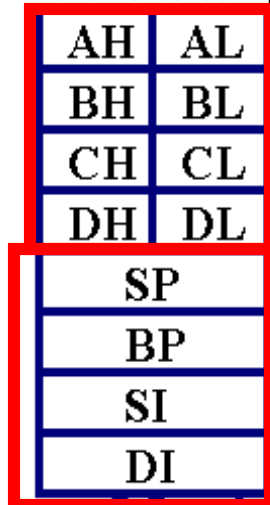
- **特点:**

两级流水线。取指部分和执行指令部分分开进行，提高了速度。



20位

通用寄存器



指针和变址寄存器

SP: 堆栈指针, 其内容与堆栈段寄存器SS的内容一起, 提供堆栈操作地址。

BP: 基址指针: 构成段内偏移地址的一部分。

SI: (Source Index) : SI含有源地址意思, 产生有效地址或实际地址的偏移量。

含有目的意思, 上的偏移量。

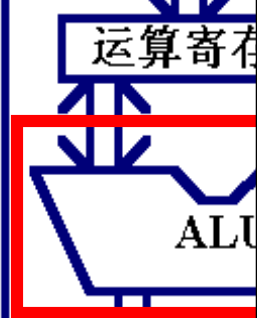
算术逻辑单元ALU:

标志寄存器:

16位字利用了9位。标志分两类:

状态标志 (6位): 反映刚刚完成的操作结果情况。

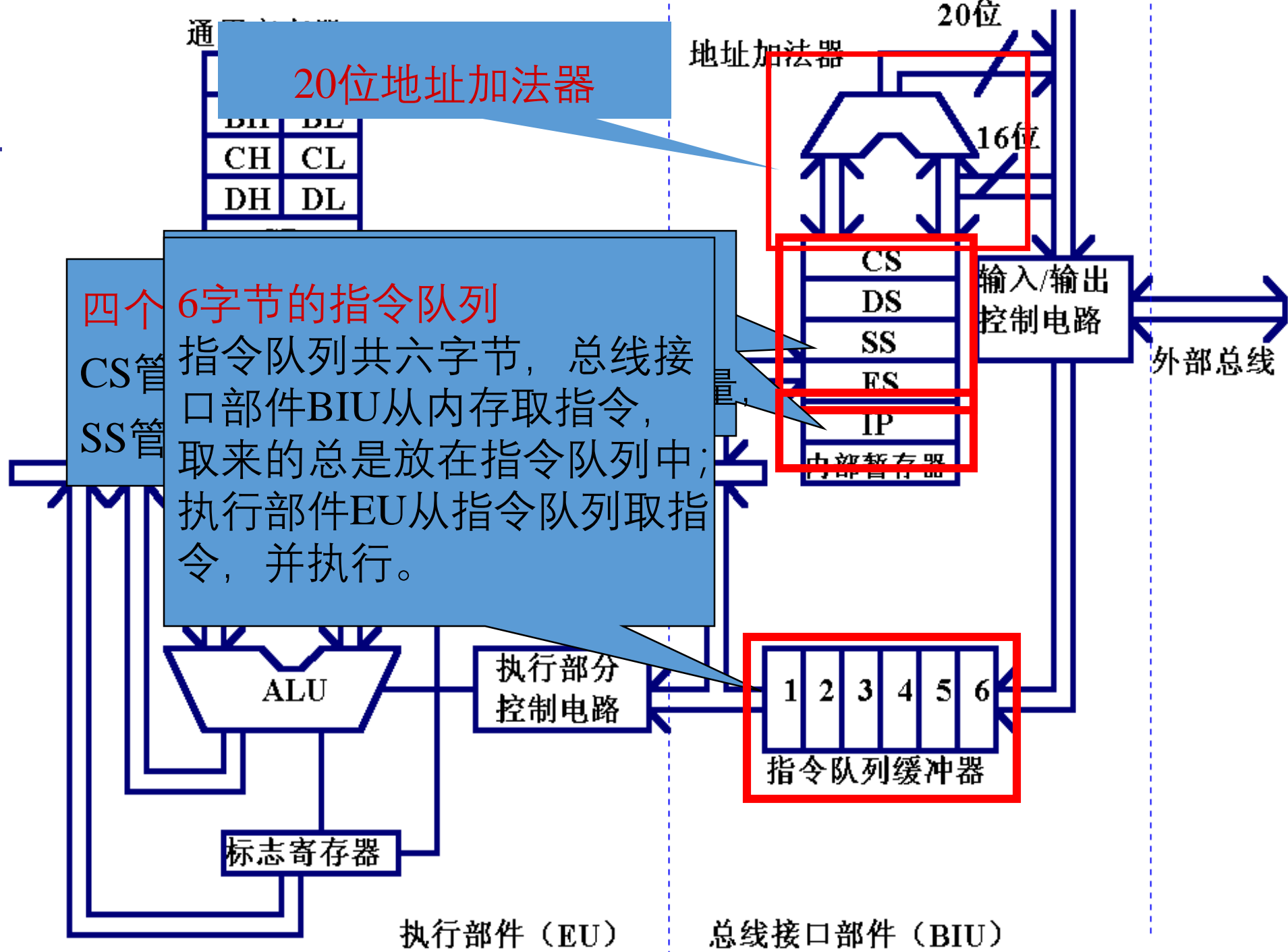
控制标志 (3位): 在某些指令操作中起控制作用。



标志寄存器

执行部件 (EU)

总线接口部件 (BIU)



20位地址加法器

四个6字节的指令队列
指令队列共六字节，总线接口部件BIU从内存取指令，取来的总是放在指令队列中；执行部件EU从指令队列取指令，并执行。

执行部件 (EU)

总线接口部件 (BIU)



总线接口部件BIU

功能：控制**存储器或IO接口**与**CPU**的信息传送，地址形成、取指令、指令排队、读/写操作、总线控制。**16位双向数据总线**和**20位地址总线**，以及**读写与总线控制信号**。

主要包括5个部分：

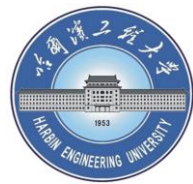
(1) 四个段寄存器：(CS. DS. ES. SS)

CS: 16位的代码段寄存器，代码段首地址高16位；

DS: 16位的数据段寄存器，数据段首地址高16位；

ES: 16位的扩展段寄存器，扩展段首地址高16位；

SS: 16位的堆栈段寄存器，堆栈段首地址高16位。



总线接口部件BIU

(2) 16位的指令指针寄存器IP

IP中的内容是下一条指令对现行代码段基地址的偏移量，与段寄存器CS的内容一起形成取指令的物理地址。

(3) 20位物理地址加法器

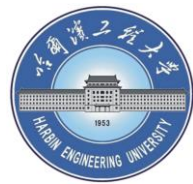
将16位的逻辑地址变换成读/写存储器、外设所需要的20位物理地址。

(4) 6字节的指令队列

预存6字节的指令代码，等待EU处理。

(5) 总线控制逻辑

发出读/写控制信号以及总线控制信号。

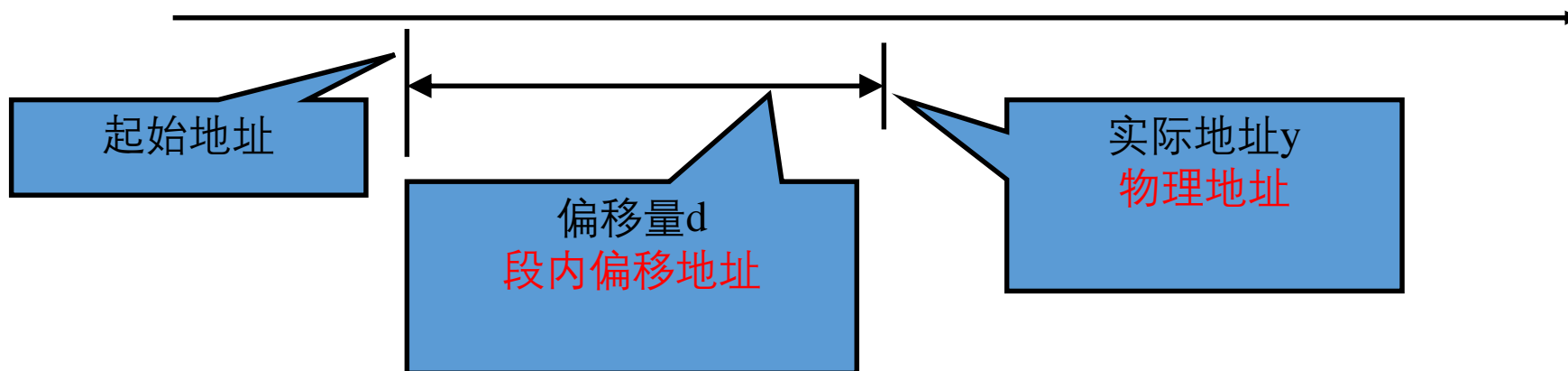


8086的内部结构

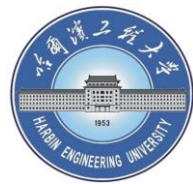
- 怎样用16位寄存器表示20位地址信息？
- 存储器是以**字节 (BYTE)** 为单位存储信息，存储器的通用单位为**字节**。
- 由于8086访问内存要**20位地址(1MB)**，而执行部件EU中所有寄存器和内部数据总线均为**16位**，只能提供**16位地址**，只能寻址 $2^{16} = 64\text{KB}$ 。因此，把1MB存储空间分为若干个“逻辑段” (Segment)，每个段最多可管理**64KB**空间。
- 一个段的最小长度是**16字节**。
- 段与段之间可以是分开的、连续的、重叠的甚至重合。任何一个存储单元都可以定义在一个段或多个段中。

8086的内部结构

- 一个存储单元的**实际地址y**用它所在**段的起始地址**以及**相对于段起始地址的16位偏移量d**表示， $y=x+d$ 。改变偏移量可在**段内64kB范围移动**。改变起始地址x可以使整个段在**1MB空间浮动**。



- **段起始地址x**是**20位二进制数**，规定它的**最低4位二进制数规定等于0**，这样起始地址只有高16位二进制数需要指定，将其存放在**16位段寄存器**中，称为**段基址**。把段基址**左移4个二进制位**，相当于乘以十进制数16，即得到**段起始地址**。



8086的内部结构

存储单元实际20位地址称**物理地址**。用**段基址**和**偏移地址**表示的地址信息，称为**逻辑地址**，写为**段基址 : 偏移地址**，编程时采用**逻辑地址**。

物理地址形成算法：段寄存器 × 16 + 偏移地址

段寄存器提供高16位，而低4位补0，二进制0，得起始地址

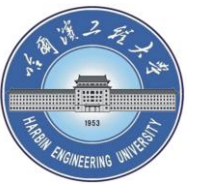
16位段基址 0000
+ 16位偏移地址

地址加法器实现，无符号或有符号？

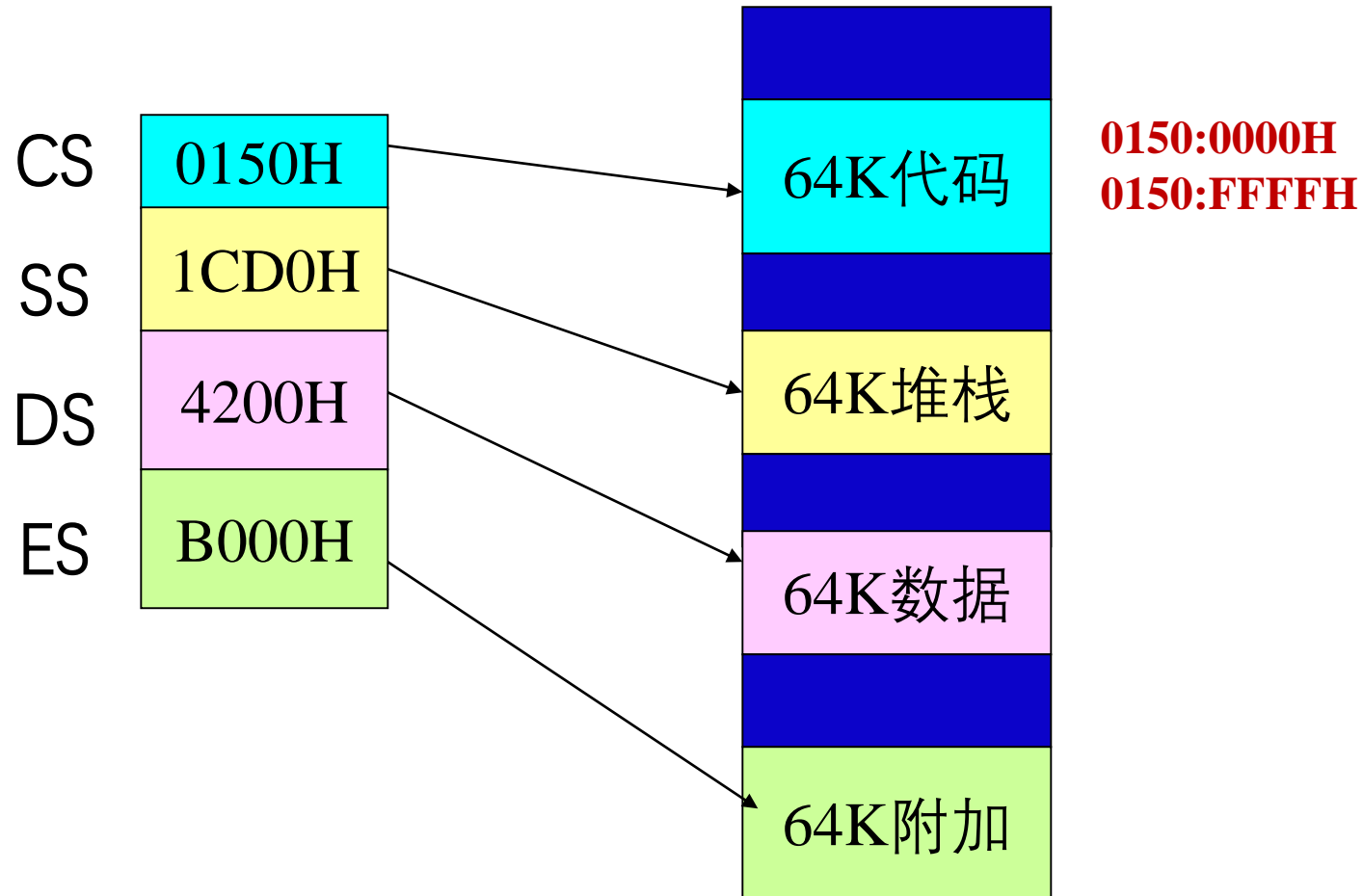
20位物理地址

给定逻辑地址，可唯一地确定一个物理地址；但给定物理地址，存在无数个逻辑地址！！

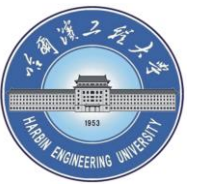
8086的内部结构



段的分配:



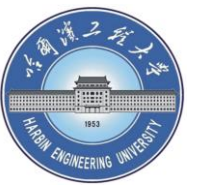
例题



例：逻辑地址1000H : 501AH对应的物理地址

$$\begin{array}{r} 10000 \\ + \underline{501A} \\ \hline 1501A \end{array}$$

例题



例2、CS=2000H，最大寻址空间可达多少？物理地址范围是多少？

CS左移四位：2000 0H

IP的内容：0000~FFFF H

$$\begin{array}{r} 20000 \\ + \quad 0000 \\ \hline 20000 \end{array}$$

$$\begin{array}{r} 20000 \\ + \quad \text{FFFF} \\ \hline 2\text{FFFF} \end{array}$$

物理地址：20000H~2FFFFH

64K

逻辑地址的来源

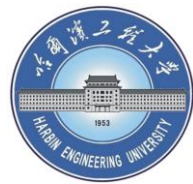
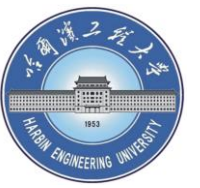


表2-1 逻辑地址来源

操作类型	隐含段地址	替换段地址	偏移地址
取指令	CS	无	IP
堆栈操作	SS	无	SP
BP间接寻址	SS	CS、DS、ES	有效地址
存取变量	DS	CS、ES、SS	有效地址
源字符串	DS	CS、ES、SS	SI
目标字符串	ES	无	DI

指令队列操作

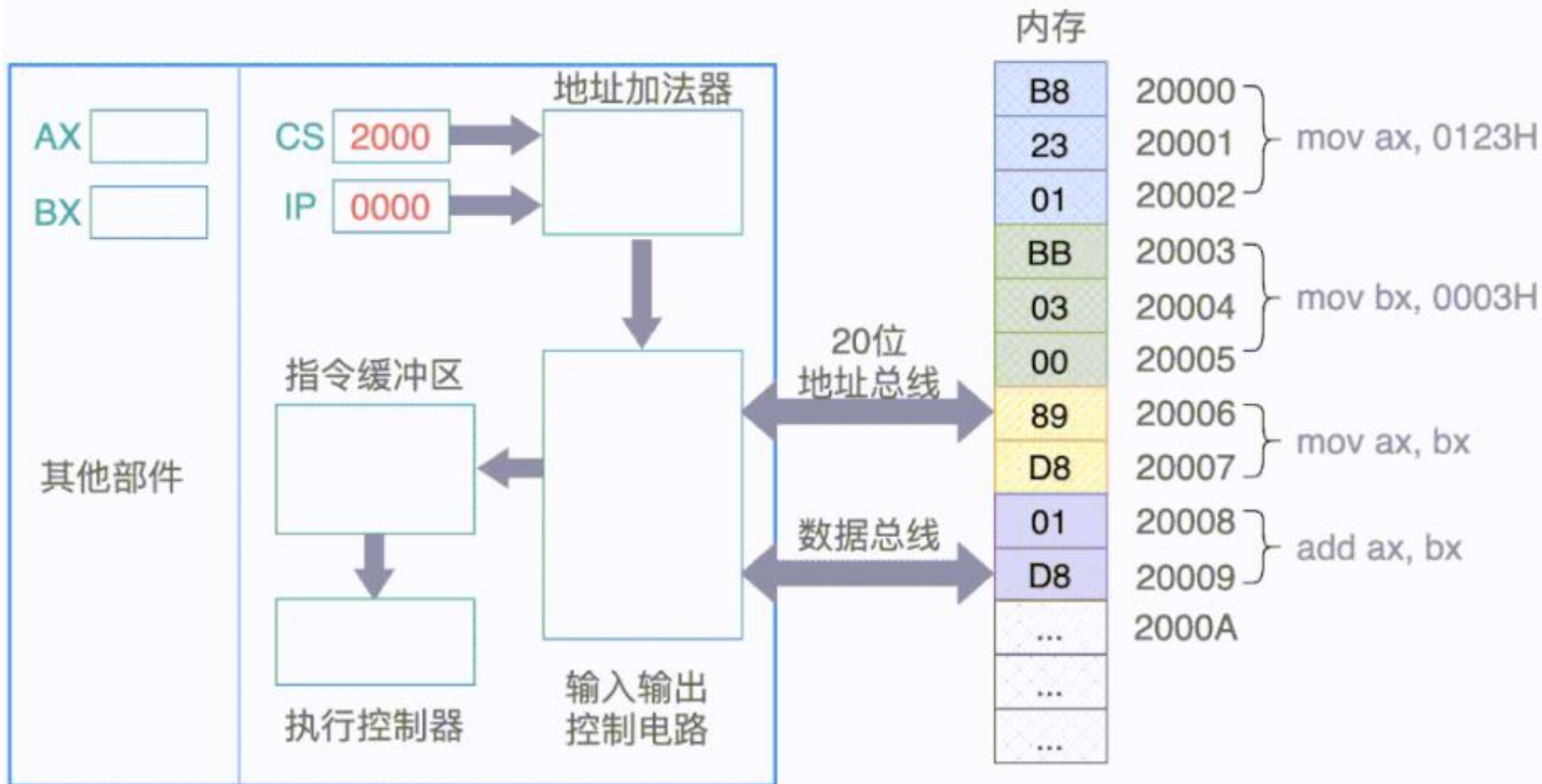


- **总线接口部件BIU**根据当前的**CS**和**IP**的值，形成20位物理地址，从其指定的存储单元取指令代码，放在指令队列中
- 指令**指针IP**的值由**BIU**自动修改，始终等于**下一条指令**在现行代码段的偏移地址；
- 执行部件**EU**从指令队列取指令码，进行指令译码和执行指令，**取指令与执行指令可同时进行**；
- 当指令队列中有**2个或2个字节以上空余**时，BIU自动将指令取到队列中。
- 指令队列已满，BIU与EU又无总线请求时，总线接口部件进入**空闲状态**
- 执行**转移指令、调用指令、返回指令**时，**BIU**自动清除指令队列，然后从新地址取指令，并立即送给EU，然后再从新单元开始，从新填满队列

指令队列操作



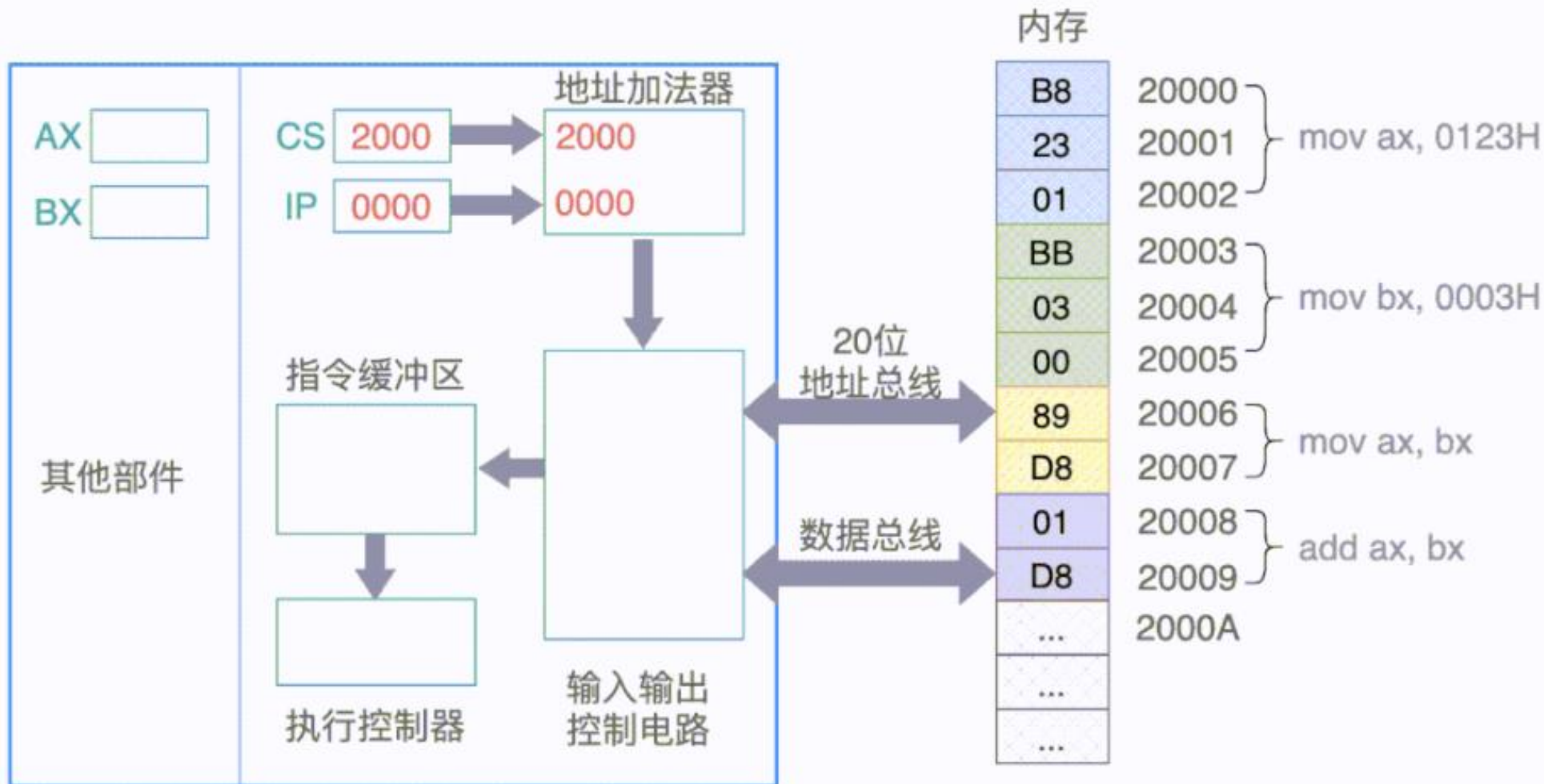
初始状态: CS:IP = 2000: 0000



指令队列操作

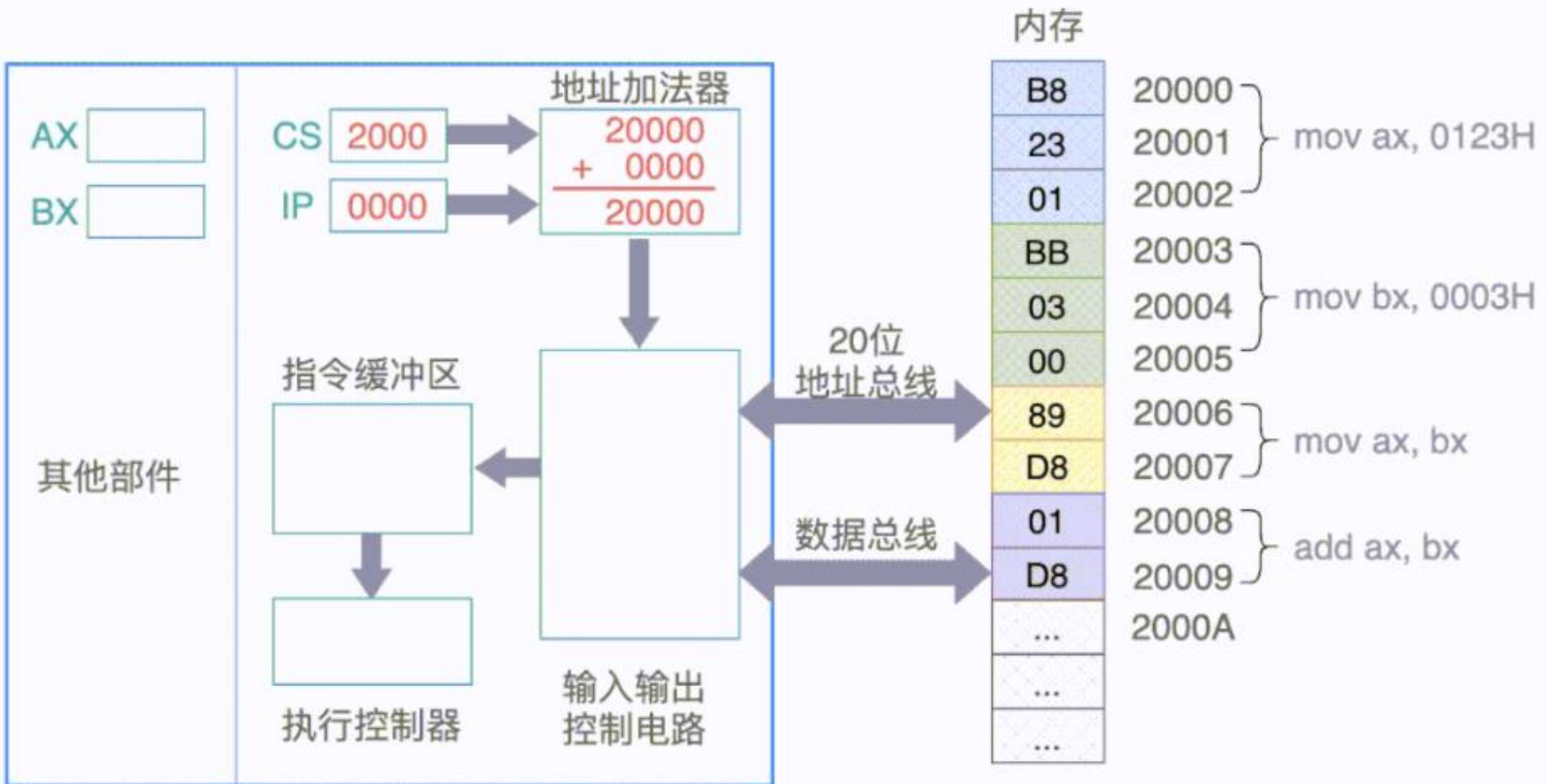


CS:IP 中的内容, 送入地址加法器



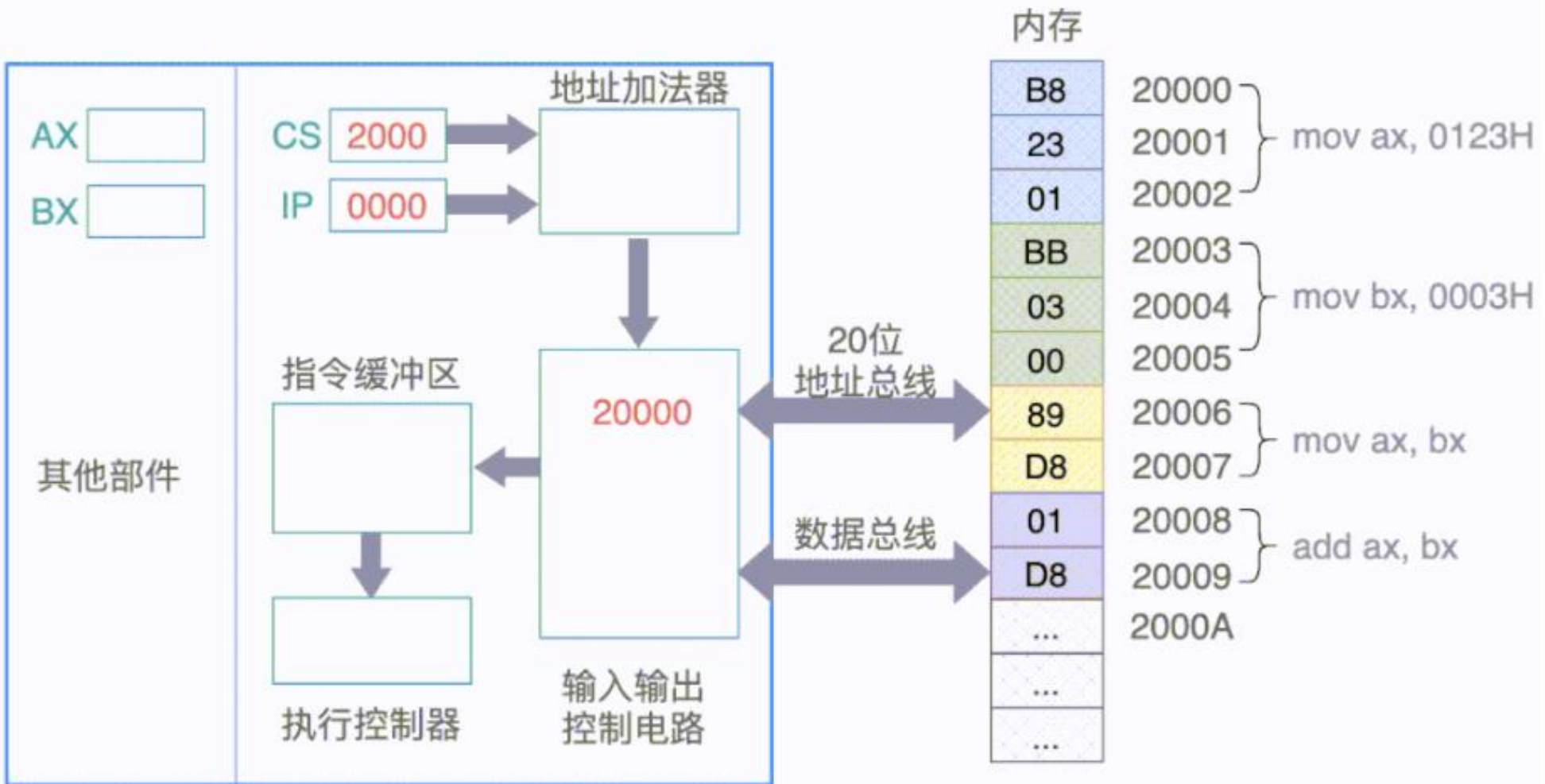
指令队列操作

计算物理地址：段地址 x 16 + 偏移地址

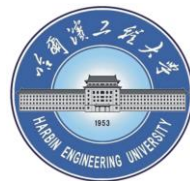


指令队列操作

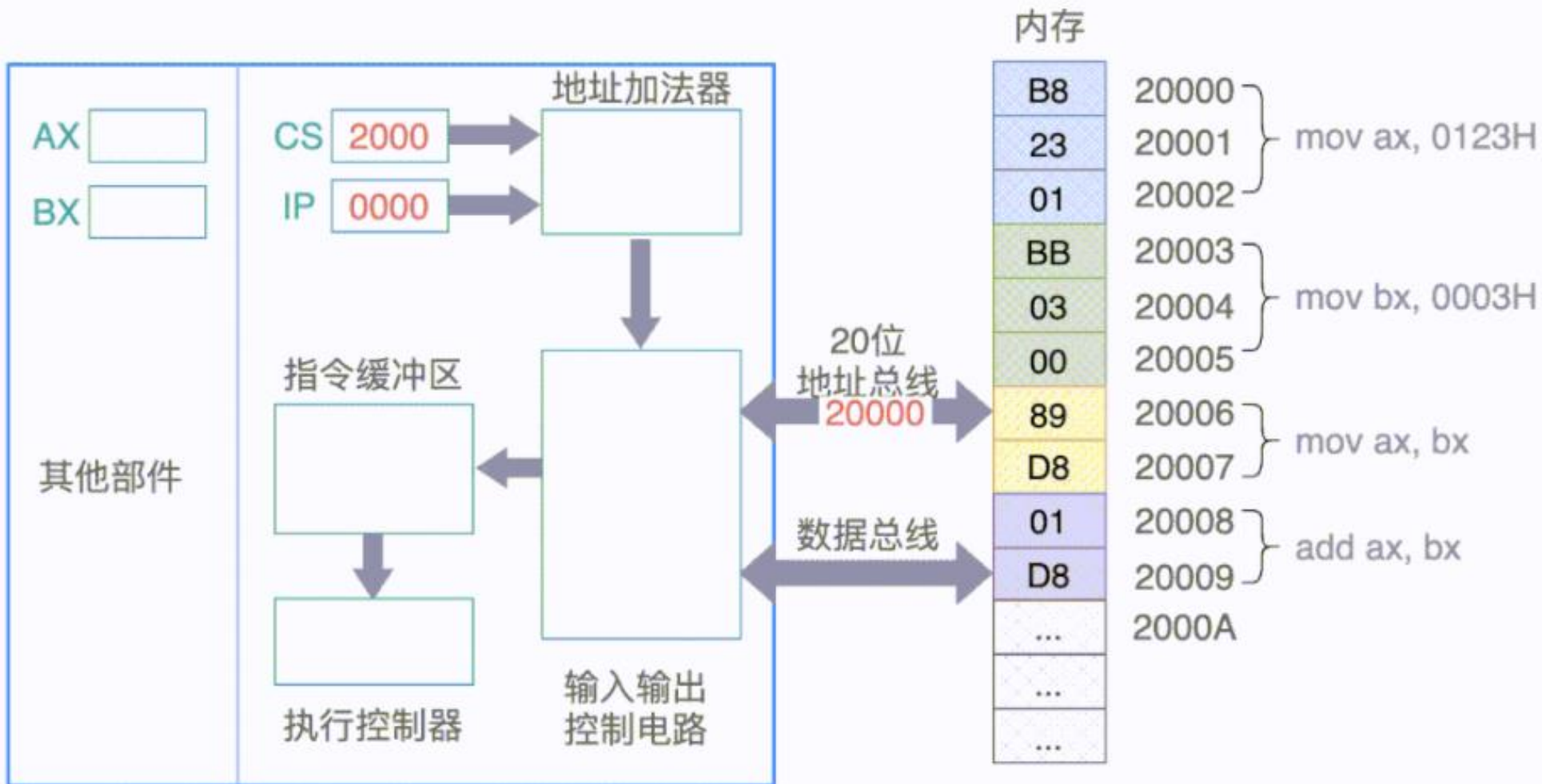
地址加法器：把物理地址送到输入输出控制电路



指令队列操作

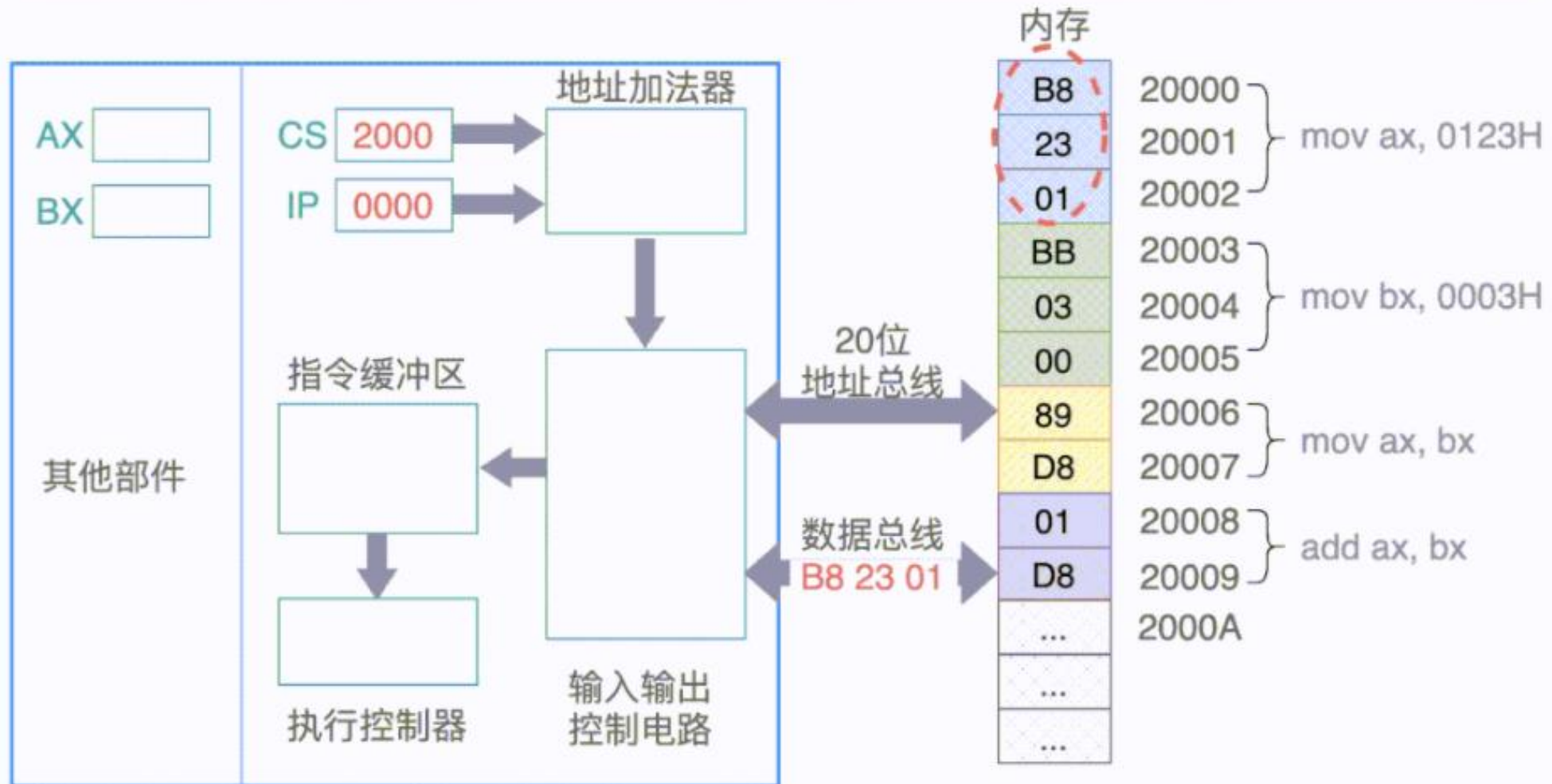


输入输出控制电路: 把物理地址送到地址总线



指令队列操作

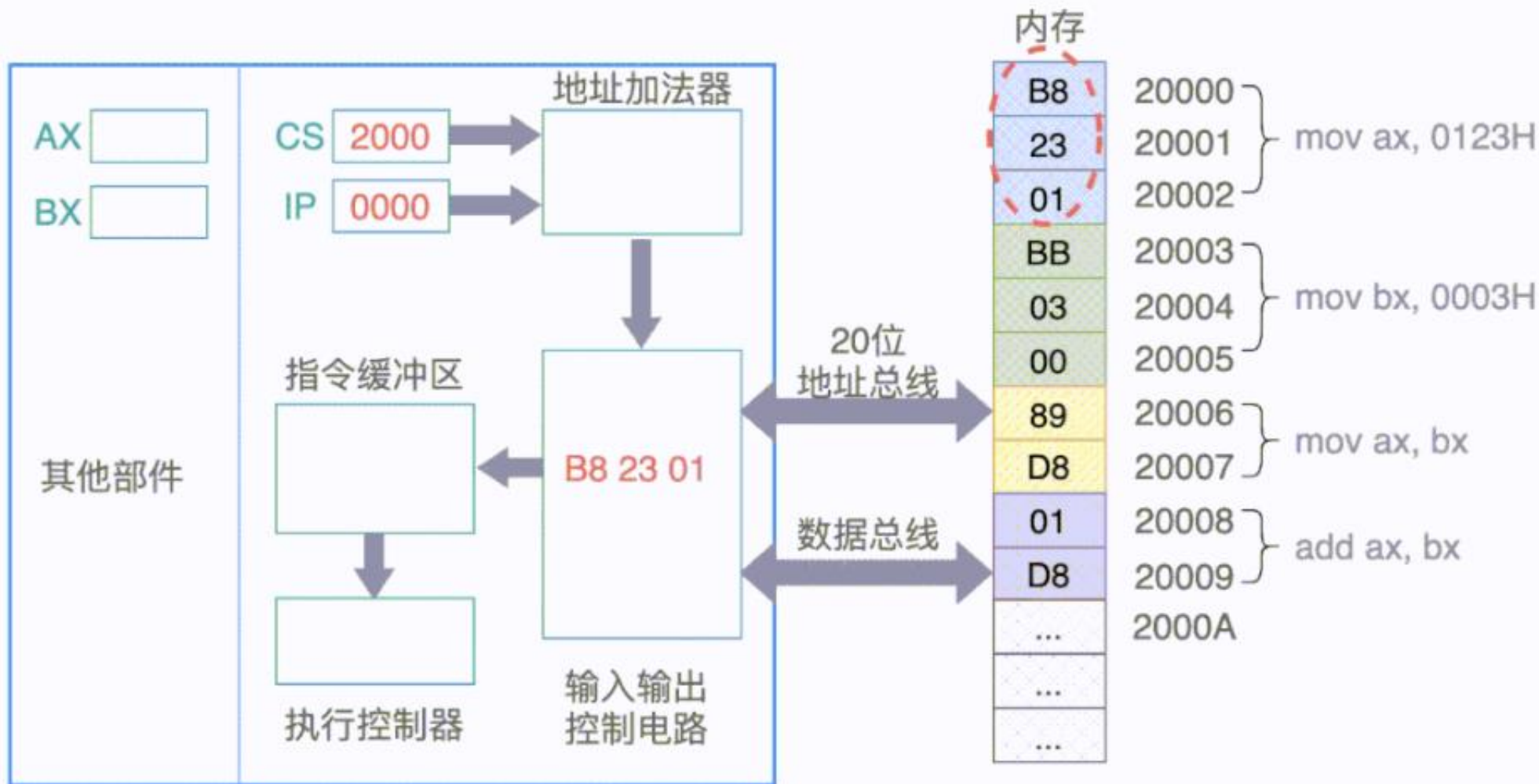
内存单元 20000 处的指令，经过数据总线被送到 CPU 的指令缓冲区



指令队列操作



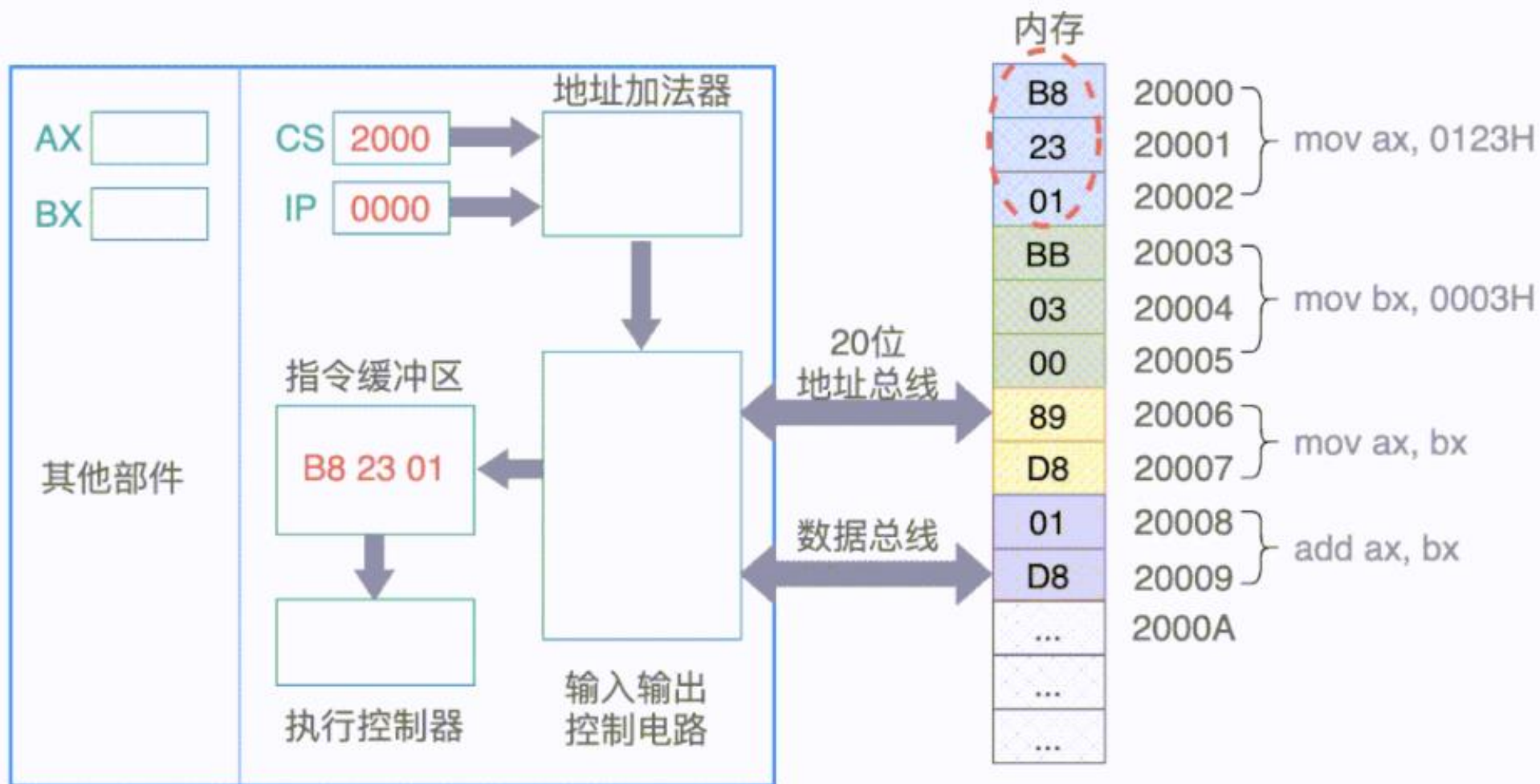
内存单元 20000 处的指令，经过数据总线被送到 CPU 的指令缓冲区



指令队列操作



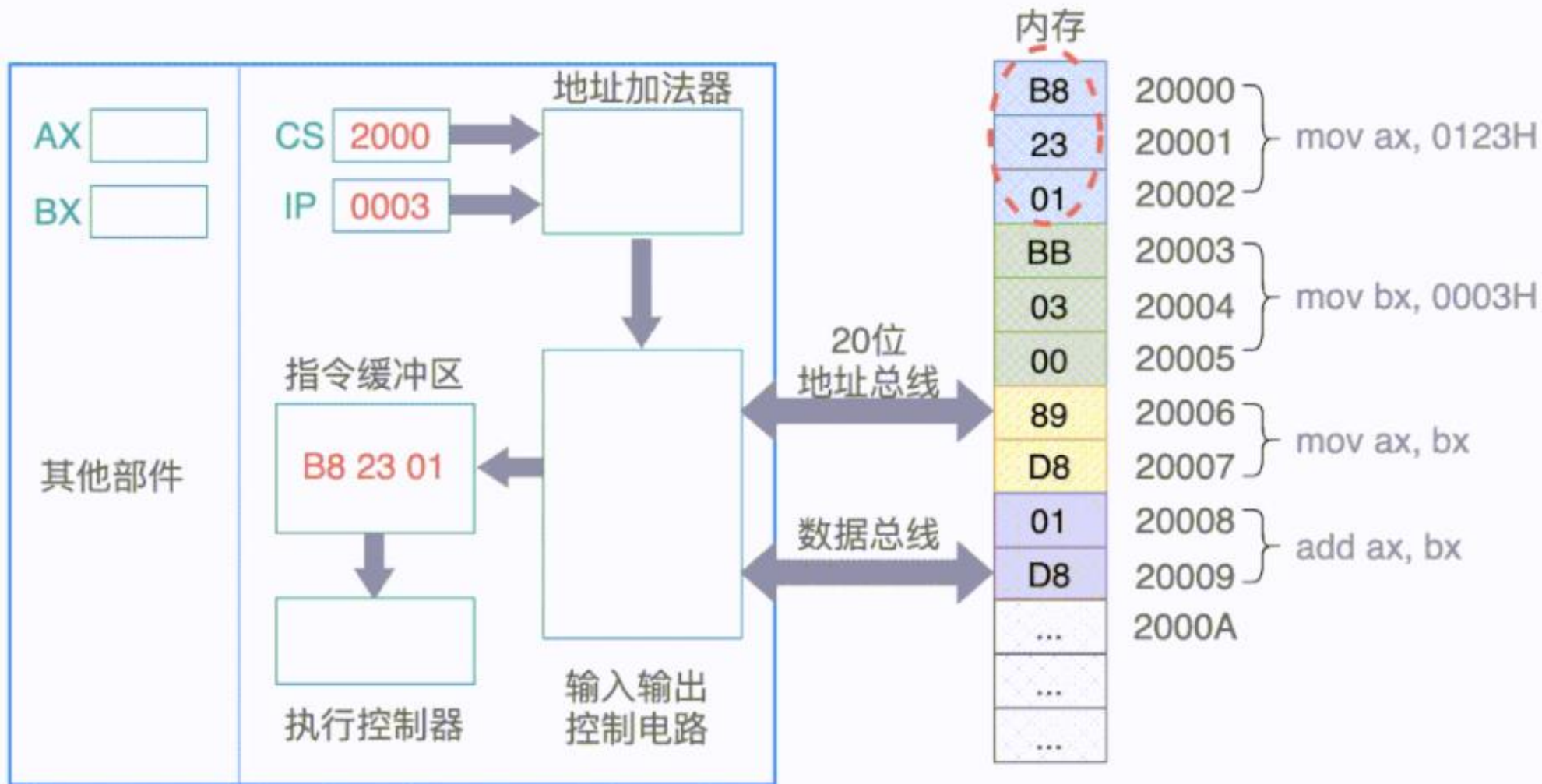
内存单元 20000 处的指令，经过数据总线被送到 CPU 的指令缓冲区



指令队列操作



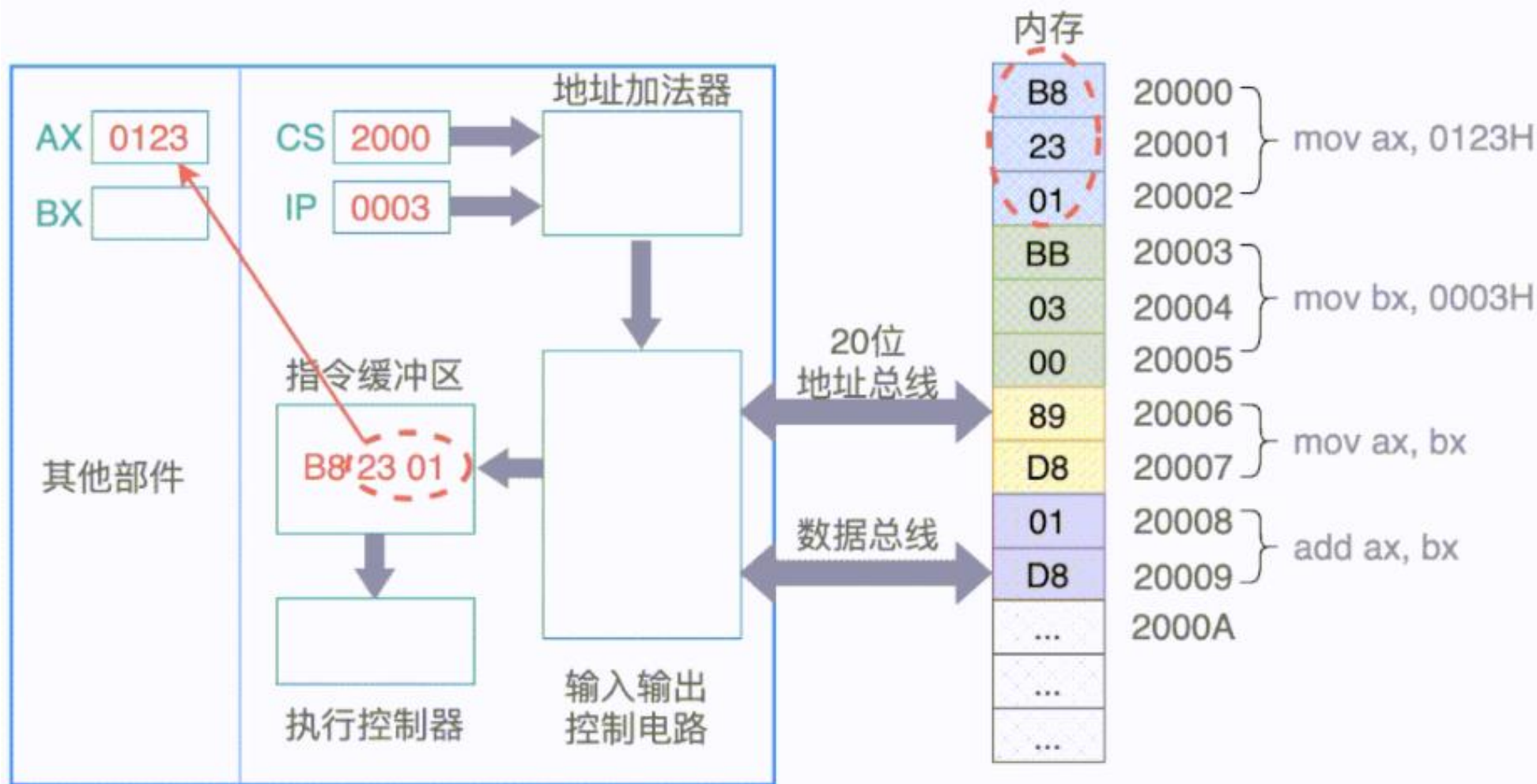
IP 寄存器自动增加，指向下一个指令的偏移地址 2000:0003



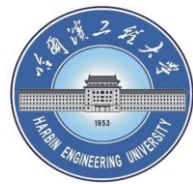
指令队列操作



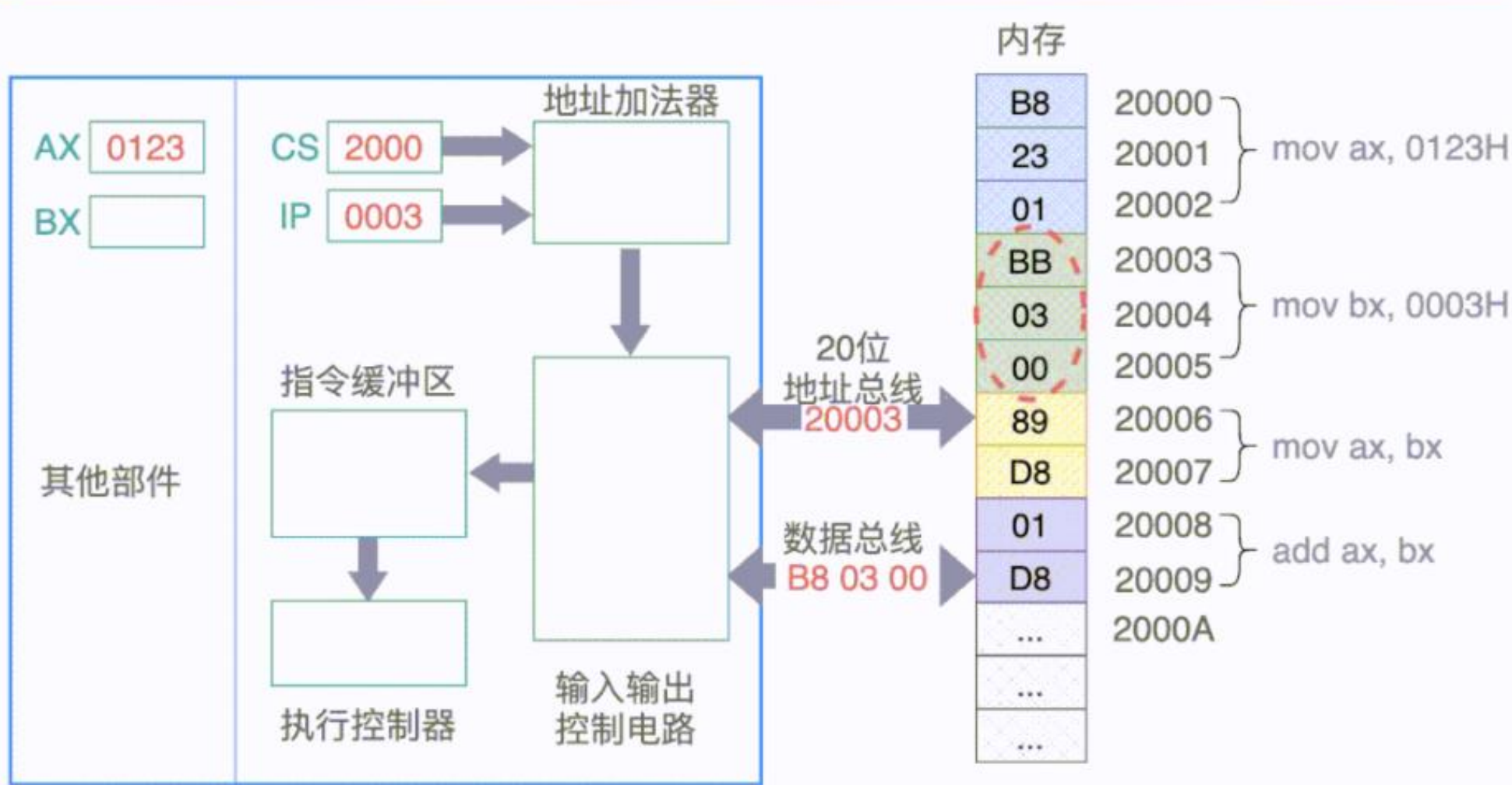
执行当前的指令：把 0123 放入 AX 寄存器中



指令队列操作



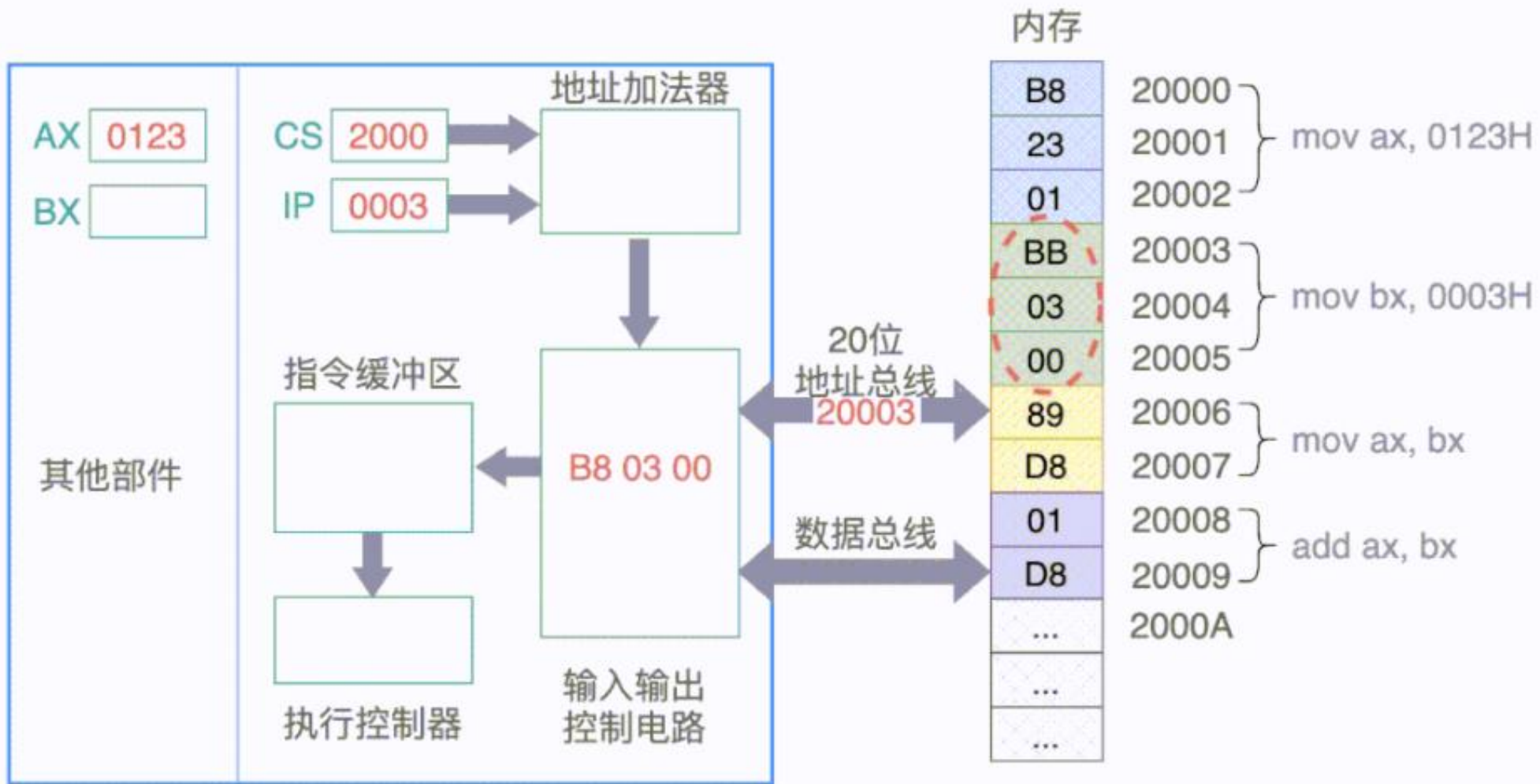
内存单元 20003 处的指令，经过数据总线送到 CPU 的指令缓冲区



指令队列操作



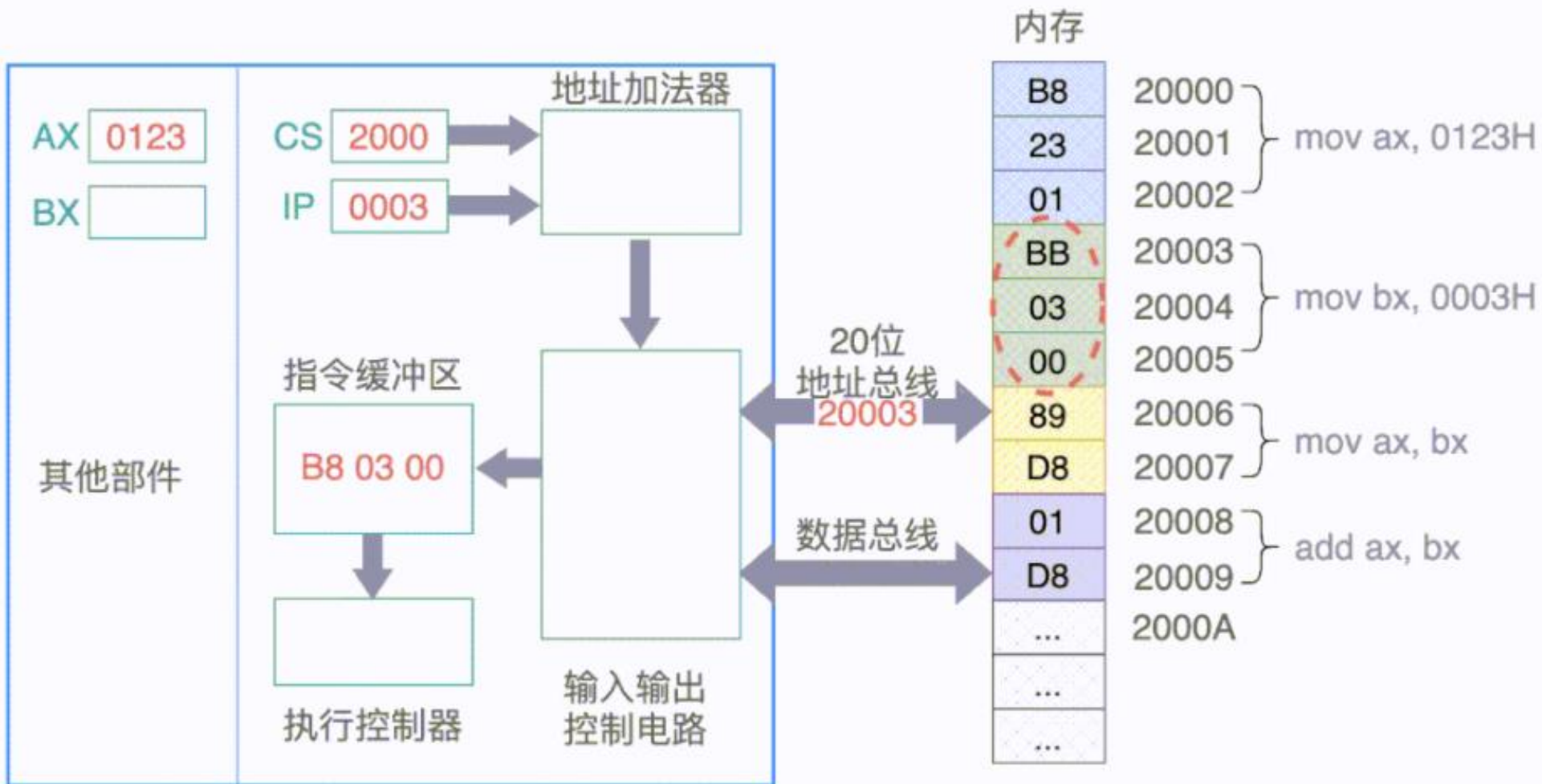
内存单元 20003 处的指令，经过数据总线送到 CPU 的指令缓冲区



指令队列操作



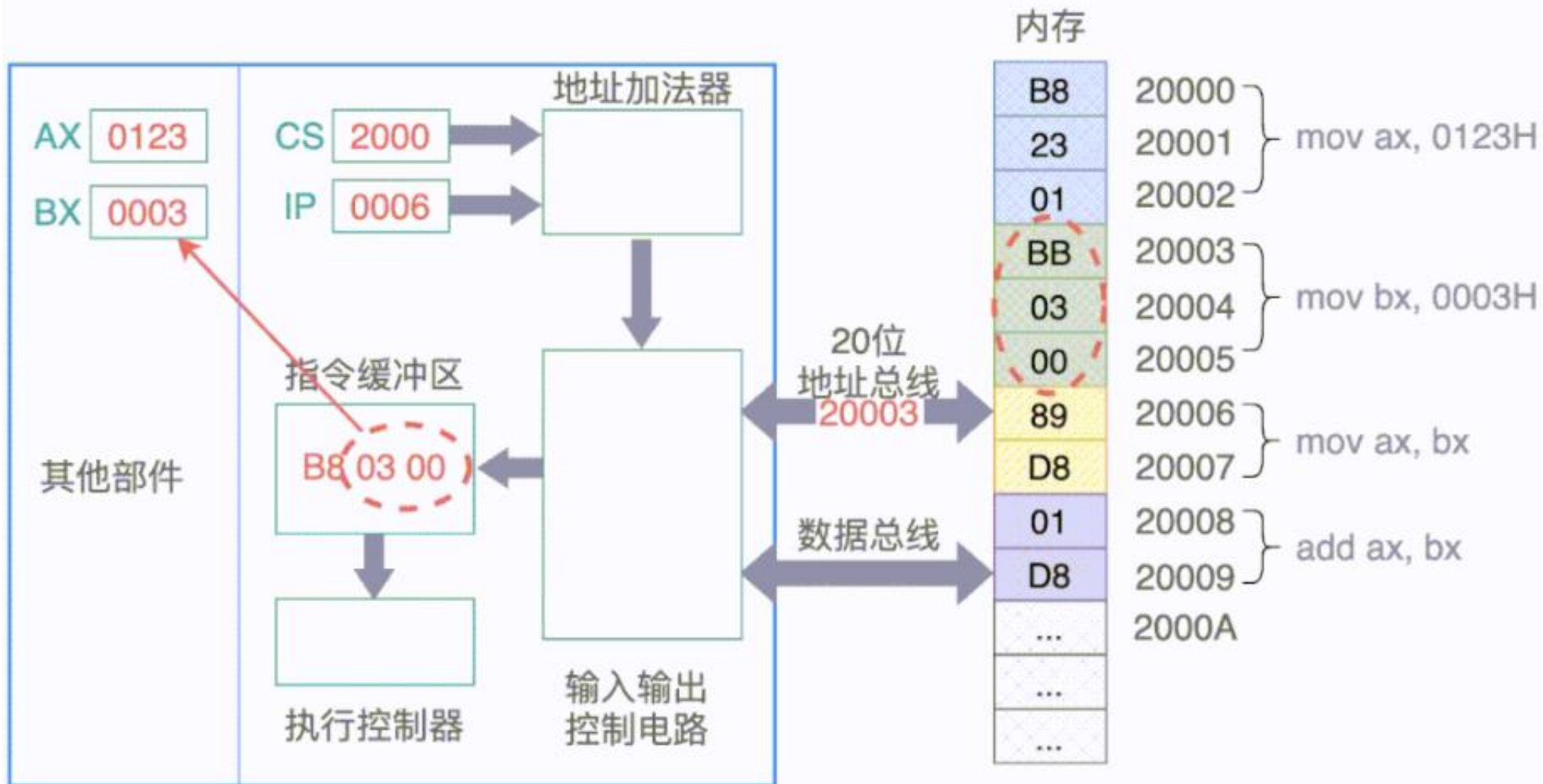
内存单元 20003 处的指令，经过数据总线送到 CPU 的指令缓冲区



指令队列操作



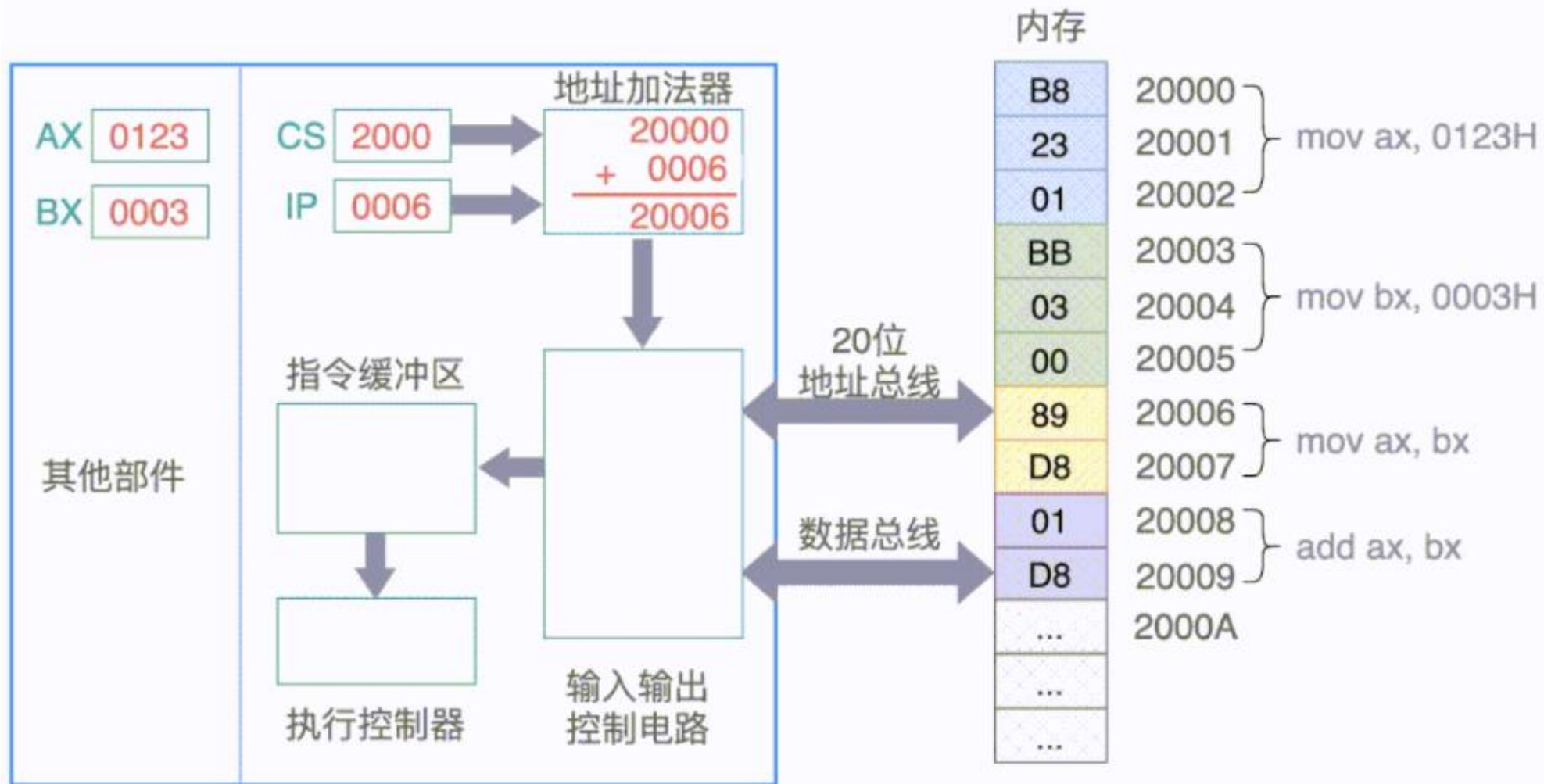
执行当前指令：把 0003 放入寄存器 BX



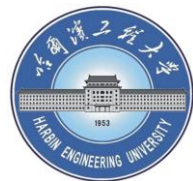
指令队列操作



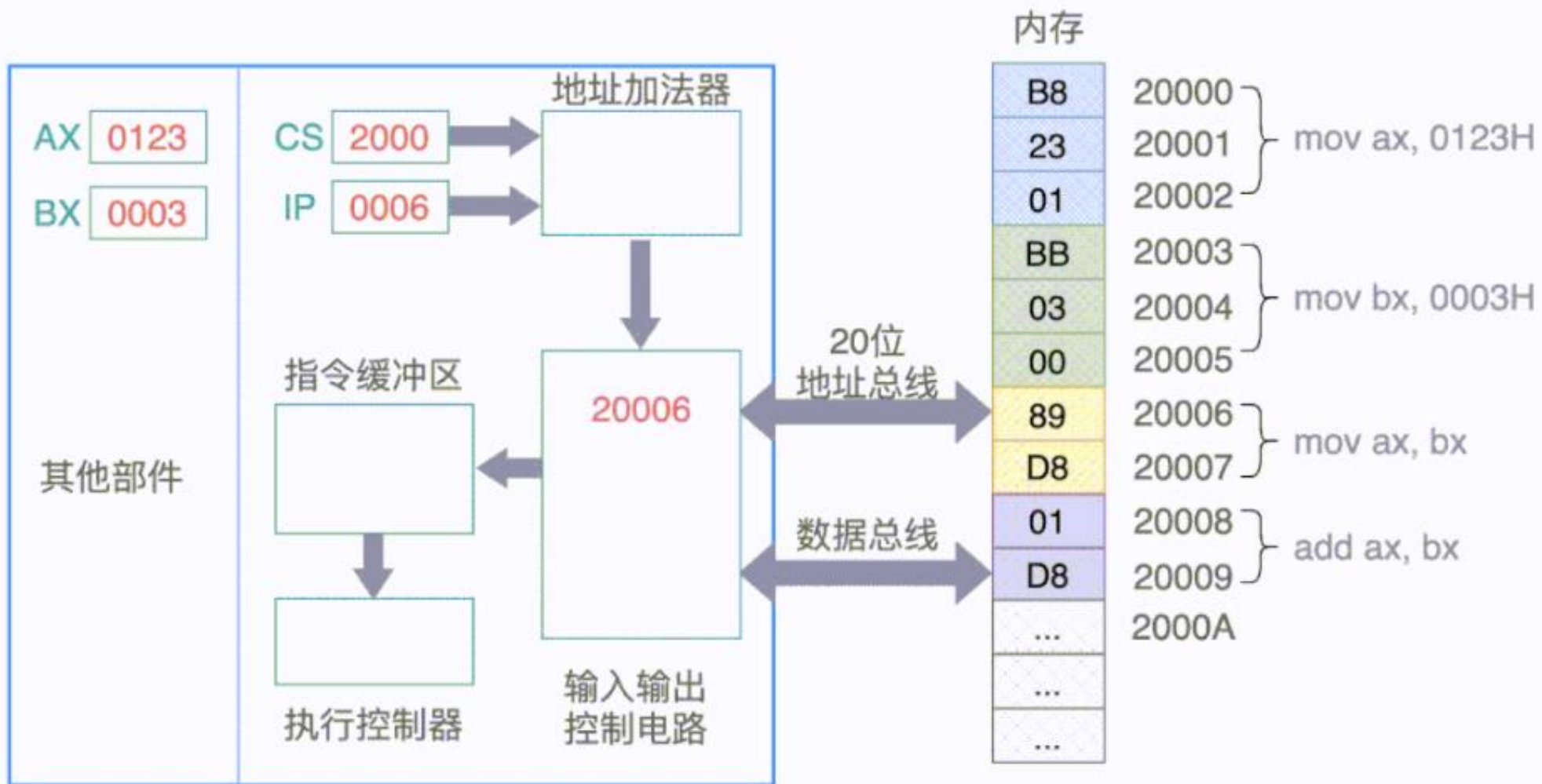
继续指令指令 89 D8 (mov ax, bx)



指令队列操作



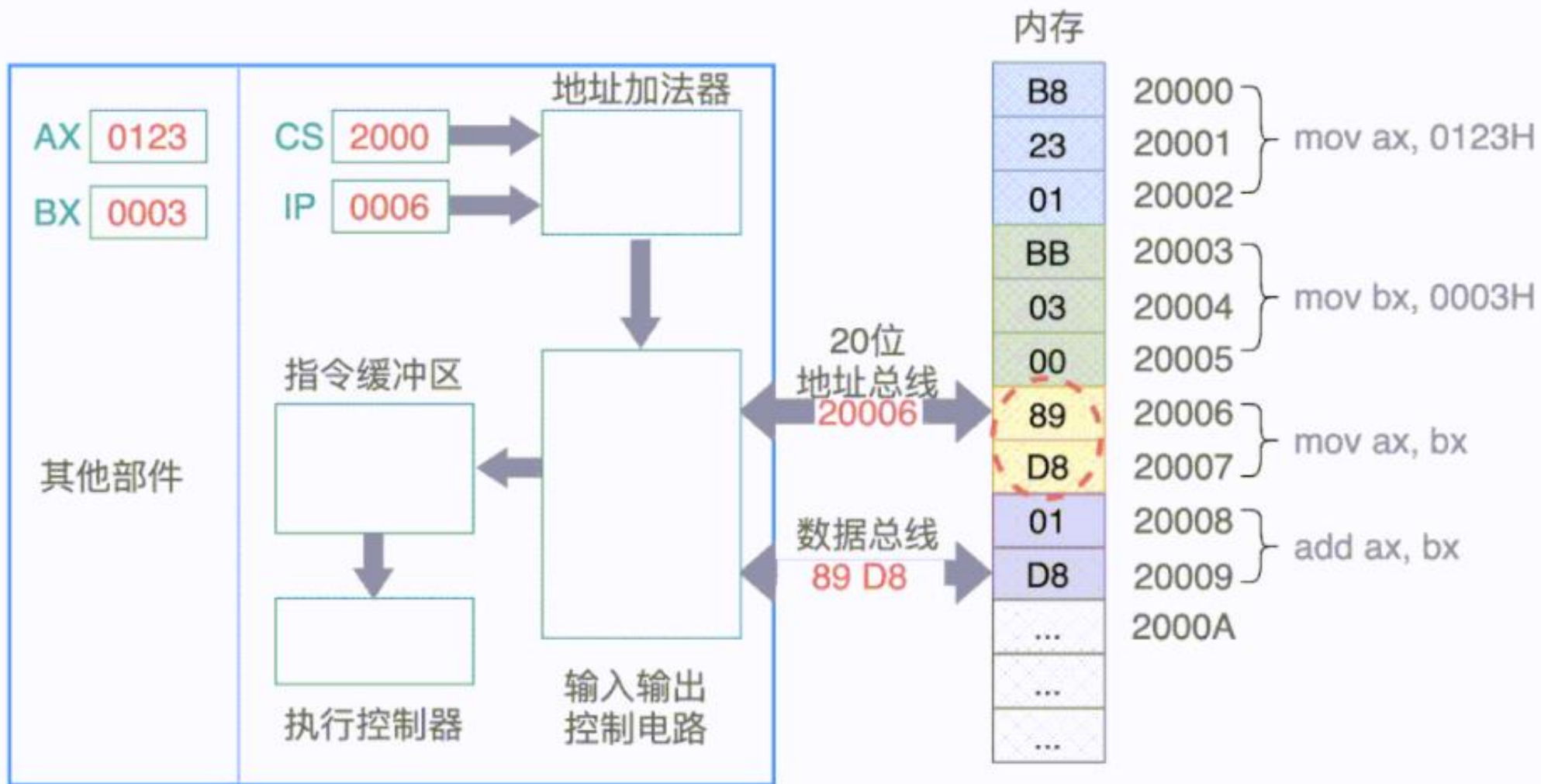
继续指令指令 89 D8 (mov ax, bx)



指令队列操作



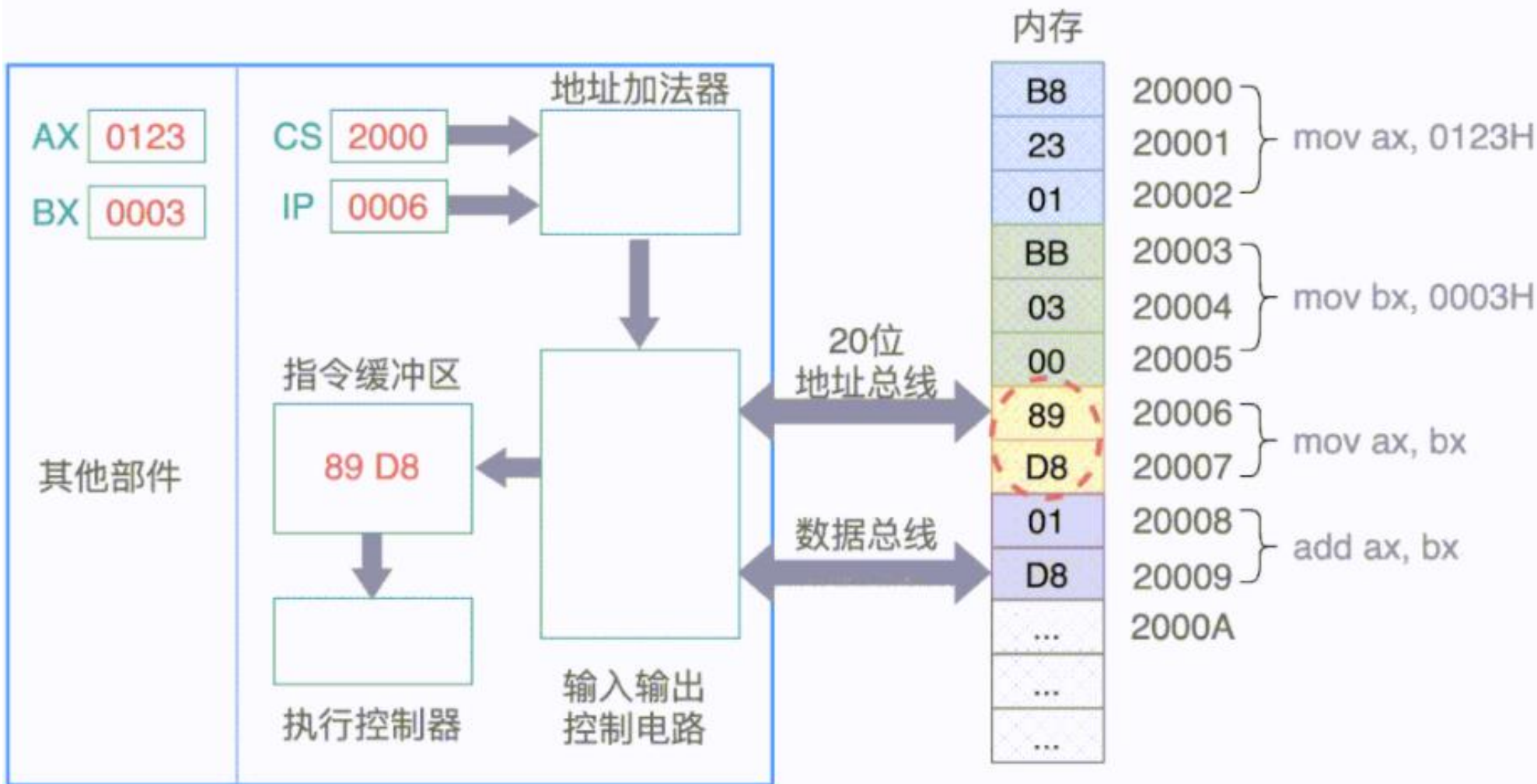
继续指令指令 89 D8 (mov ax, bx)



指令队列操作



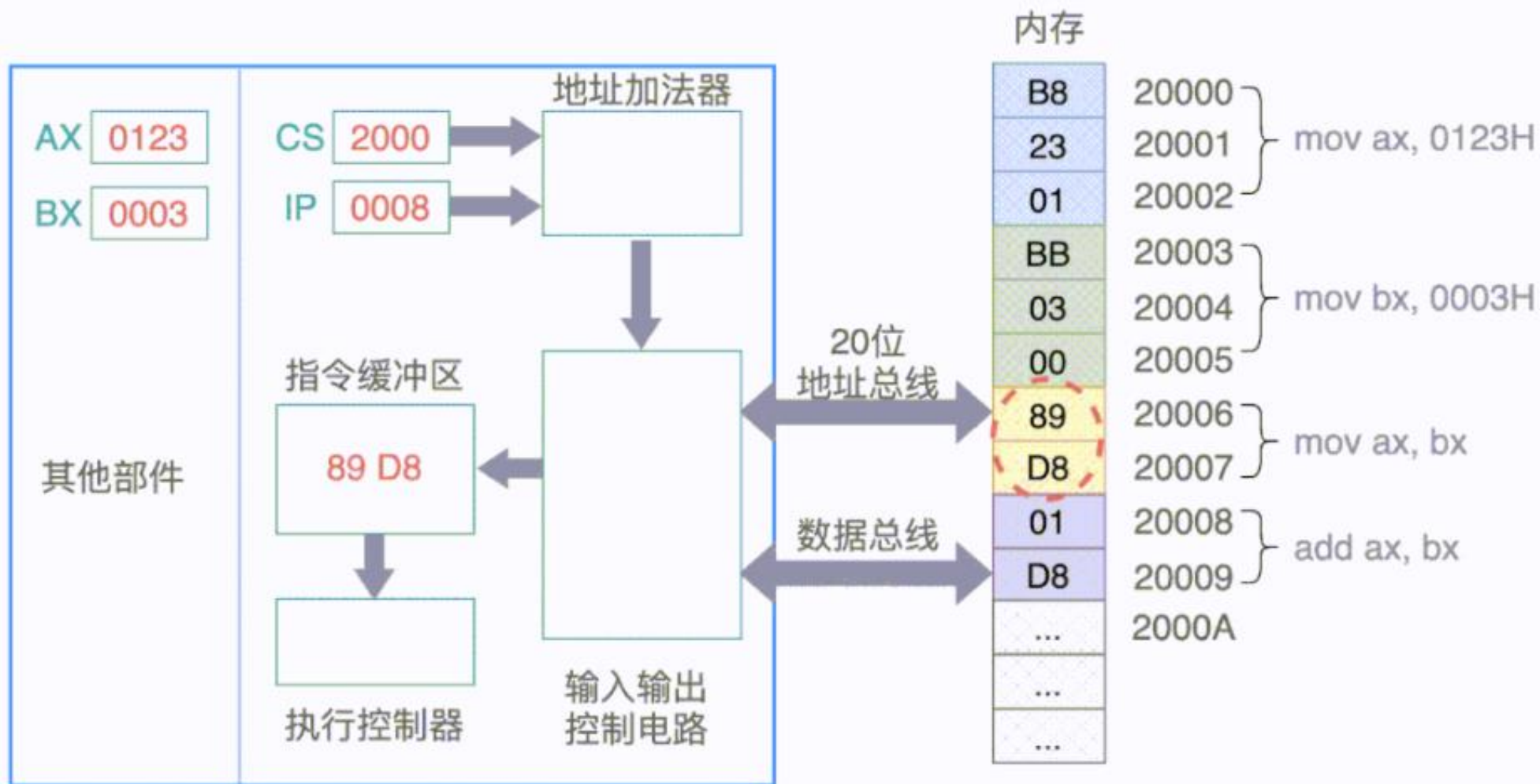
继续指令指令 89 D8 (mov ax, bx)



指令队列操作



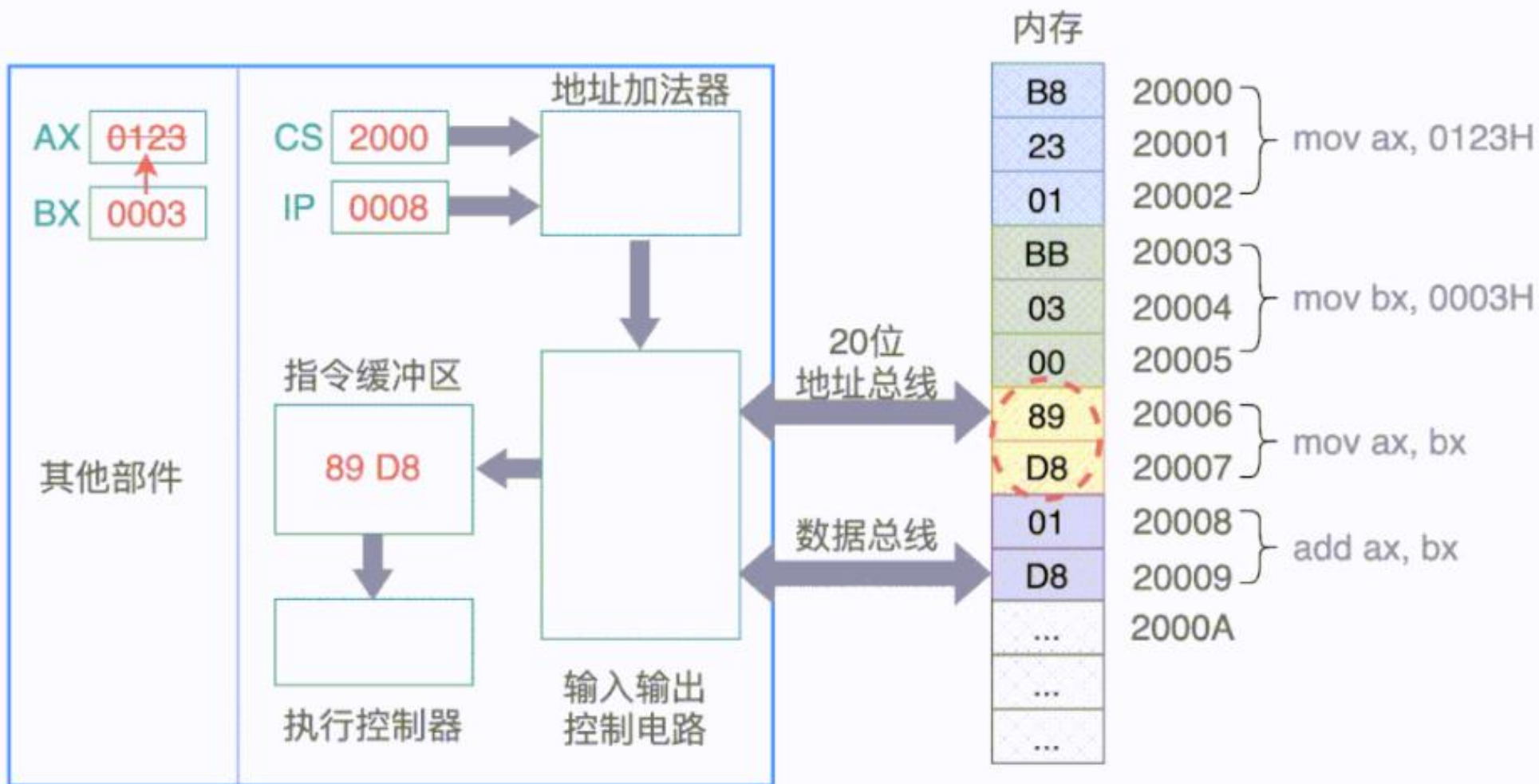
继续指令指令 89 D8 (mov ax, bx)



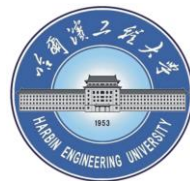
指令队列操作



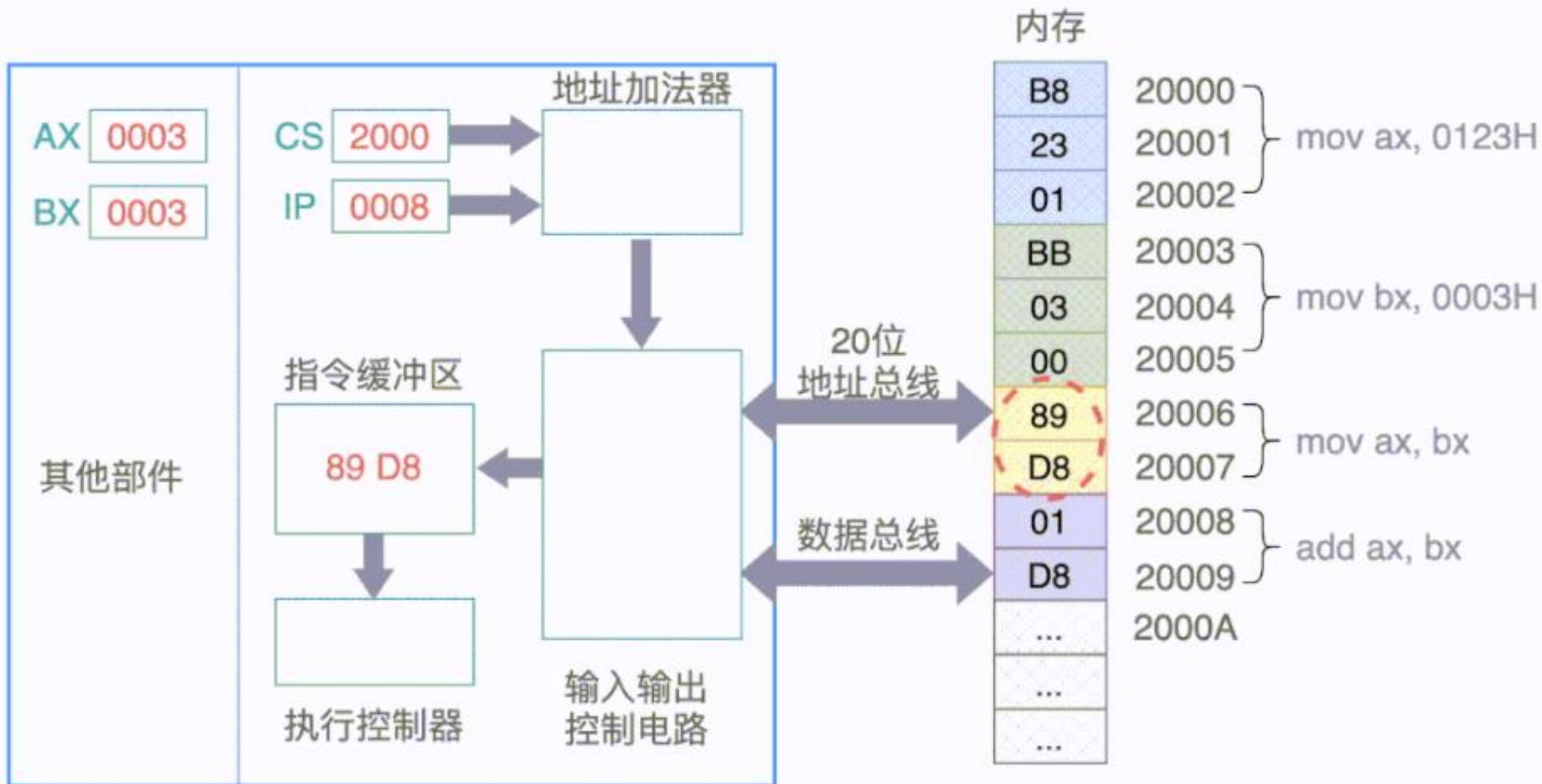
继续指令指令 89 D8 (mov ax, bx)



指令队列操作



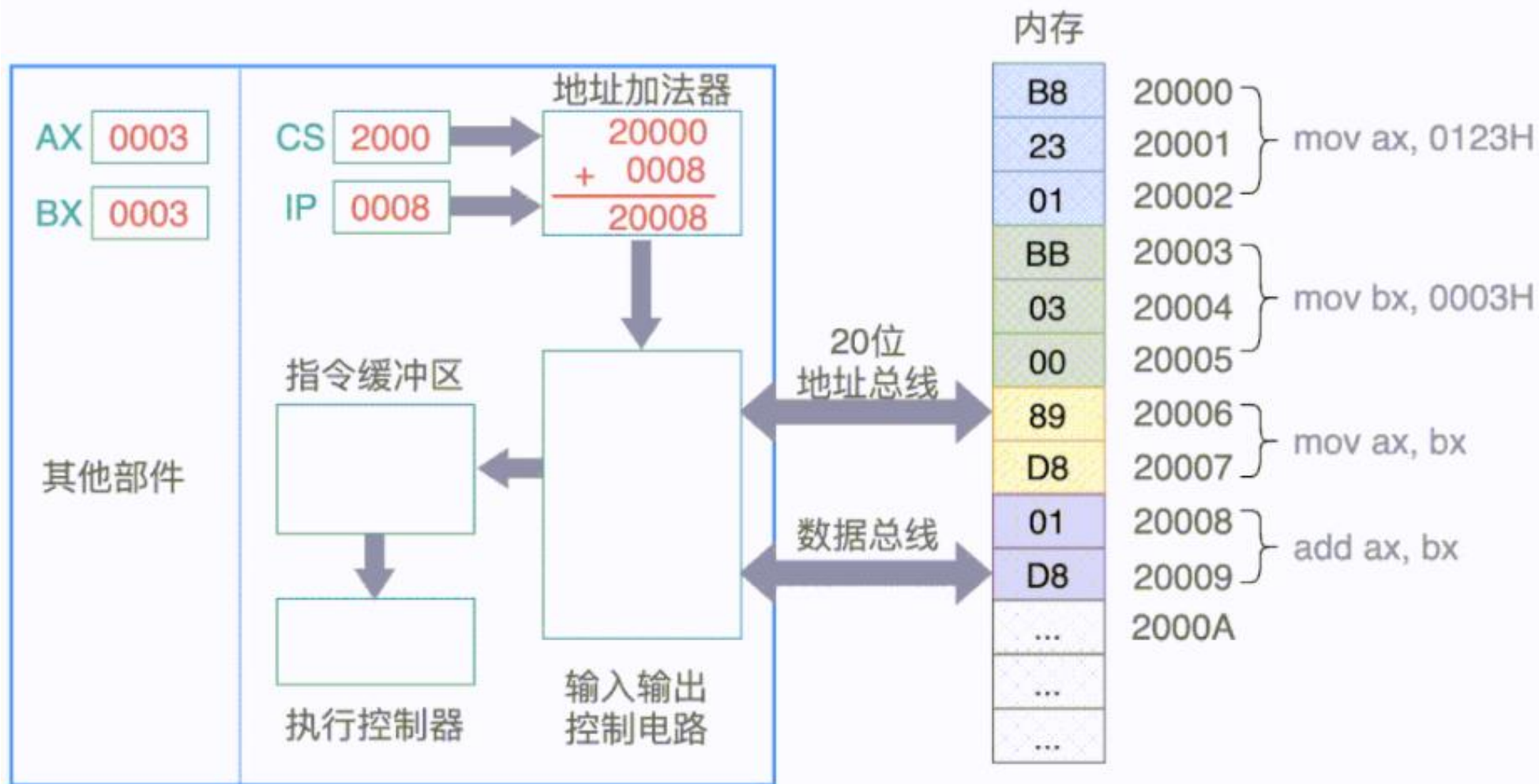
继续指令指令 89 D8 (mov ax, bx)



指令队列操作



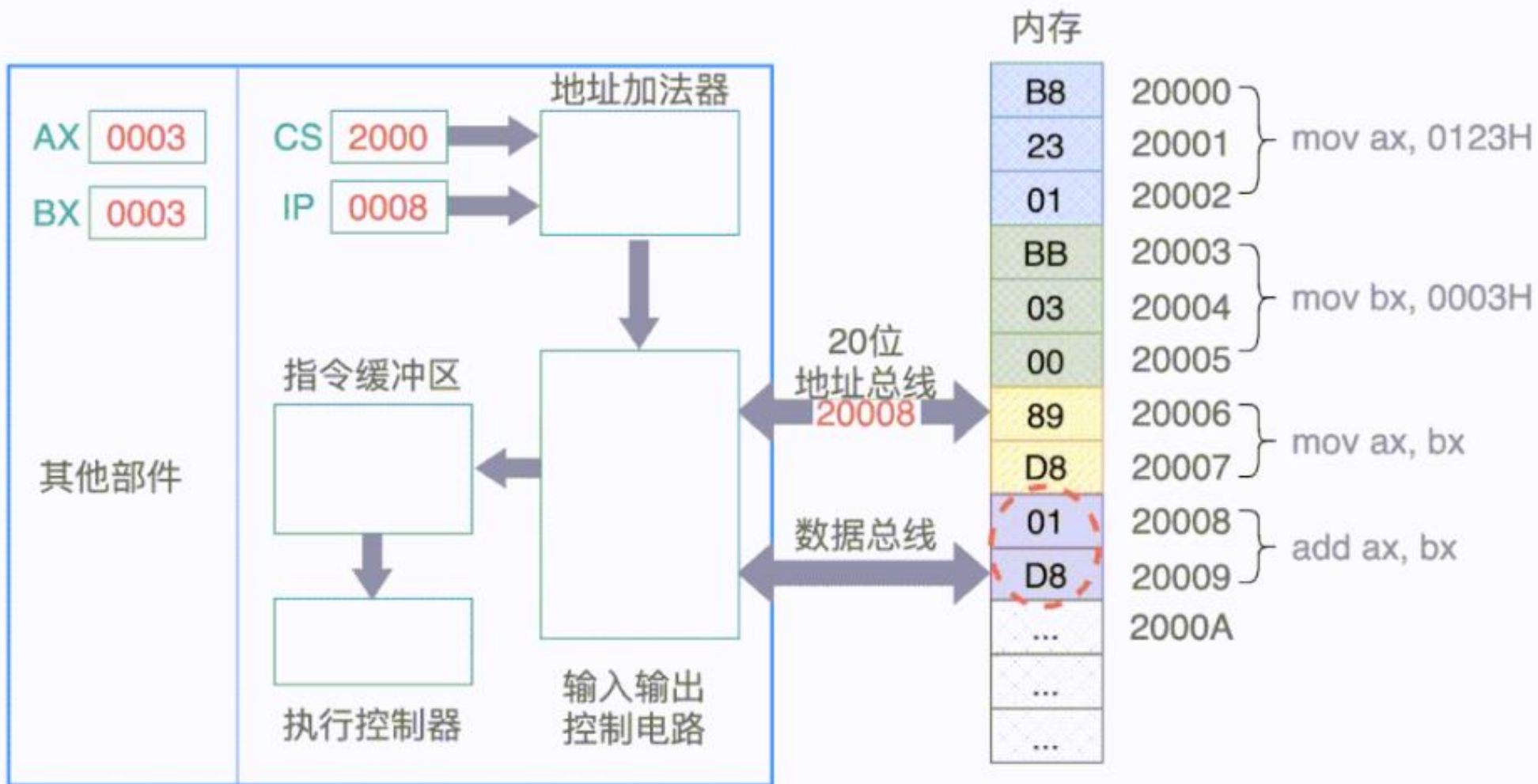
继续指令指令 01 D8 (add ax, bx)



指令队列操作



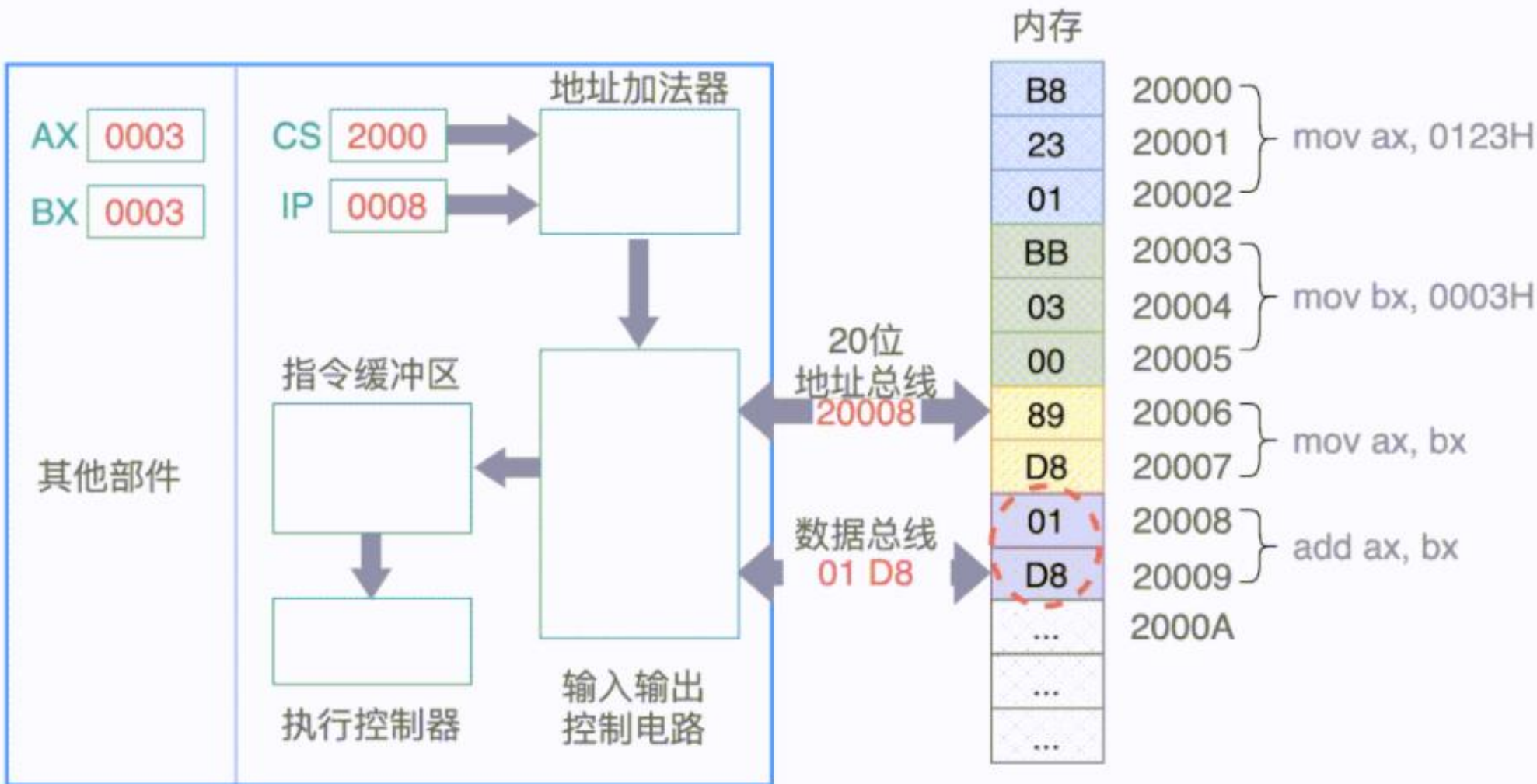
继续指令指令 01 D8 (add ax, bx)



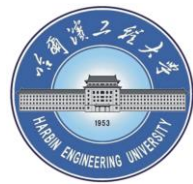
指令队列操作



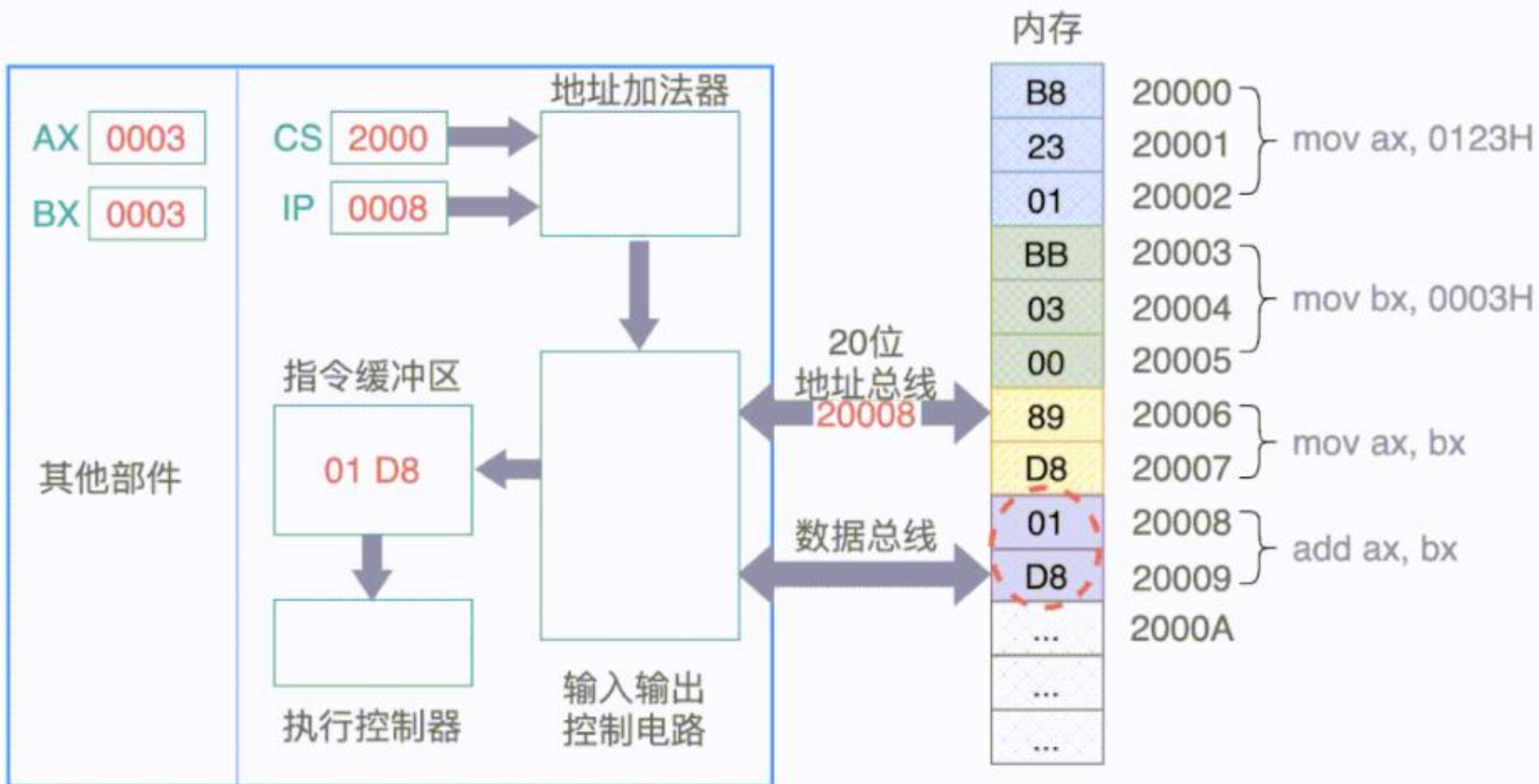
继续指令指令 01 D8 (add ax, bx)



指令队列操作



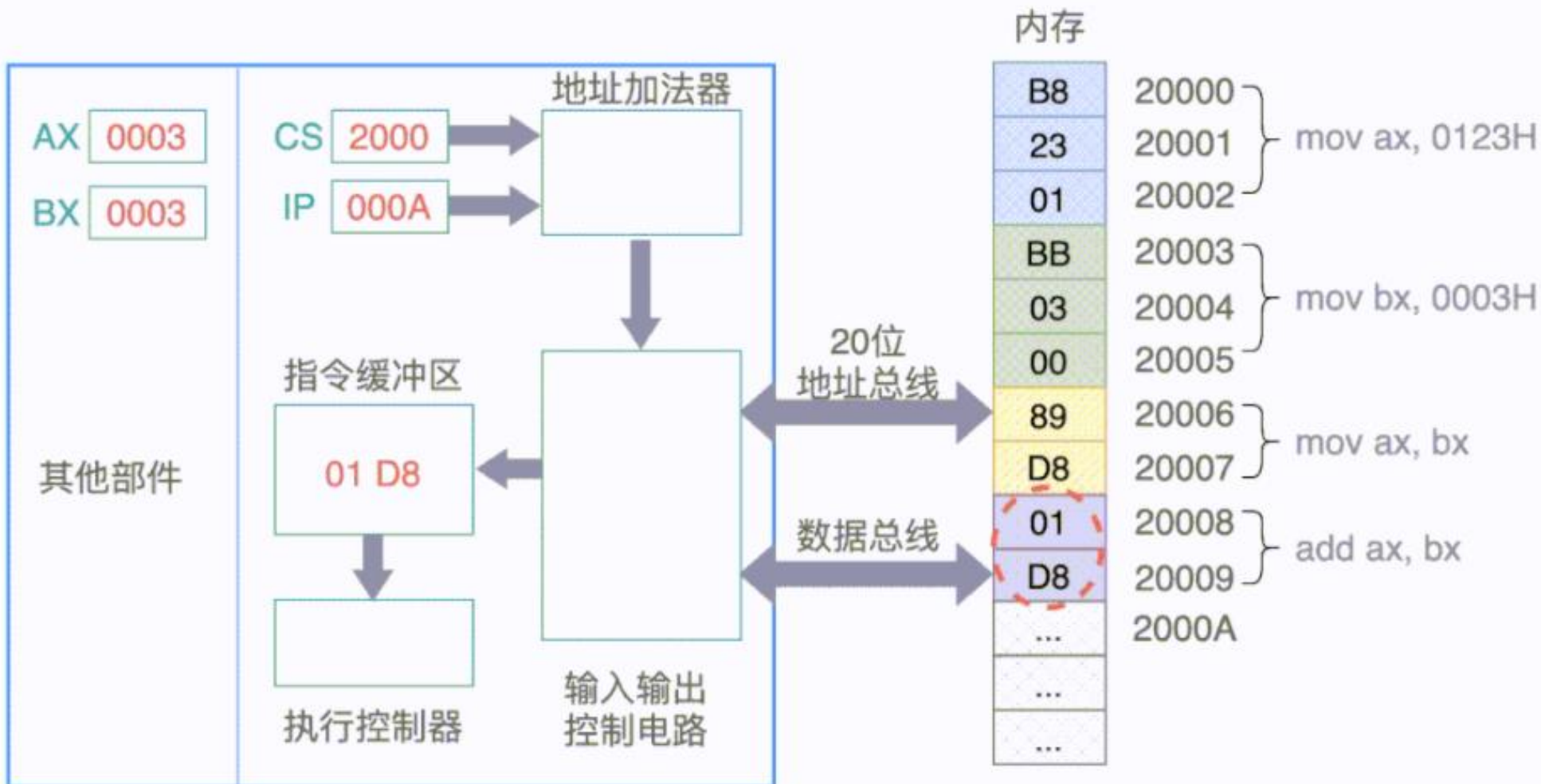
继续指令指令 01 D8 (add ax, bx)



指令队列操作



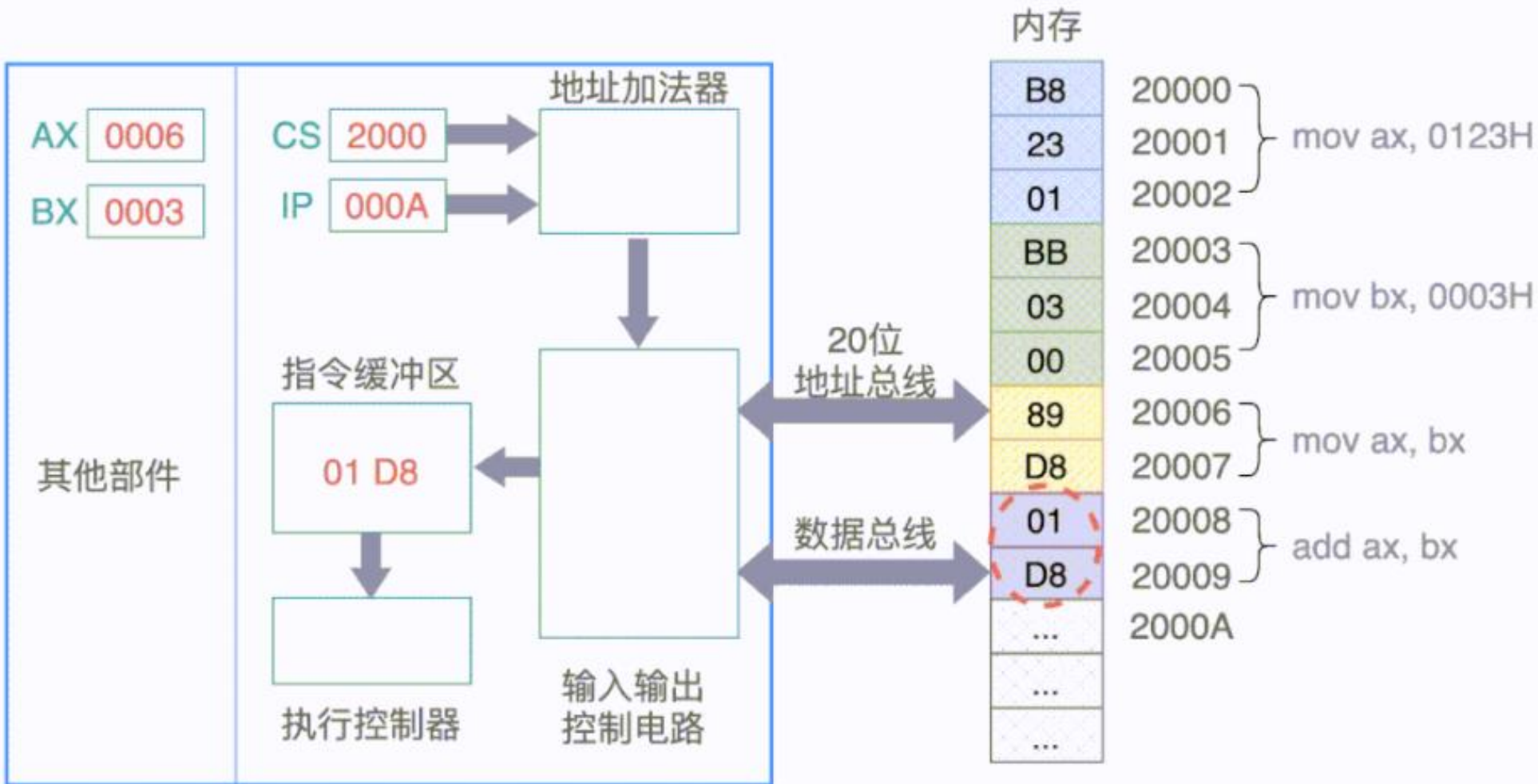
继续指令指令 01 D8 (add ax, bx)



指令队列操作



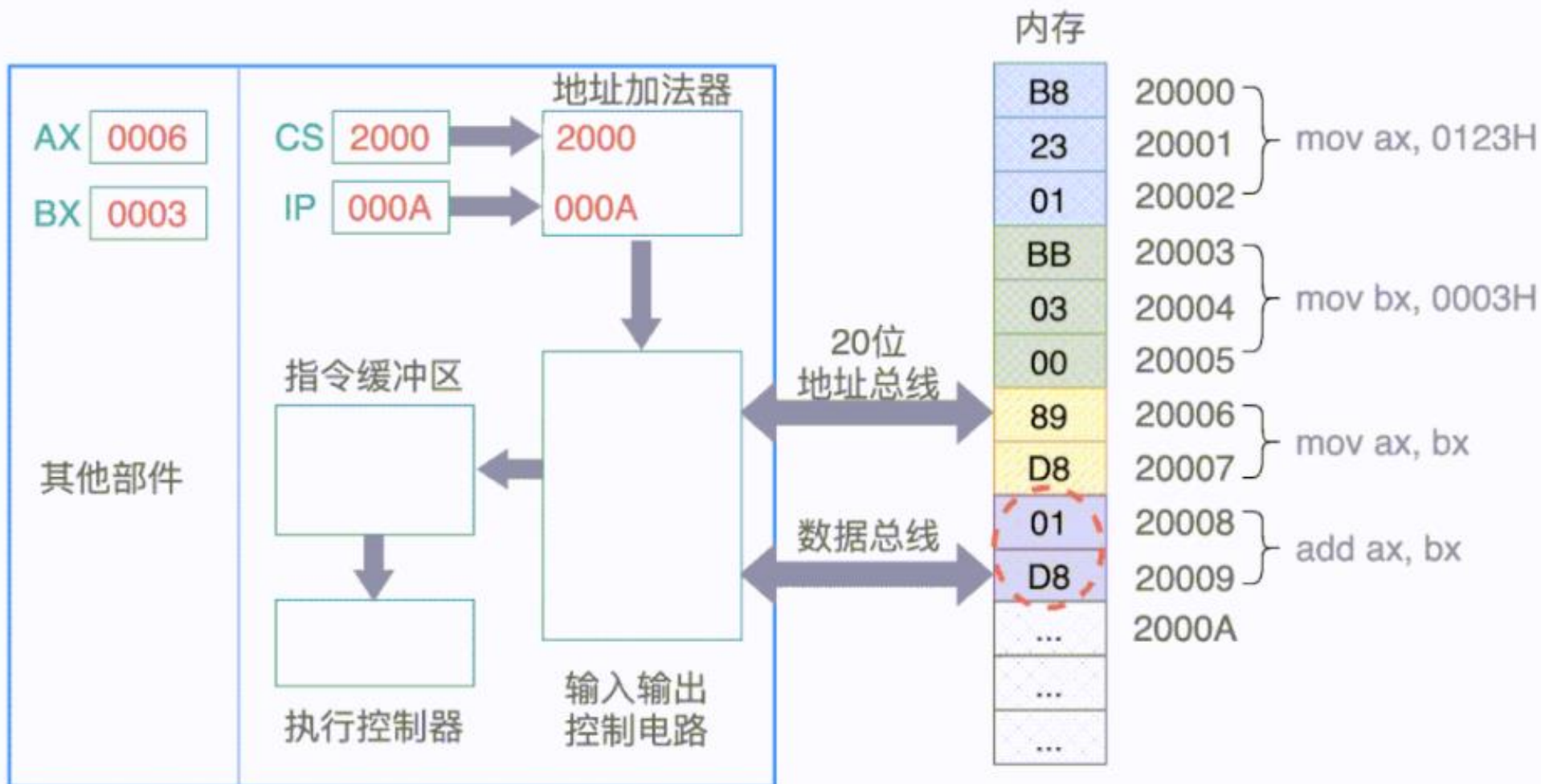
继续指令指令 01 D8 (add ax, bx)



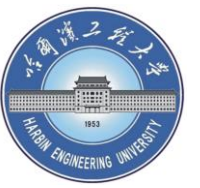
指令队列操作



继续执行后面的指令 2000:000A



执行部件EU (Execution Unit)



(1) 四个16位数据寄存器:

AX、BX、CX、DX, 每个16位寄存器可作8位寄存器。

- 16位 8位 8位
 - AX → AH AL
 - BX → BH BL
 - CX → CH CL
 - DX → DH DL
- 可进行16位数据运算——字运算；8位数据运算——字节运算



执行部件EU (Execution Unit)

(2) 指针和变址寄存器

二个指针寄存器:

- SP: 堆栈指针: 永远指向系统栈最上面一个栈帧的栈顶。
- BP: 基址指针: 永远指向系统栈最上面一个栈帧的底部。

二个变址寄存器:

- SI: (Source Index) : SI含有源地址意思, 产生有效地址或实际地址的偏移量。
- DI: (Destination Index) : DI含有目的意思, 产生有效地址或实际地址的偏移量。



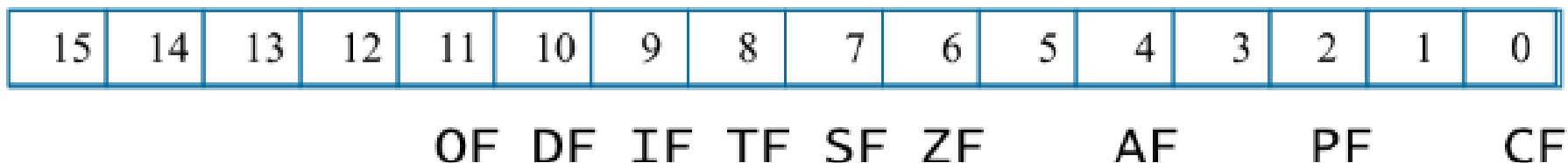
执行部件EU (Execution Unit)

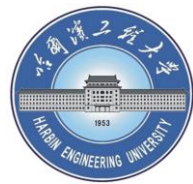
(3) 算术逻辑单元ALU:

- 主要是**加法器**。大部分指令的执行由加法器完成，8位字节运算和16位字运算。

(4) 标志寄存器:

- 16位字利用了**9位**，标志分两类：
 - **状态标志 (6位)**：反映刚刚完成的操作结果的特点。
 - **控制标志 (3位)**：在某些指令操作中起控制CPU行为的作用。





执行部件EU (Execution Unit)

标志位：反映指令执行结果的特征

(1) 进位标志CF:

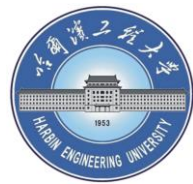
- 本次运算结果的最高位向更高位有**进位**或**借位**时, $CF=1$, 否则 $CF=0$;
- 最高位: 16位运算(字运算)时指 **D_{15}** , 8位运算(字节)时指 **D_7** 。

(2) 辅助进位标志AF:

- 本次运算结果的**最低4位**向更高位有进位或借位时, $AF=1$, 否则 $AF=0$;
- 不论是字运算还是字节运算, 都是指 **D_3 位**向 **D_4 位**的进位或借位;

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

OF DF IF TF SF ZF AF PF CF



执行部件EU (Execution Unit)

标志位：反映指令执行结果的特征

(3) 奇偶校验标志PF:

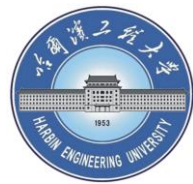
- 本次运算结果的**低8位**有偶数个“1”时，PF=1，否则PF=0；
- 不论是字运算还是字节运算，PF都只与运算结果的**低8位**有关，而与**高8位**无关；

(4) 全零标志ZF:

- **运算结果全为零时**，ZF=1，否则ZF=0；

(5) 符号标志SF:

- 与本次运算结果的**最高位D₁₅或D₇**相同；



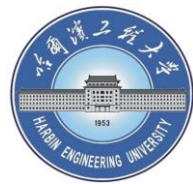
执行部件EU (Execution Unit)

加法运算的溢出

- 运算溢出的定义：运算结果超出了结果寄存器所能表示的带符号数范围。此时运算结果不能代表正确运算结果，也就是说，结果是错误的。
- 溢出的判别：双高位判别法

最高位有进位， $CF=1$ ；否则 $CF=0$ 。

数值部分最高位有进位， $CP=1$ ，否则 $CP=0$ 。



执行部件EU (Execution Unit)

加法运算的溢出判断

- $CF \oplus CP = 1$, 有溢出。

- CF CP $CF \oplus CP$

- 1 0 1 溢出

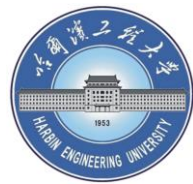
- 0 1 1 溢出

- 0 0 0 无溢出

- 1 1 0 无溢出

- $CF = CP$: CF 把 CP 的进位状态正确传递给更高

- $CF \neq CP$: CF 没有把 CP 的进位状态正确传递给更高位



执行部件EU (Execution Unit)

八位补码运算： (-128 ~ +127)

• 0101 1010 +90

• + 0110 0011 +99

• 1011 1101 -67

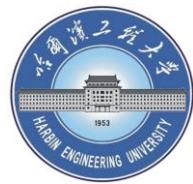
• CF=0, CP=1, $CF \oplus CP=1$, 产生溢出, 结果出错。

• 1001 0010 -110

• + 0010 0110 + 54

• 1011 1000 -56

• CF=0, CP=0, $CF \oplus CP=0$, 没有溢出, 结果正确。



执行部件EU (Execution Unit)

八位补码运算： (-128 ~ +127)

- $1101\ 1010$ -38
- $+ \underline{1110\ 0011}$ $\underline{-29}$
- $11011\ 1101$ -67
- $CF=1, CP=1, CF \oplus CP=0$, 没有溢出, 结果正确。



执行部件EU (Execution Unit)

控制位：控制CPU的某些行为

(1) 跟踪标志TF:

- TF=0, CPU按正常方式执行指令。
- TF=1, CPU按单步方式执行指令。

(2) 中断允许标志IF:

- IF=1, 允许CPU响应可屏蔽中断;
- IF=0, 禁止CPU响应可屏蔽中断。

(3) 方向标志DF:

- DF=0, 串指令的地址向增加方向变化;
- DF=1, 串指令的地址向减小方向变化。



执行部件EU (Execution Unit)

例1、CF、OF、PF、AF、ZF、SF各为何值？

```

      0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 1
+     0 1 1 0 0 1 0 1 1 0 0 1 0 1 1 0
-----
      1 0 1 1 0 0 1 0 0 0 0 0 0 0 1 1
  
```

最高位向
更高位无
进位
CF=0

SF =
结果的
最高位
SF=1

次高位向
最高位
有进位
CP=1

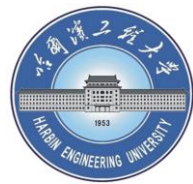
低8位有偶
数个“1”
PF=1

D3向D4有
进位
AF=1

结果不全
为零
ZF=0

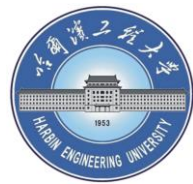
OF = CF ⊕ CP = 1

CF ≠ CP: CF没有正确地把CP的进位状态传送给更高位



第 2 章 8086微处理器

- 2-1 8086 CPU结构
- 2-2 8086 CPU的引脚及其功能
- 2-3 8086 存储器组织
- 2-4 8086 系统配置
- 2-5 8086 CPU时序



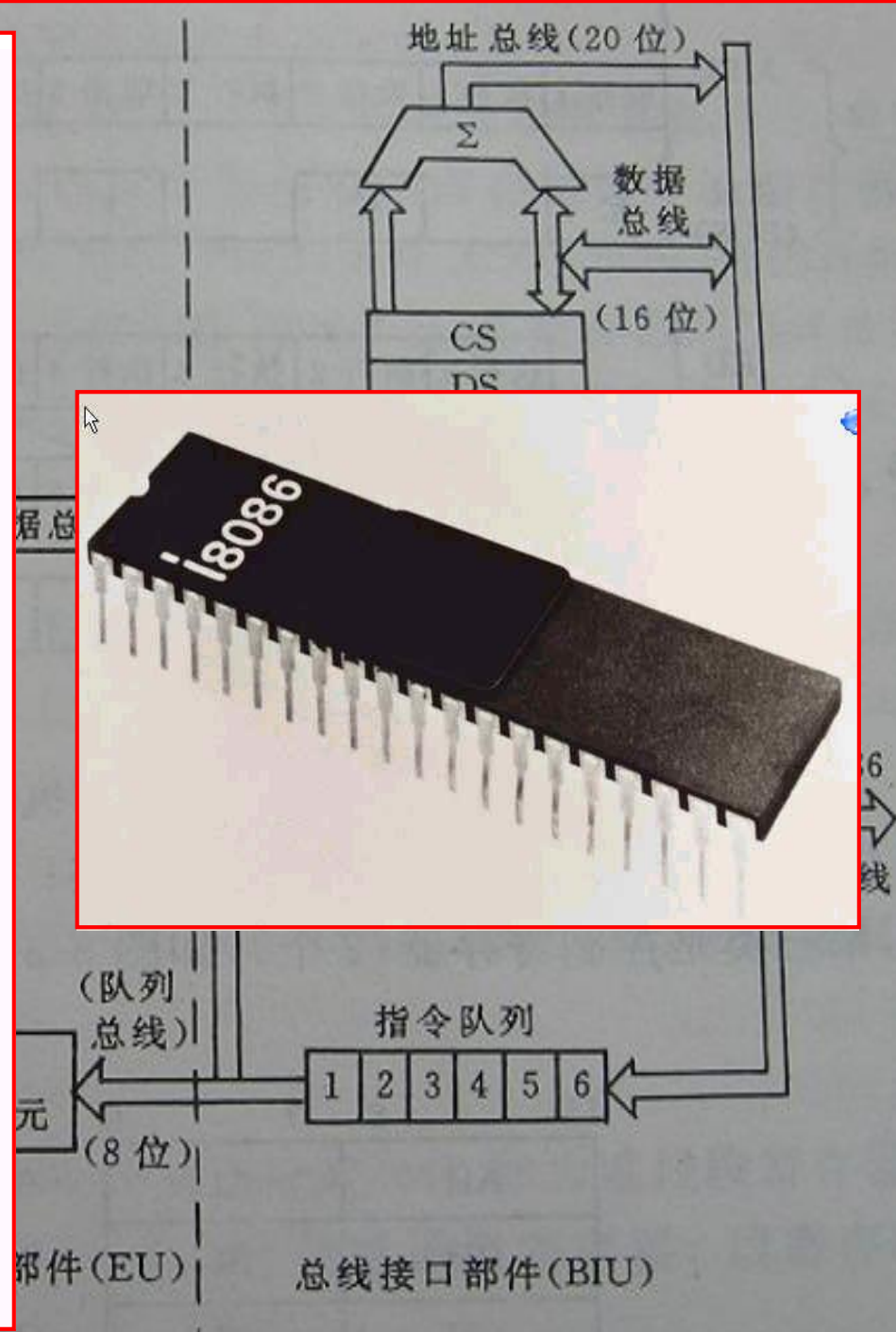
2-2 8086的引脚及其功能

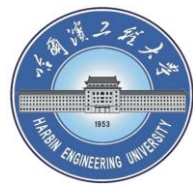
8086为DIP-40封装，包括**20条地址线**（**16条数据/地址线**，**4条状态/地址线**），**3个电源引脚**，**17条控制或状态信号线**。

两种工作模式：

- **最小模式：**系统中只有8086一个微处理器，所有的总线控制信号均为8086产生，系统中的总线控制逻辑电路，减少到最少。（**要求掌握**）
- **最大模式：**用于大型（中型）8086/8088系统中。系统总是包含有两个或多个微处理器，其中一个主处理器就是8086或8088，其它的处理器称协处理器，协助主处理器工作。所有的总线控制信号均由总线控制器8288产生。

地	1		40	$V_{CC}(5V)$
AD_{14}	2		39	AD_{13}
AD_{13}	3		38	A_{16}/S_3
AD_{12}	4		37	A_{17}/S_4
AD_{11}	5		36	A_{18}/S_5
AD_{10}	6		35	A_{19}/S_6
AD_9	7		34	\overline{BHE}/S_7
AD_8	8		33	$\overline{MN}/\overline{MX}$
AD_7	9		32	\overline{RD}
AD_6	10	8086	31	$\overline{HOLD}(\overline{RQ}/\overline{GT_0})$
AD_5	11	CPU	30	$\overline{HLDA}(\overline{RQ}/\overline{GT_1})$
AD_4	12		29	$\overline{WR}(\overline{LOCK})$
AD_3	13		28	$M/\overline{IO}(S_2)$
AD_2	14		27	$DT/\overline{R}(S_1)$
AD_1	15		26	$\overline{DEN}(S_0)$
AD_0	16		25	$ALE(QS_0)$
NMI	17		24	$\overline{INTA}(QS_1)$
INTR	18		23	TEST
CLK	19		22	READY
地	20		21	RESET



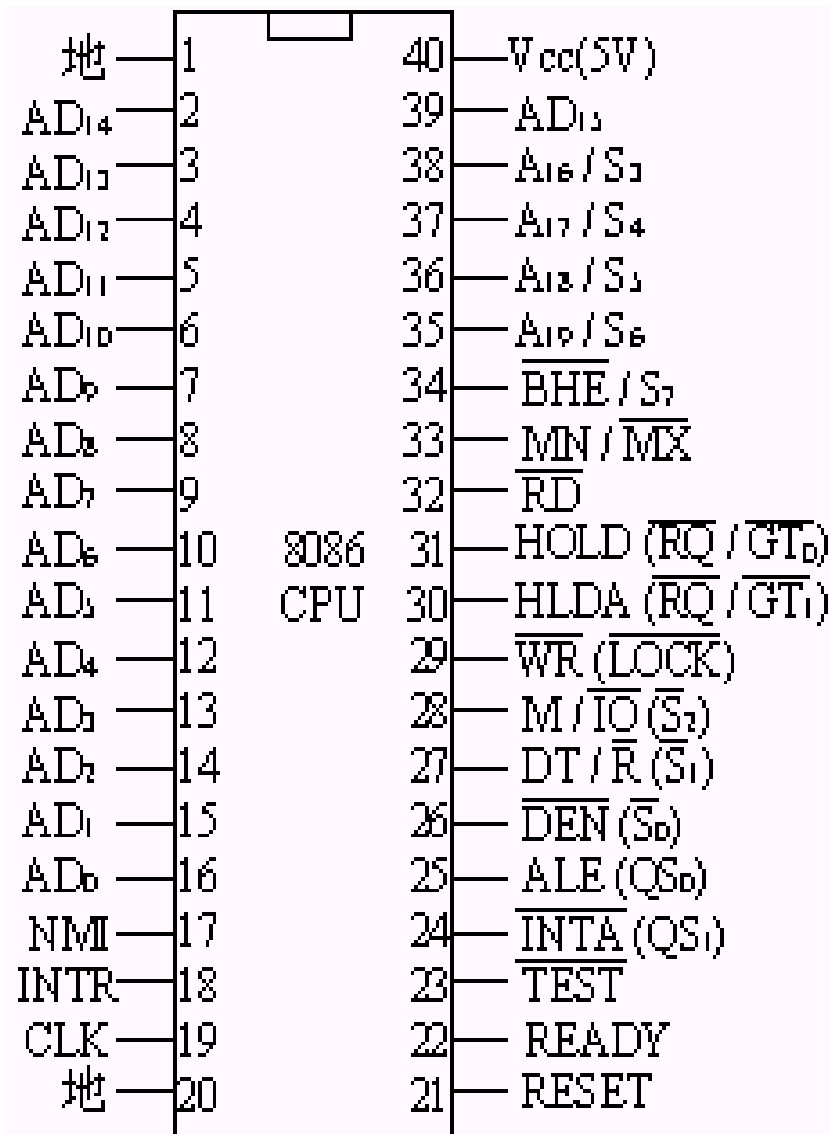


8086CPU外部引脚

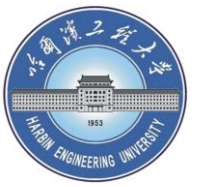
- 8088/8086CPU都具有40条引脚，采用双列直插式封装，有的引脚具有**双功能**。

① 为了减少芯片的引线，8088/8086的许多引脚具有双重定义和功能，采用**分时复用**方式工作，即在不同时刻，这些引线上的信号是不相同的。

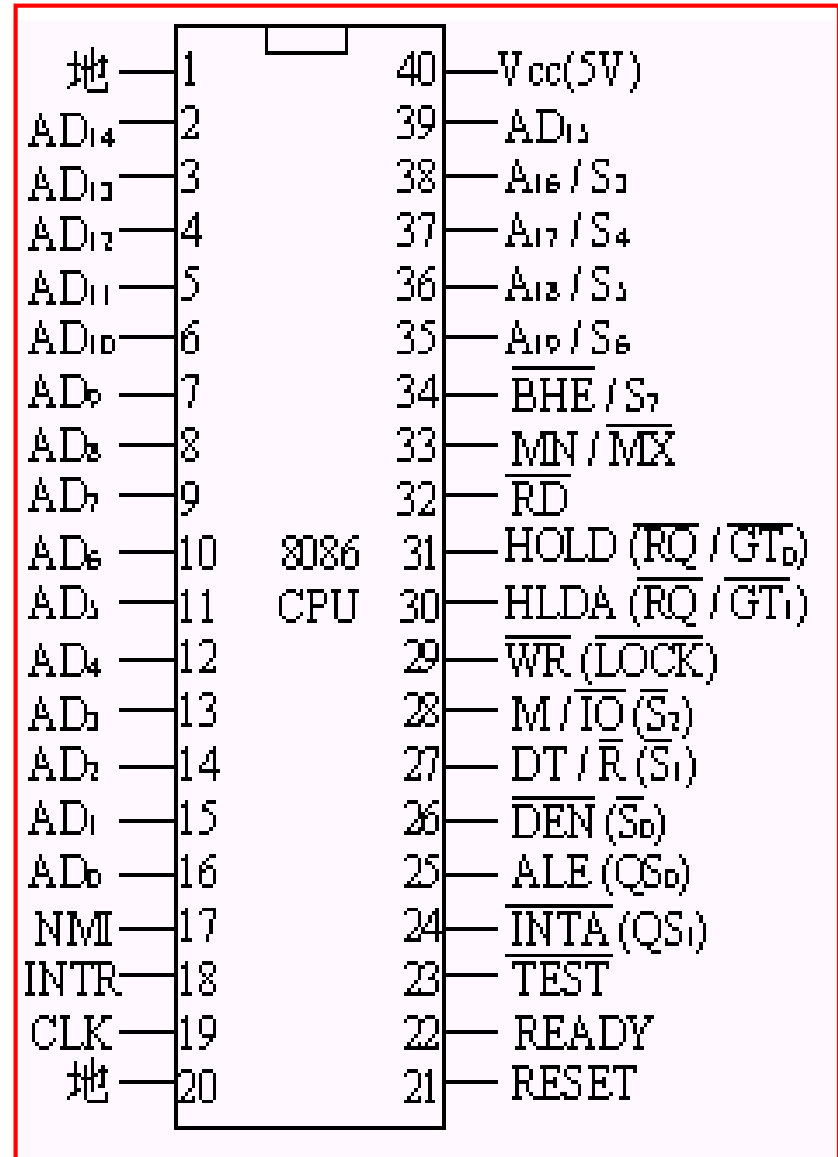
② 8088/8086的**最大**和**最小**两种工作模式可以通过引脚选择。



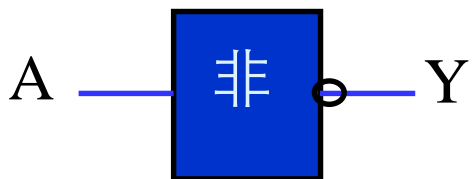
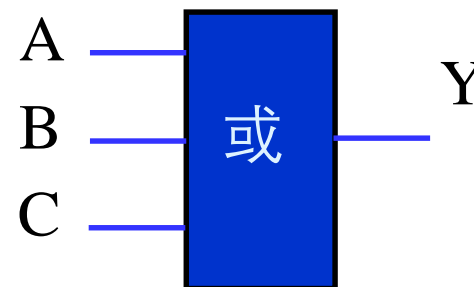
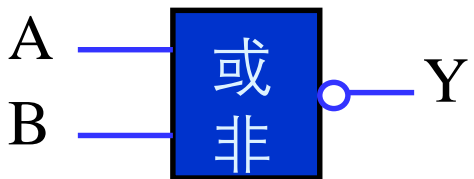
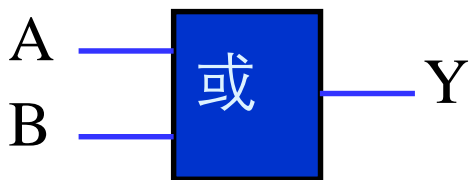
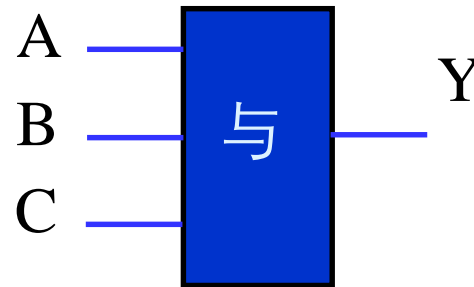
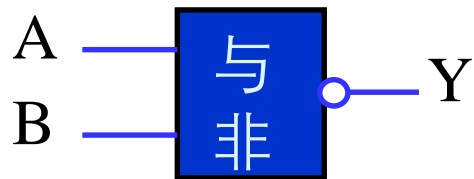
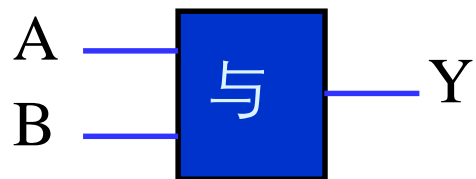
基本概念



- 1、**引脚的功能**：即引脚的定义，其名字反映了该信号的作用即含义。
- 2、**信号的流向**：信号从芯片输出，还是从外部输入芯片，或双向。
- 3、**有效电平**：使引脚起作用的**逻辑电平**。
- 4、**三态能力**：有些引脚除了能正常输入输出外，还能输出**高阻状态**，此时表示芯片已放弃了对该引脚的控制，使之**悬空**，以方便其他设备接管对它的控制

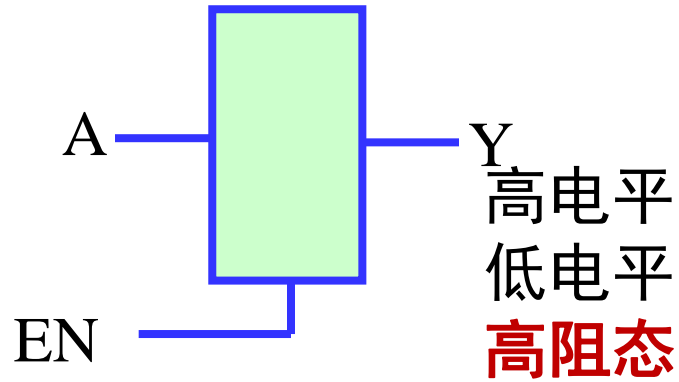
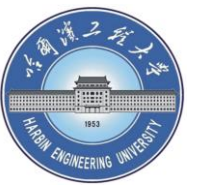


逻辑门

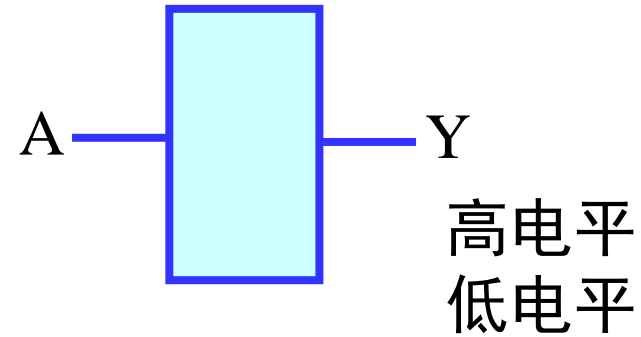


一般逻辑门只有两种输出状态：高电平和低电平

三态门



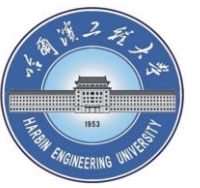
三态门



逻辑门

- 三态门比逻辑门增加了一个**控制端EN**（又称高能端）当控制端有效时，三态门处于工作状态，否则处于**高阻态**。
- **高阻态**：相当于该门和它连接的电路处于**断开**的状态，主要是用于总线的连接，因为**总线只允许同时有一个使用者**。

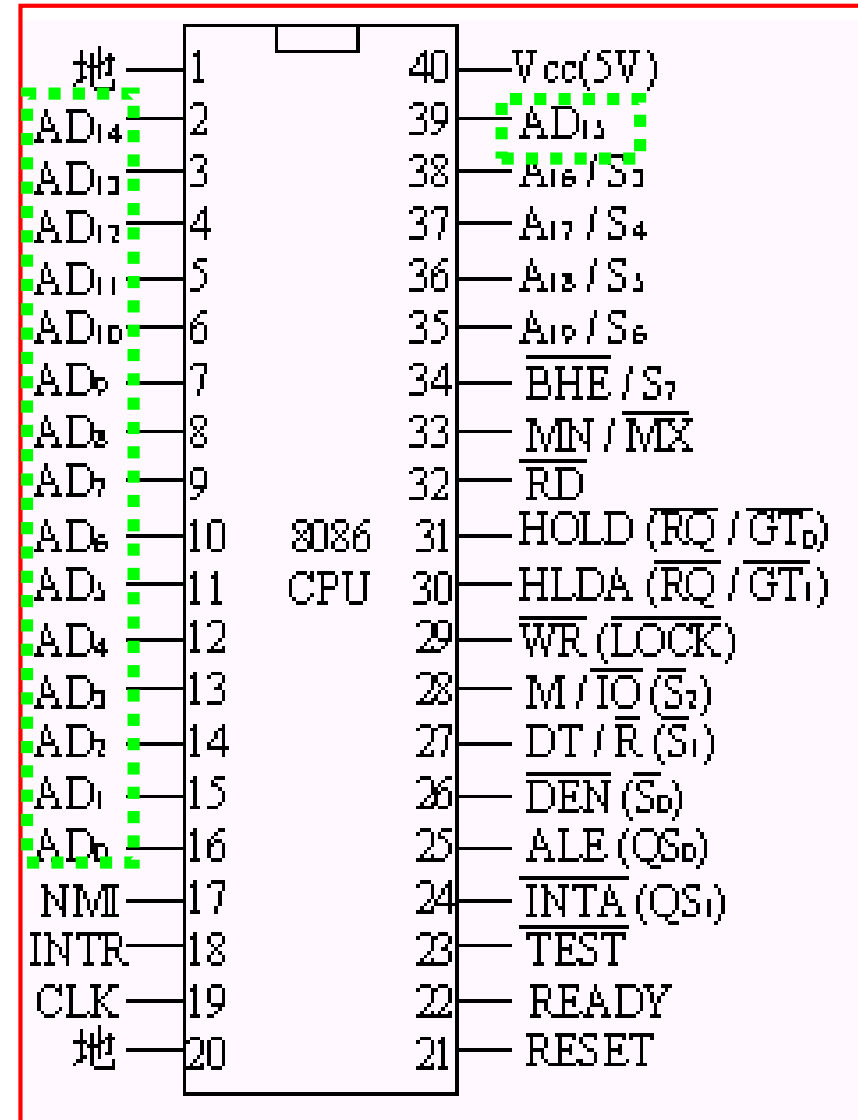
8086/8088CPU外部引脚



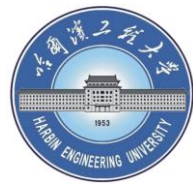
1、 $AD_{15}-AD_0$ (Address Data Bus)

分时复用的地址/数据线，

- 传送地址时单向三态输出。
- 传送数据时可双向三态输入/输出。



8086/8088CPU外部引脚



2、 $A_{19}/S_6 - A_{16}/S_3$ (Address /Status) 分时复用的地址/状态线

- 作地址线时，用 $A_{19} \sim A_{16}$ 与 $AD_{15} \sim AD_0$ 一起构成访问存储器的20位物理地址。
- 注意：当CPU访问I/O端口时 $A_{19} - A_{16}$ 保持为0。

地	1		40	$V_{CC}(SV)$
AD_{14}	2		39	AD_0
AD_{13}	3		38	A_{16}/S_3
AD_{12}	4		37	A_{17}/S_4
AD_{11}	5		36	A_{18}/S_5
AD_{10}	6		35	A_{19}/S_6
AD_9	7		34	BHE/S_7
AD_8	8		33	$\overline{MN}/\overline{MX}$
AD_7	9		32	\overline{RD}
AD_6	10	8086	31	$\overline{HOLD}(\overline{RQ}/\overline{GT}_D)$
AD_5	11	CPU	30	$\overline{HLDA}(\overline{RQ}/\overline{GT}_I)$
AD_4	12		29	$\overline{WR}(\overline{LOCK})$
AD_3	13		28	$M/\overline{IO}(\overline{S}_7)$
AD_2	14		27	$DT/\overline{R}(\overline{S}_1)$
AD_1	15		26	$\overline{DEN}(\overline{S}_6)$
AD_0	16		25	$ALE(QS_0)$
NMI	17		24	$\overline{INTA}(QS_1)$
INTR	18		23	TEST
CLK	19		22	READY
地	20		21	RESET

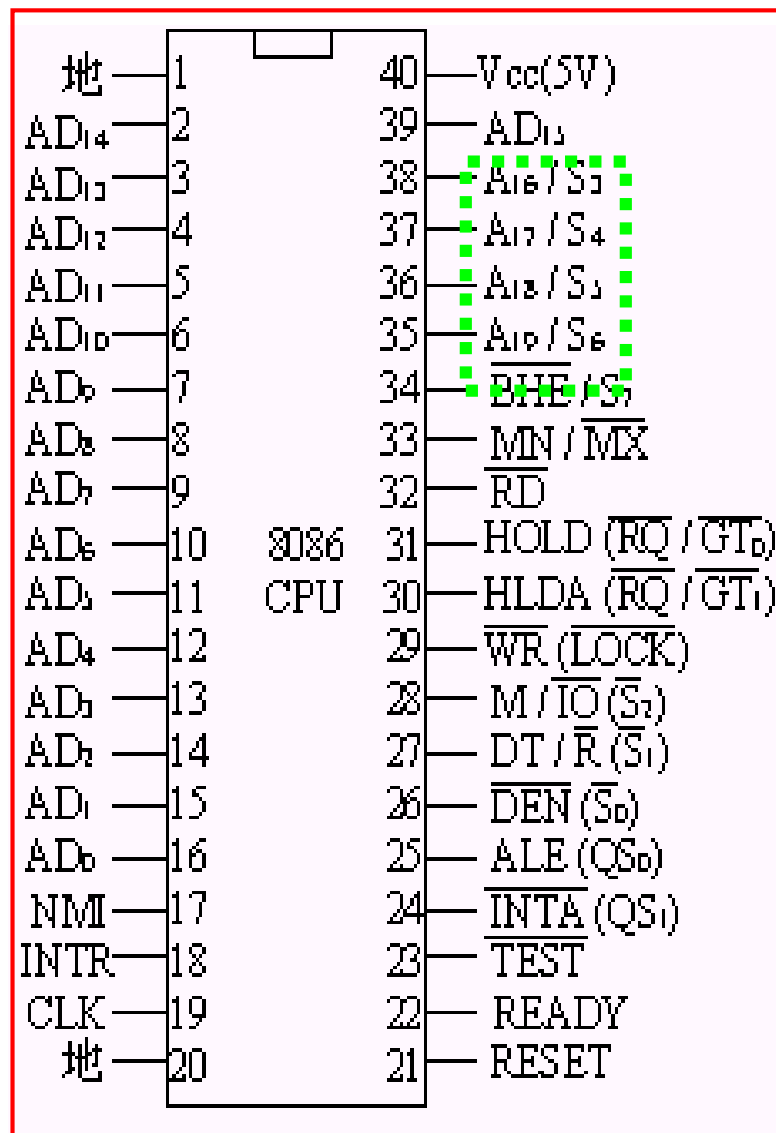
8086/8088CPU外部引脚



2、 $A_{19}/S_6 - A_{16}/S_3$ (Address /Status) 分时复用的地址/状态线

- 作状态线时， $S_6 - S_3$ 用来输出状态信息。
 $S_4 - S_3$ 用来确定当前使用的段寄存器。

S_4	S_3	段寄存器
0	0	ES
0	1	SS
1	0	CS(I/O,INT)
1	1	DS



8086/8088CPU外部引脚



$A_{19}/S_6 - A_{16}/S_3$ (Address /Status)

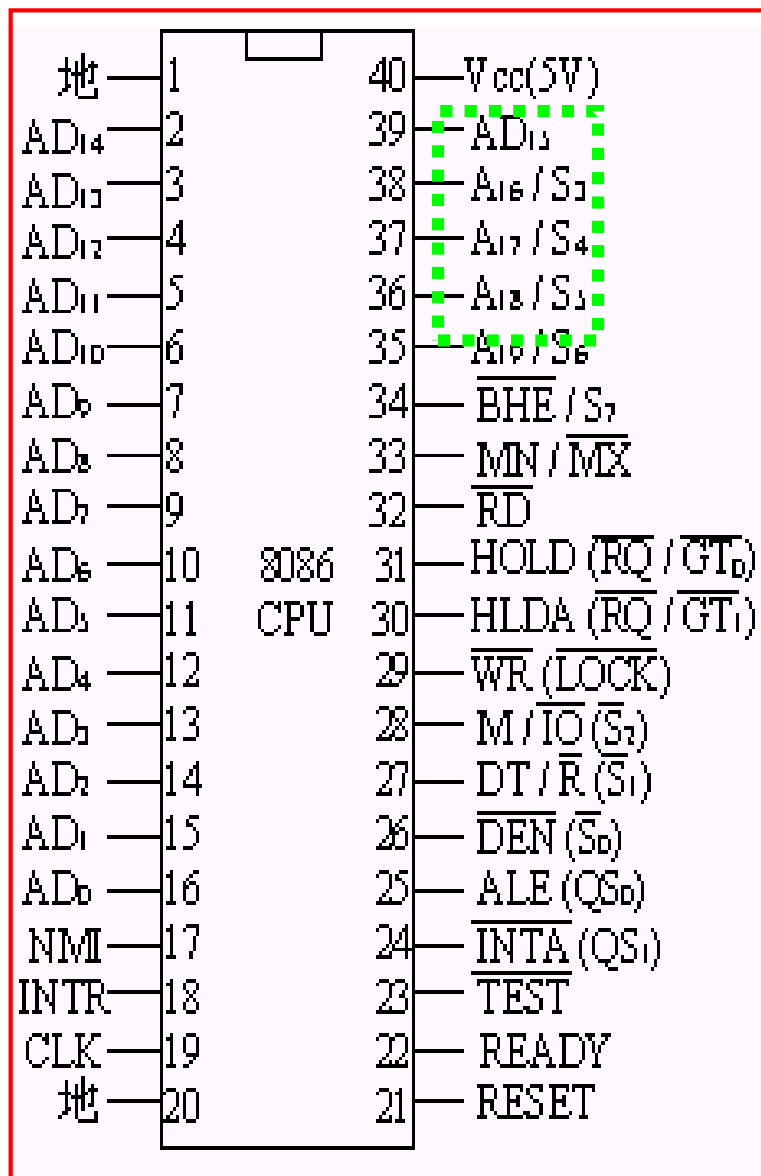
分时复用的地址/状态线。

S_4	S_3	段寄存器
0	0	ES
0	1	SS
1	0	CS(I/O,INT)
1	1	DS

S_5 用来表示中断标志状态。

当 $IF=1$ 时, S_5 置1。

S_6 恒保持为0。



8086/8088CPU外部引脚



3、RD(Read)

读信号，三态输出，低电平有效。

表示当前CPU正在读存储器或I/O端口。

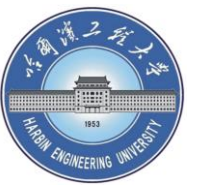
4、WR(Write)

写信号，三态输出，低电平有效。

表示当前CPU正在写存储器或I/O端口。

地	1	40	V _{CC} (5V)
AD ₁₄	2	39	AD ₁₃
AD ₁₃	3	38	A ₁₆ /S ₁
AD ₁₂	4	37	A ₁₇ /S ₄
AD ₁₁	5	36	A ₁₂ /S ₃
AD ₁₀	6	35	A ₁₀ /S ₆
AD ₉	7	34	$\overline{\text{BHE}}/S_7$
AD ₈	8	33	$\overline{\text{MN}}/\overline{\text{MX}}$
AD ₇	9	32	$\overline{\text{RD}}$
AD ₆	10	31	HOLD (RQ / GT ₀)
AD ₅	11	30	HLDA (RQ / GT ₁)
AD ₄	12	29	$\overline{\text{WR}}(\text{LOCK})$
AD ₃	13	28	M / IO (S ₂)
AD ₂	14	27	DT / R (S ₁)
AD ₁	15	26	$\overline{\text{DEN}}(\overline{S_0})$
AD ₀	16	25	ALE (QS ₀)
NMI	17	24	$\overline{\text{INTA}}(\text{QS}_1)$
INTR	18	23	$\overline{\text{TEST}}$
CLK	19	22	READY
地	20	21	RESET

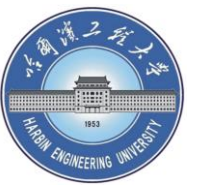
8086/8088CPU外部引脚



5、M/I $\overline{\text{O}}$ (Memory/I $\overline{\text{O}}$)存储器或I/O端口访问信号，三态输出。

- 高电平：表示CPU正在访问**存储器**。
- 低电平：表示CPU正在访问**I/O端口**。

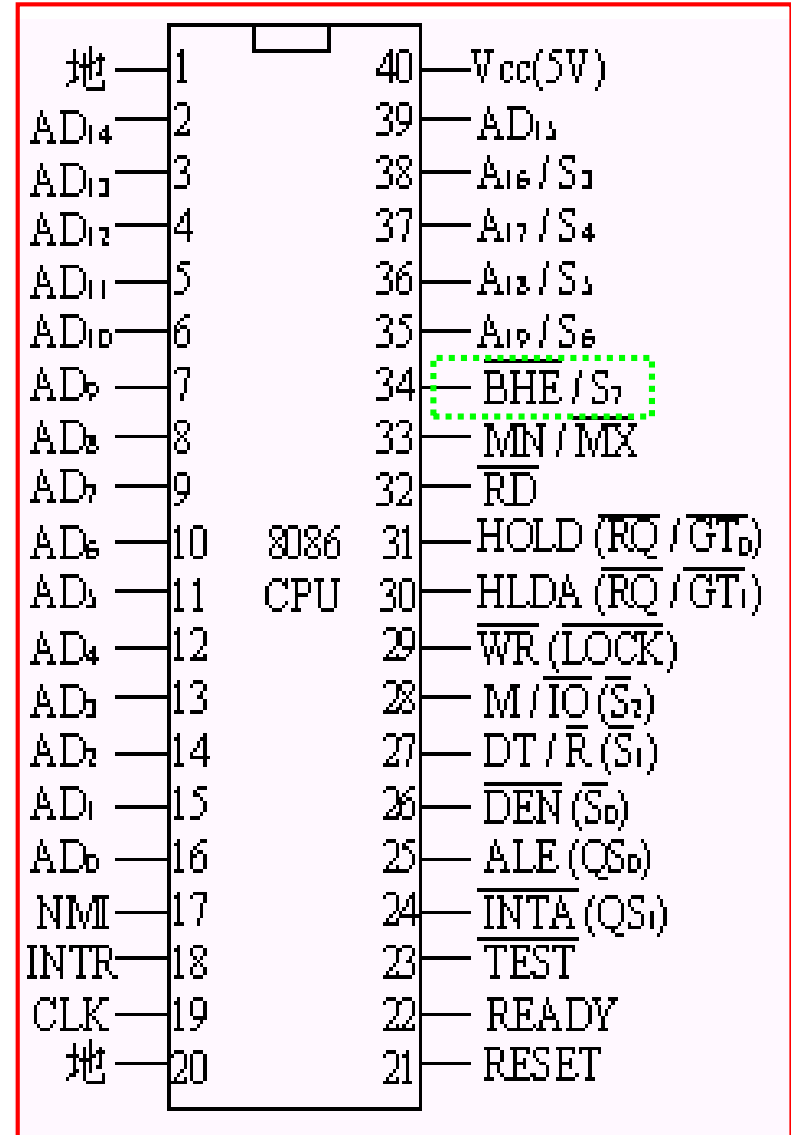
地	1	40	V _{CC} (5V)
AD ₁₄	2	39	AD ₁₃
AD ₁₃	3	38	A ₁₆ /S ₃
AD ₁₂	4	37	A ₁₇ /S ₄
AD ₁₁	5	36	A ₁₈ /S ₅
AD ₁₀	6	35	A ₁₉ /S ₆
AD ₉	7	34	$\overline{\text{BHE}}$ /S ₇
AD ₈	8	33	$\overline{\text{MN}}/\overline{\text{MX}}$
AD ₇	9	32	$\overline{\text{RD}}$
AD ₆	10	8086 CPU 31	HOLD ($\overline{\text{RQ}}$ / $\overline{\text{GT}}_0$)
AD ₅	11	30	HLDA ($\overline{\text{RQ}}$ / $\overline{\text{GT}}_1$)
AD ₄	12	29	$\overline{\text{WR}}$ (LOCK)
AD ₃	13	28	M/I $\overline{\text{O}}$ (S ₂)
AD ₂	14	27	DT/R (S ₁)
AD ₁	15	26	$\overline{\text{DEN}}$ (S ₀)
AD ₀	16	25	ALE (QS ₀)
NMI	17	24	$\overline{\text{INTA}}$ (QS ₁)
INTR	18	23	$\overline{\text{TEST}}$
CLK	19	22	READY
地	20	21	RESET



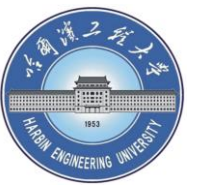
8086/8088CPU外部引脚

- **6、 $\overline{\text{BHE}}/\text{S}_7$ (Bus High Enable/ Status)总线高字节有效信号**
- 该信号三态输出，低电平有效。用来表示**数据总线上的数据是否有效。**

<u>BHE</u>	<u>AD₀</u>	总线使用情况
0	0	16位数据总线上进行字传送
0	1	高8位数据总线上进行字节传送
1	0	低8位数据总线上进行字节传送
1	1	无效

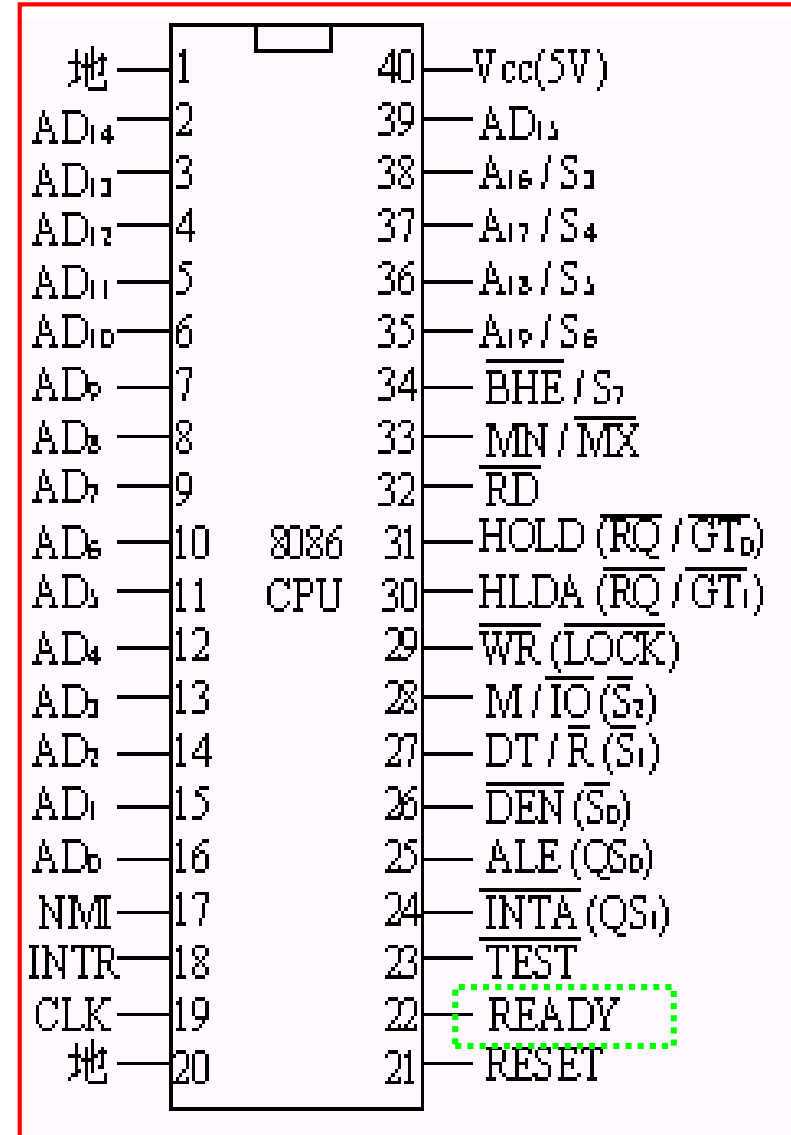


8086/8088CPU外部引脚



7、READY

- **准备就绪信号**，由**外部输入**，高电平有效。它是由被访问的内存或I/O设备发出的响应信号，当其有效时，表示存储器或I/O设备已准备好，CPU可以进行数据传送。
- **高电平**：表示CPU要访问的存储器或I/O端口已为传送做好准备。
- **无效时**：CPU插入一个或几个等待周期 T_w ，直到READY信号有效为止。



8086/8088CPU外部引脚



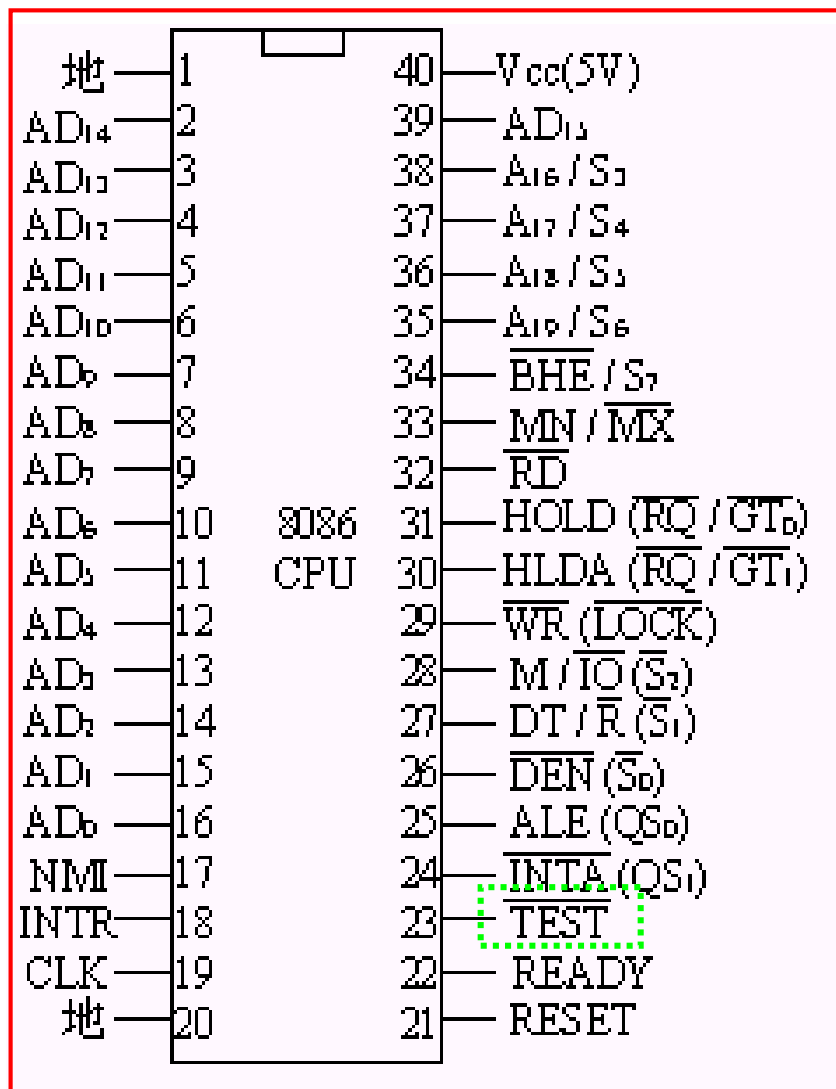
8、 $\overline{\text{TEST}}$

- 测试信号，由外部输入，低电平有效。

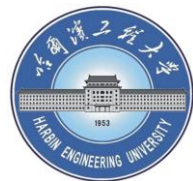
当CPU执行WAIT指令时，每隔5个时钟周期对TEST进行一次测试。

有效：CPU执行下一条指令。

无效：CPU处于等待状态。



8086/8088CPU外部引脚



• 9、INTR (Interrupt Request)

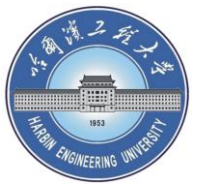
可屏蔽中断请求信号，由外部输入，高电平有效。

高电平：表示外部向CPU发出中断请求。

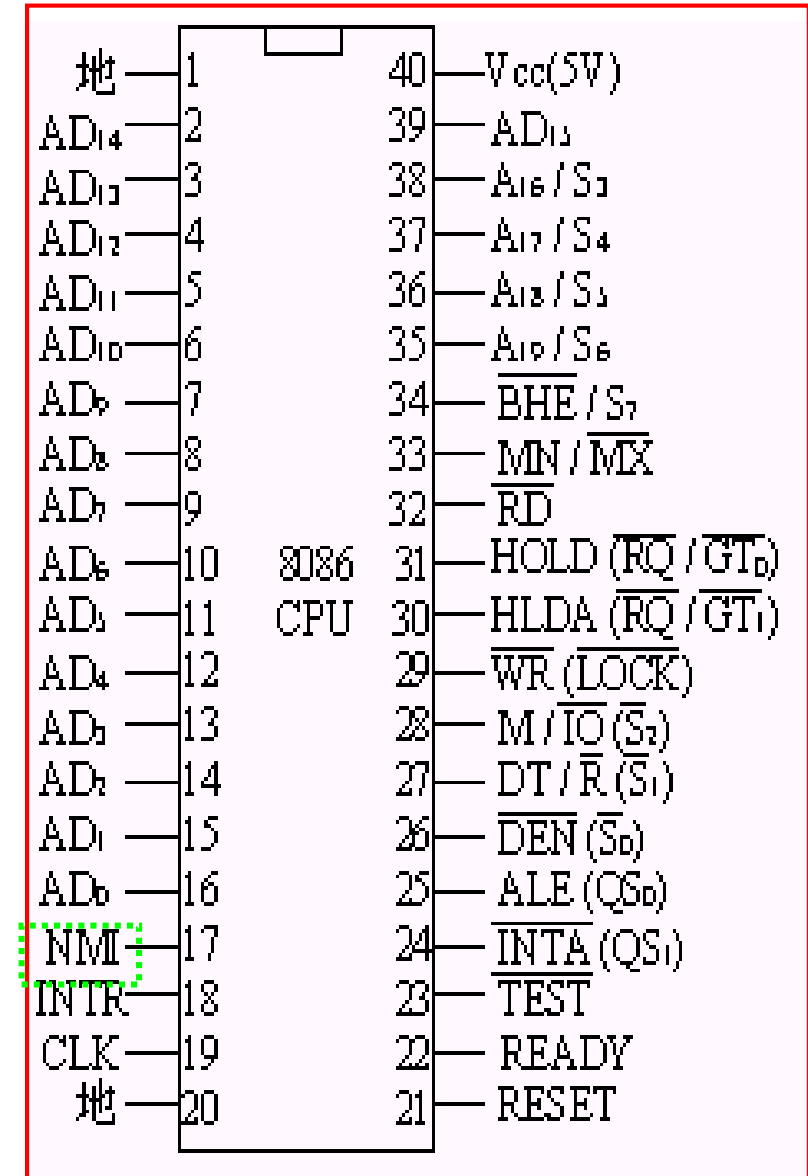
CPU一旦测试到中断请求信号，并且**IF=1**时，暂停正在执行的操作转入中断响应周期。

地	1		40	V _{cc} (5V)
AD ₁₄	2		39	AD ₁₅
AD ₁₃	3		38	A ₁₆ / S ₁
AD ₁₂	4		37	A ₁₇ / S ₄
AD ₁₁	5		36	A ₁₈ / S ₃
AD ₁₀	6		35	A ₁₉ / S ₆
AD ₉	7		34	BHE / S ₇
AD ₈	8		33	MN / MX
AD ₇	9		32	RD
AD ₆	10	8086	31	HOLD (RQ / GT ₀)
AD ₅	11	CPU	30	HLDA (RQ / GT ₁)
AD ₄	12		29	WR (LOCK)
AD ₃	13		28	M / IO (S ₂)
AD ₂	14		27	DT / R (S ₁)
AD ₁	15		26	DEN (S ₀)
AD ₀	16		25	ALE (QS ₀)
NMI	17		24	INTA (QS ₁)
INTR	18		23	TEST
CLK	19		22	READY
地	20		21	RESET

8086/8088CPU外部引脚



•10、NMI(Non-Mackable Interrupt Reguest)
不可屏蔽中断请求信号，CPU一旦测试到NMI有效信号，当前指令执行完后就自动响应中断。

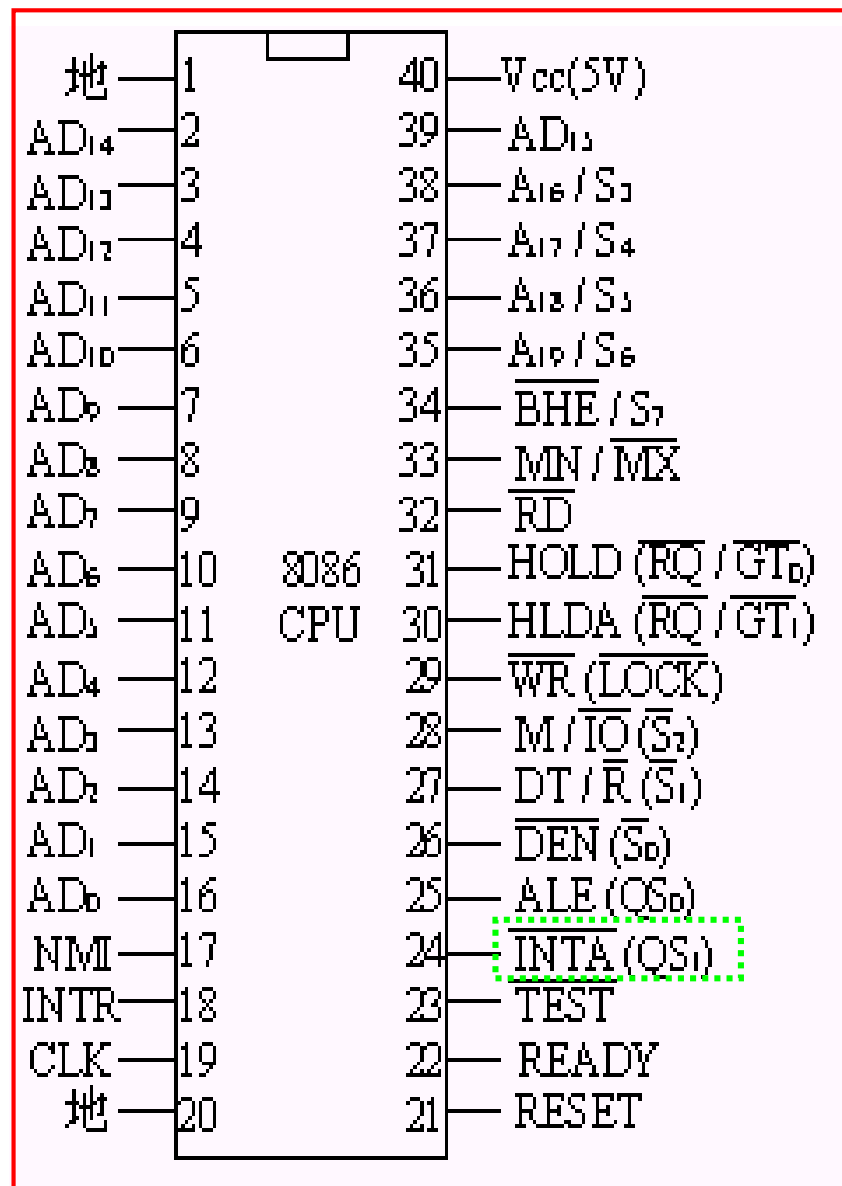


8086/8088CPU外部引脚

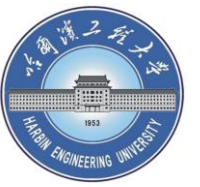


11、 $\overline{\text{INTA}}$ (Interrupt Acknowledge)

- 中断响应信号，向外部输出，低电平有效。
- 表示CPU响应了外部发来的INTR信号。

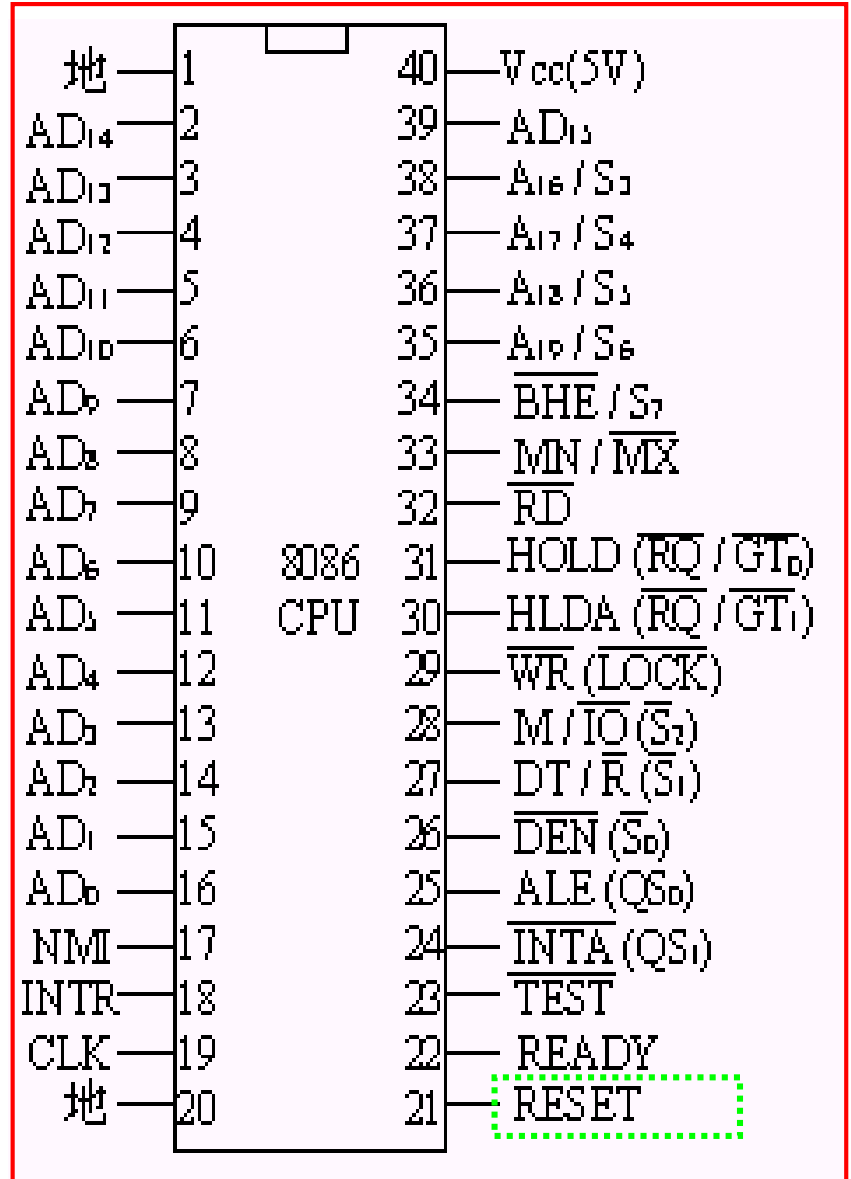


8086/8088CPU外部引脚



12、RESET

- 复位信号，由外部输入，高电平有效
- RESET信号至少保持4个时钟周期。CPU接收到该信号后，停止操作，并将标志寄存器，段寄存器，指令指针IP和指令队列等复位到初始状态。



8086/8088CPU外部引脚

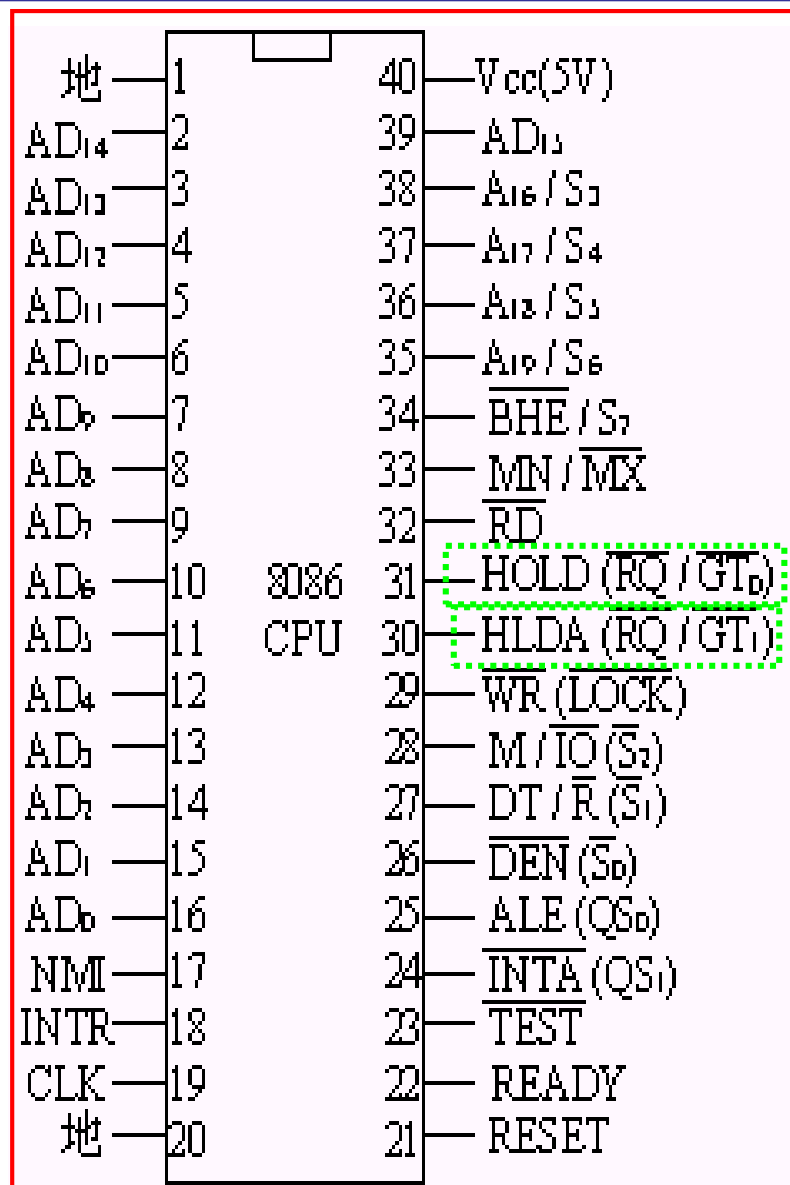


13、HOLD (Hold Request)

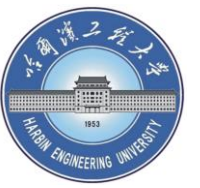
- 总线请求信号，由外部输入，
- 高电平有效。通过此引脚总线的主控者向CPU请求使用总线。

14、HLDA (Hold Acknowledge)

- 总线响应信号，向外部输出，高电平有效。
- CPU一旦测试到HOLD请求时，就在当前总线周期结束时，使HLDA有效，表示响应这一总线请求，并立即让出总线使用权。



8086/8088CPU外部引脚



15、CLK (Clock)

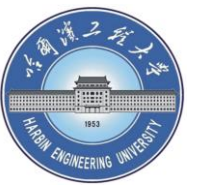
- 主时钟信号，由8284时钟发生器输入。

16、V_{CC} (电源)

- 8086CPU只需要单一的+5V电源，由V_{CC}输入。

地	1		40	V _{CC} (5V)
AD ₁₄	2		39	AD ₁₃
AD ₁₃	3		38	A ₁₆ /S ₃
AD ₁₂	4		37	A ₁₇ /S ₄
AD ₁₁	5		36	A ₁₂ /S ₅
AD ₁₀	6		35	A ₁₀ /S ₆
AD ₉	7		34	BHE/S ₇
AD ₈	8		33	MN/MX
AD ₇	9		32	RD
AD ₆	10	8086	31	HOLD (RQ/GT ₀)
AD ₅	11	CPU	30	HLDA (RQ/GT ₁)
AD ₄	12		29	WR (LOCK)
AD ₃	13		28	M/ \overline{IO} (S ₂)
AD ₂	14		27	DT/ \overline{R} (S ₁)
AD ₁	15		26	\overline{DEN} (S ₀)
AD ₀	16		25	ALE (QS ₀)
NMI	17		24	\overline{INTA} (QS ₁)
INTR	18		23	TEST
CLK	19		22	READY
地	20		21	RESET

8086/8088CPU外部引脚

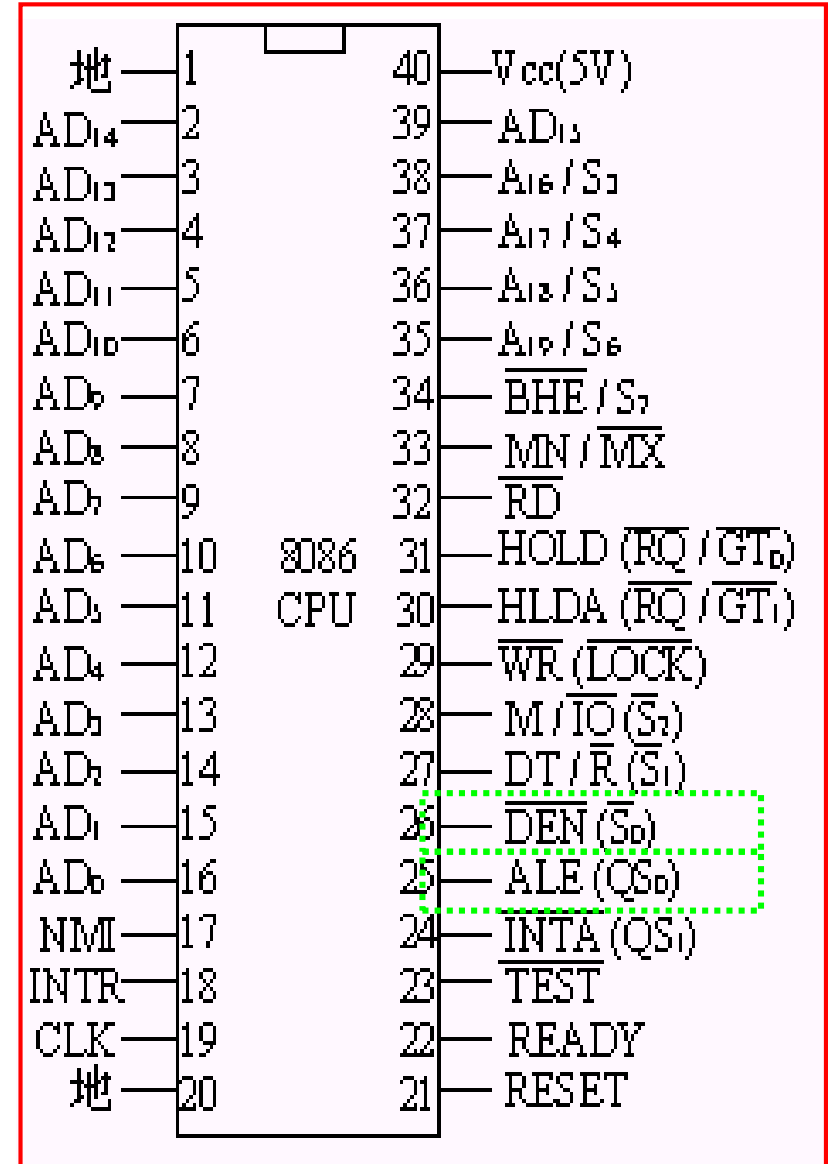


17、ALE (Address Latch Enable)

- **地址锁存允许信号**，向外部输出，高电平有效。当它为**高电平时**，表明CPU地址线上有有效地址。当它为**低电平时**，对地址进行锁存
- 在最小模式系统中用来作地址锁存器8282/8283的**选通信号**

18、 $\overline{\text{DEN}}$ (Data Enable)

- **数据允许信号**，三态输出，低电平有效。当它为低电平时，表明数据线上有有效数据。在最小模式系统中用来用数据收发器8286/8287的**选通信号**。

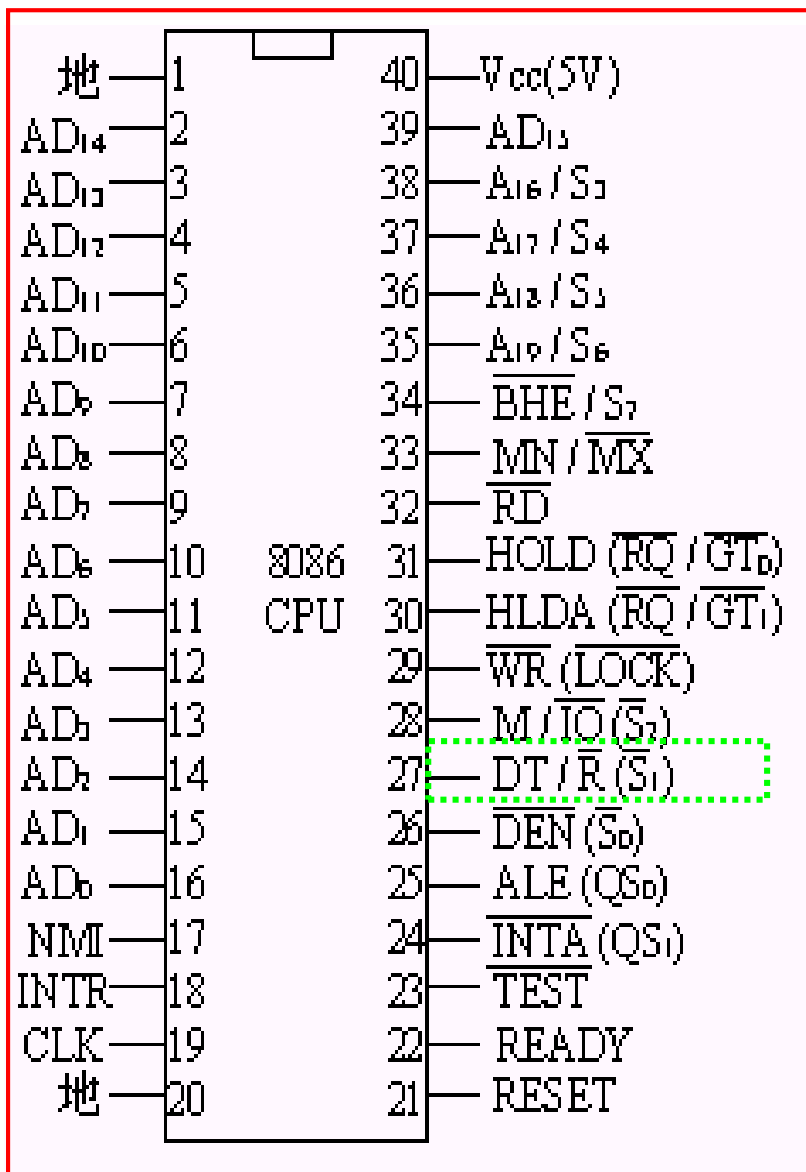


8086/8088CPU外部引脚

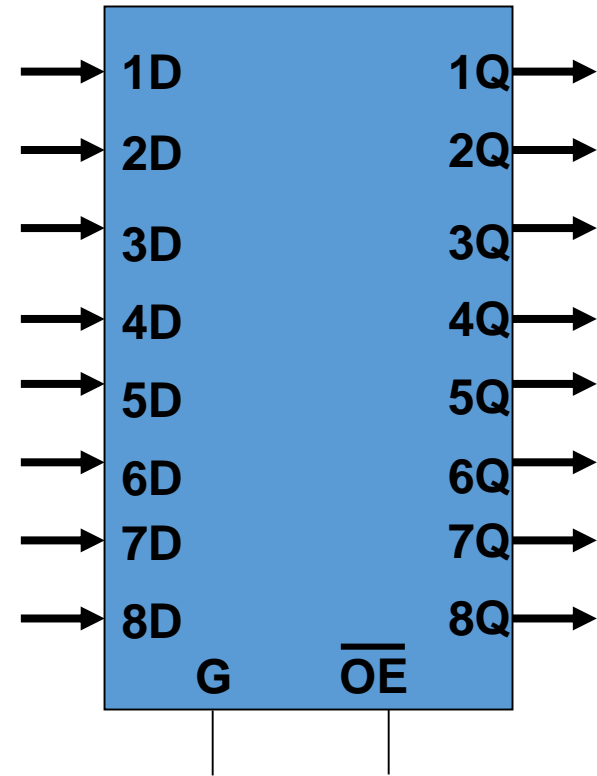
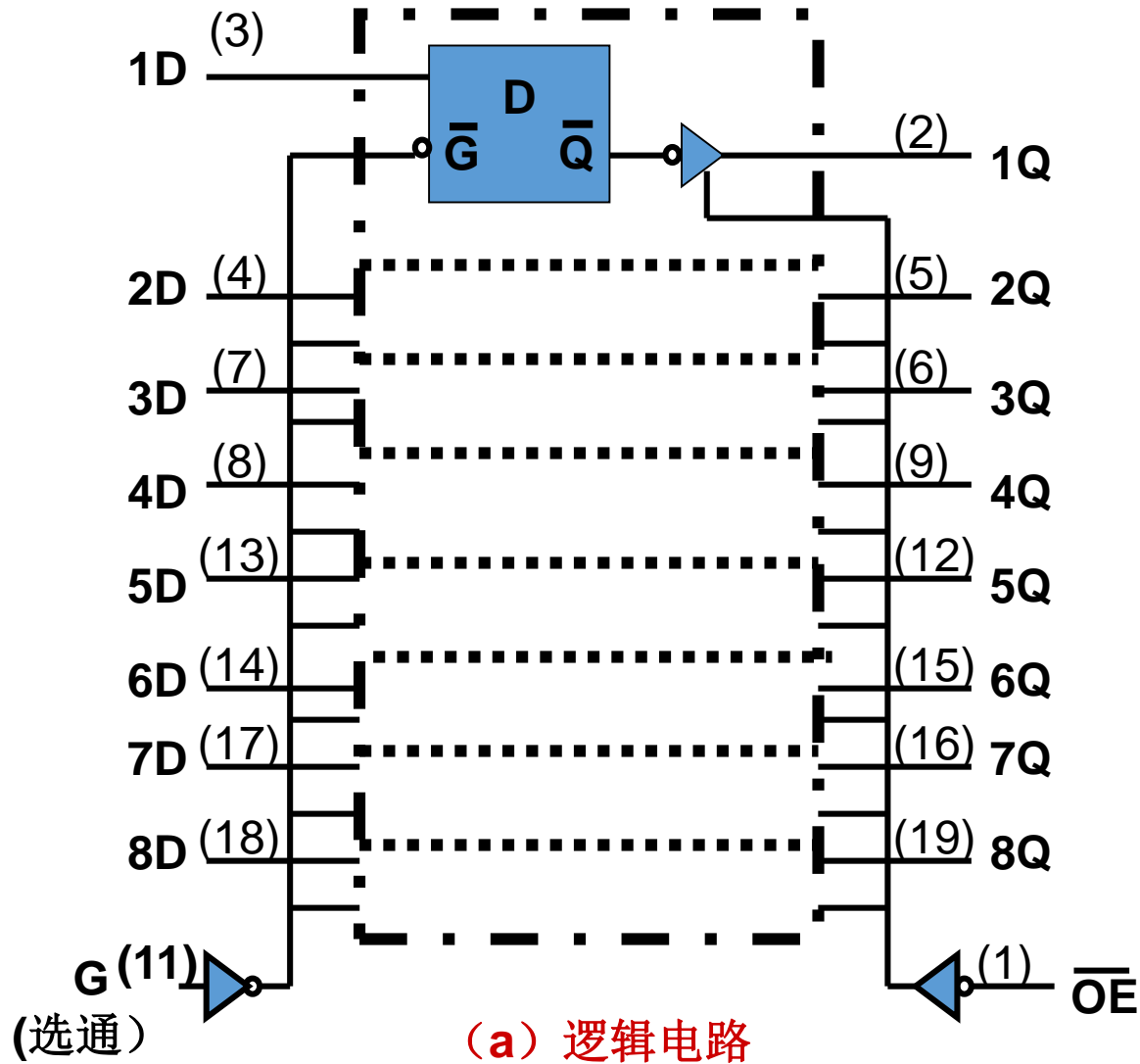
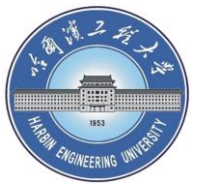


19、DT/R (Data Transmit/Receive)

- 数据发送/接收控制信号，三态输出。
- 在最小模式系统中用来控制数据的传送方向。
- 高电平时，表示数据从CPU向外部输出，即完成写操作。
- 低电平时，表示数据从外部向CPU输入，即完成读操作。

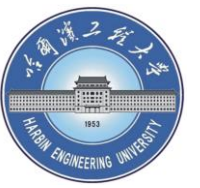


8086/8088CPU外部引脚



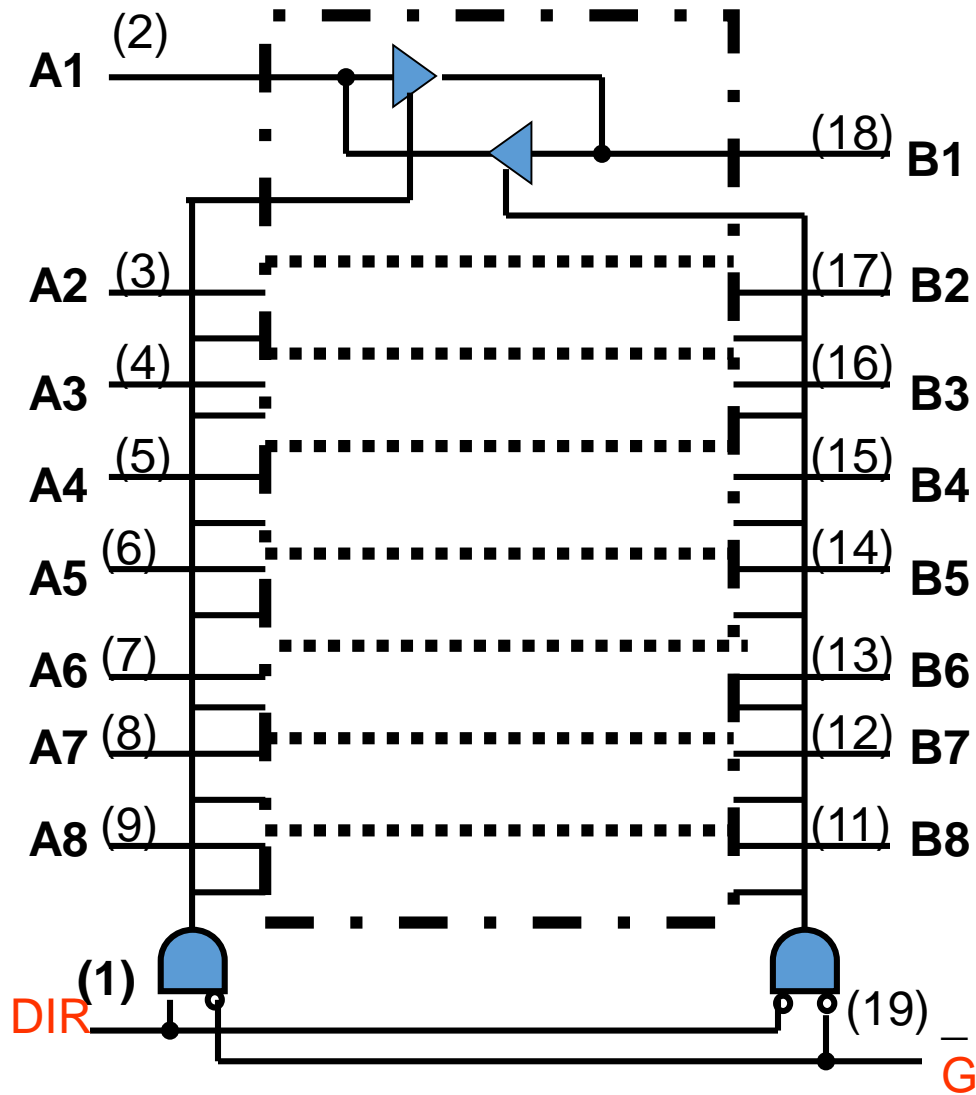
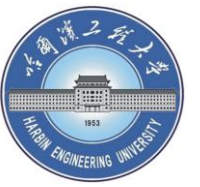
地址锁存器

8086/8088CPU外部引脚

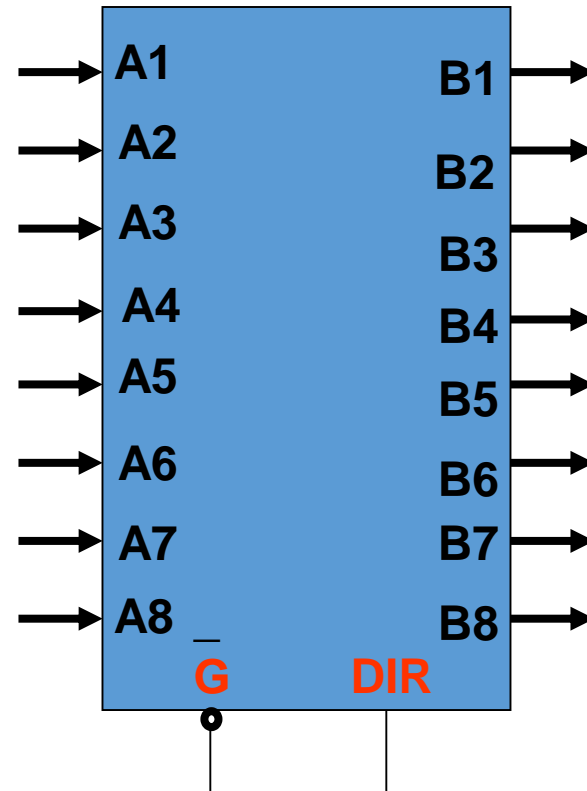


- 由图可见，该电路由8个D锁存器构成(图中仅画一个，其余与此相同)，有8个输入端1D~8D，8个输出端1Q~8Q，两个控制端：选通端G和输出允许端 \overline{OE} 。
- 当选通端G为高电平时，则D锁存器的输出跟随相应输入数据端的逻辑电平；
- 当G变为低电平时，则D锁存器中的当前内容被锁存。
- 只有当输出允许信号 \overline{OE} 有效(为低电平)时，锁存器中的信息才出现在输出端1Q~8Q上；如果 \overline{OE} 为高电平，则输出处于高阻态。

8086/8088CPU外部引脚



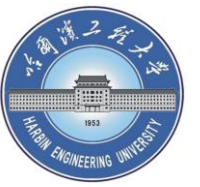
(a) 逻辑电路



(b) 引脚图

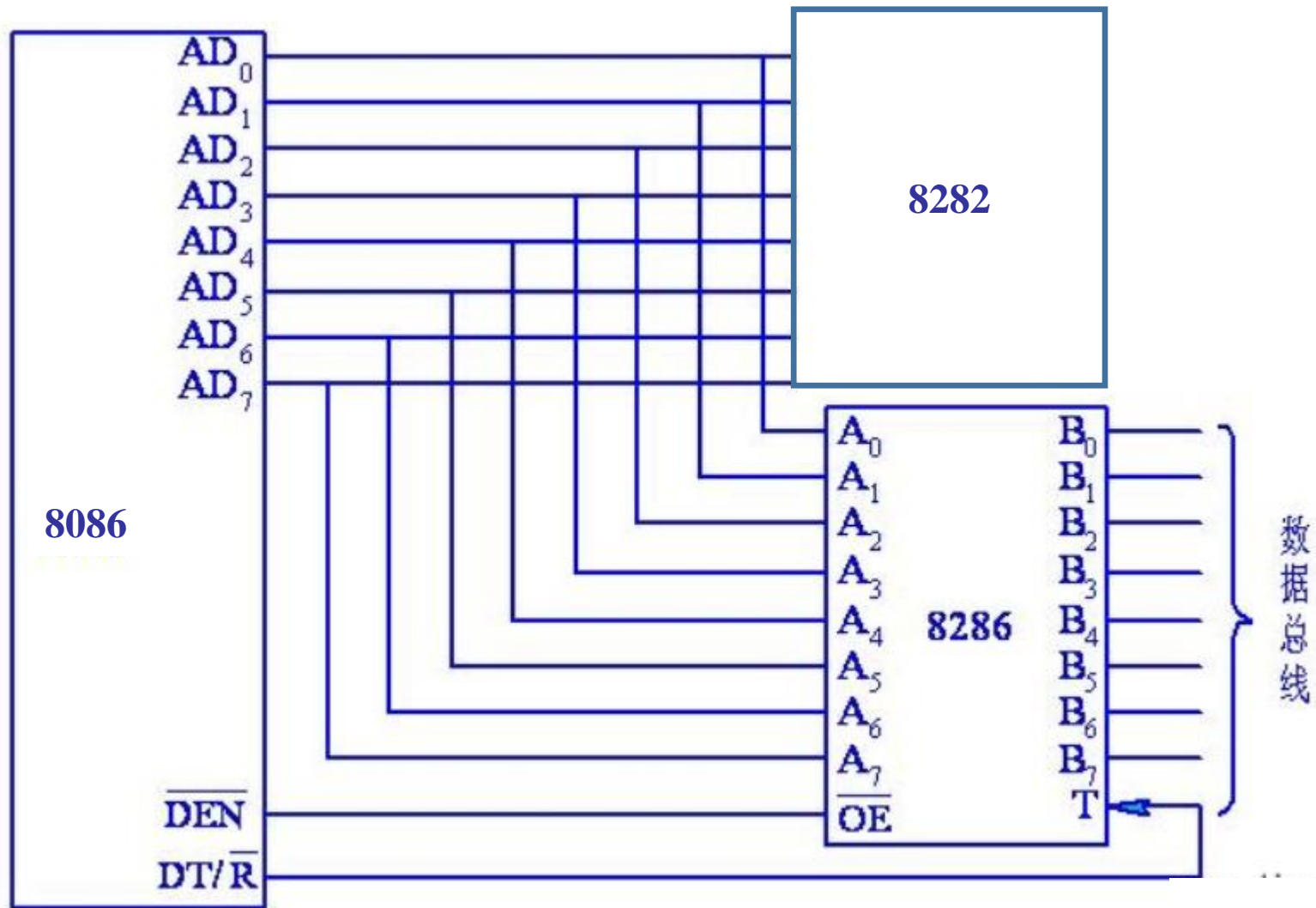
数据总线收发器

8086/8088CPU外部引脚



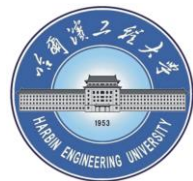
- 该数据收发器有16个双向传输的数据端，即A1~A8和B1~B8；另有两个控制端，即允许端G和方向控制端DIR。G用于允许该收发器的操作，DIR用于控制数据传送的方向(A→B或B→A)。
- 由图可见，若G信号无效(为高电平)，则无论DIR为何种电平，下面两个“与门”的输出均为低电平，从而使两个方向上的三态门的输出均为高阻态，收发器处于“隔开”状态，即两个方向上的数据传送均不能进行。

8086/8088CPU外部引脚



8286数据收发器

8086/8088CPU外部引脚

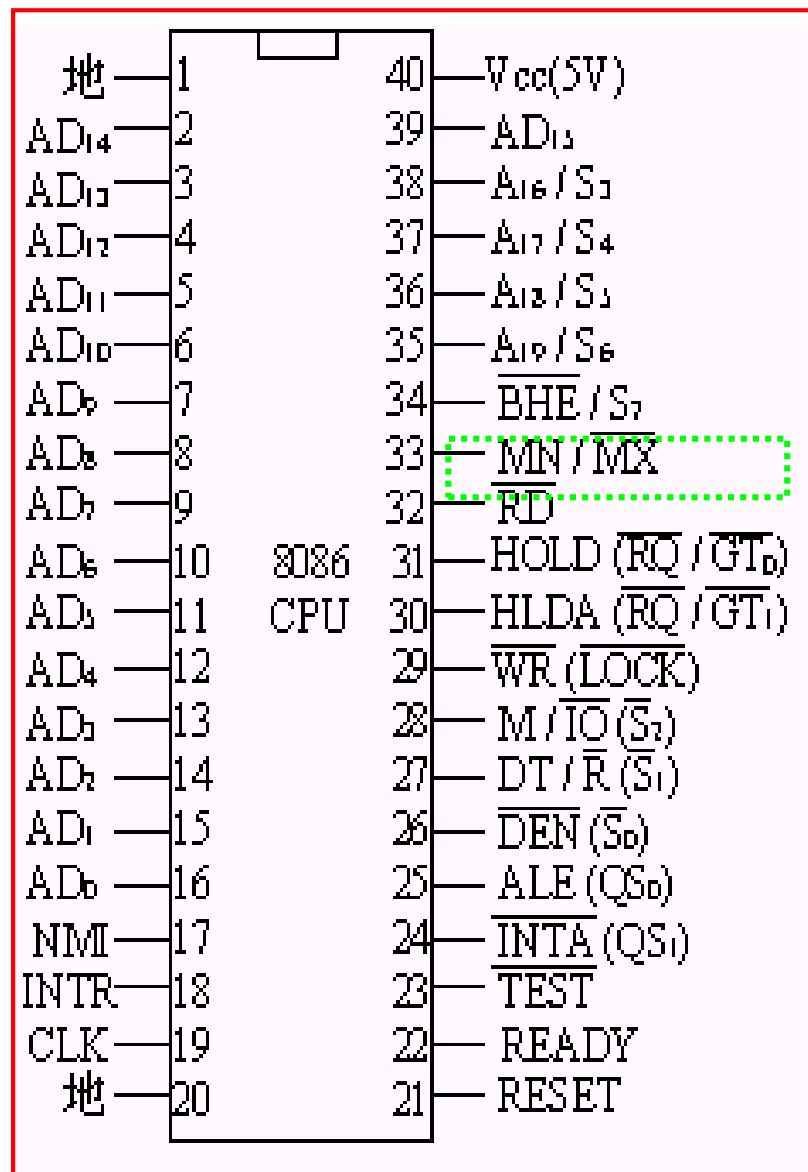


• 最小模式

- **MN/MX接+5V**
- 构成小规模的应用系统，只有8086一个微处理器。
- 所有的总线控制信号均为8086产生，系统中的总线控制逻辑电路，减少到最少。

• 最大模式

- **MN/MX接地。**
- 用于大型（中型）8086/8088系统中，系统总是包含有两个或多个微处理器，其中一个主处理器就是8086或8088，其它的处理器称协处理器，协助主处理器工作。
- 需要**总线控制器**来变换和组合控制信号。

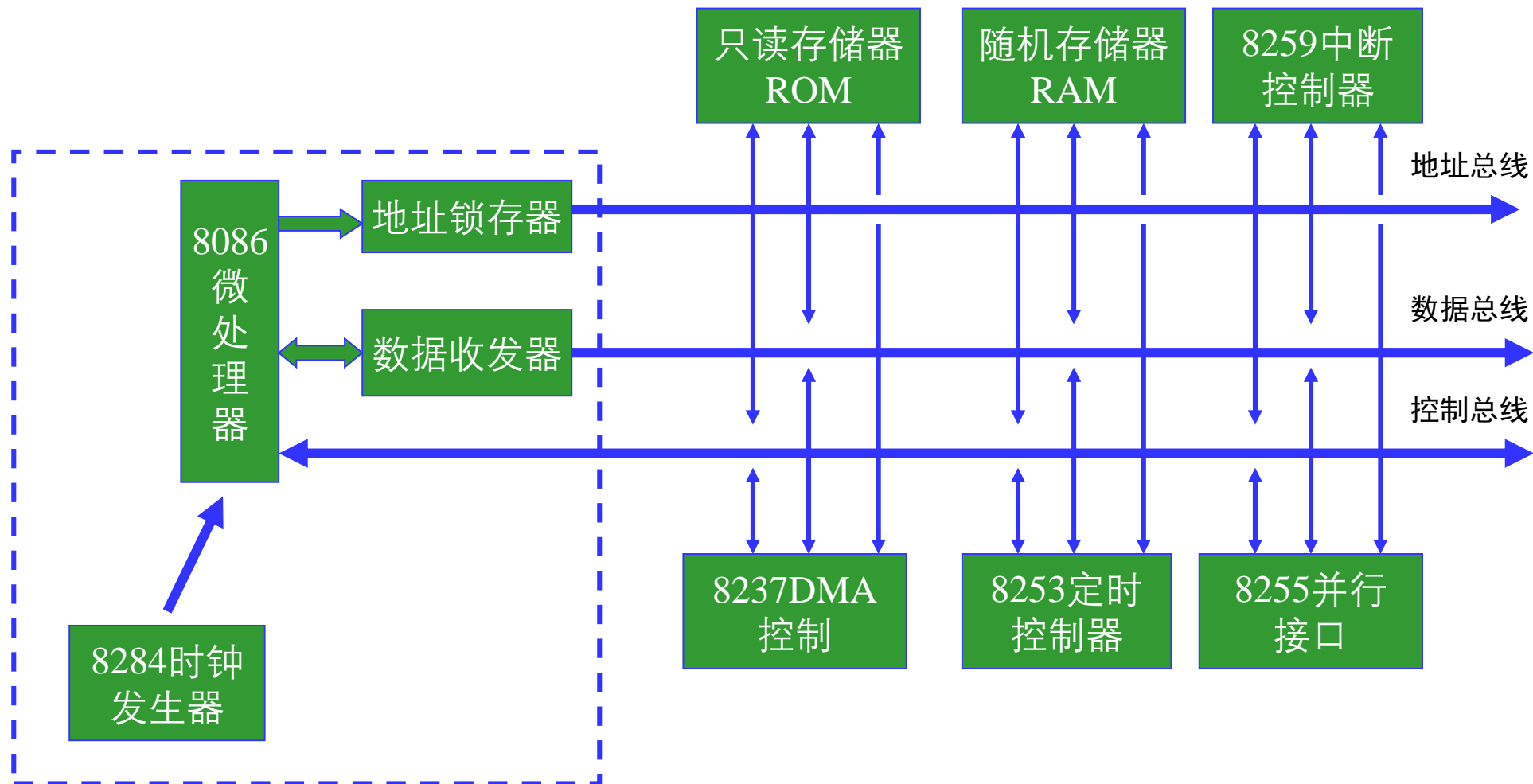
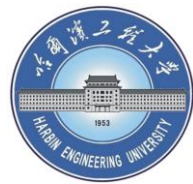




8086/8088CPU外部引脚

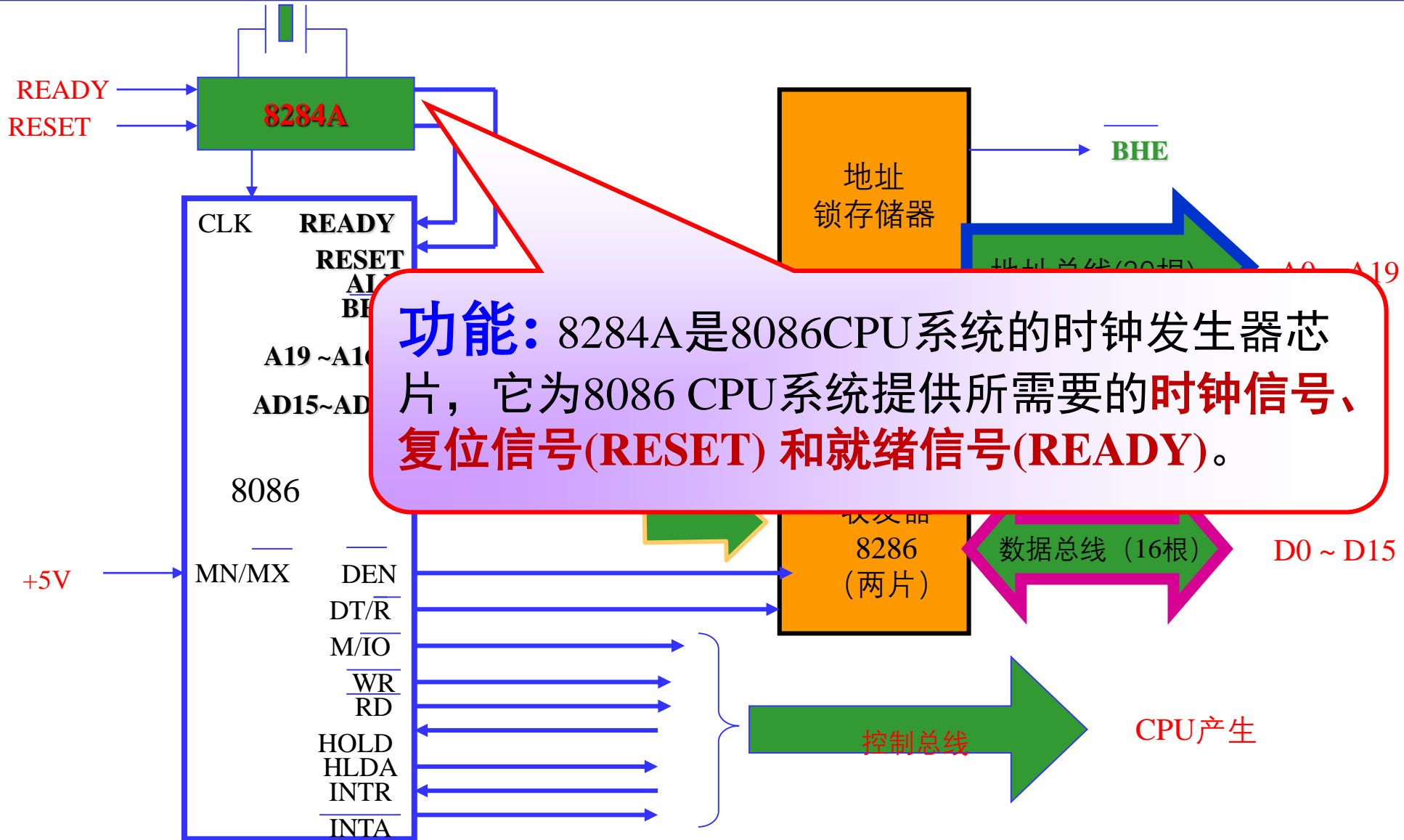
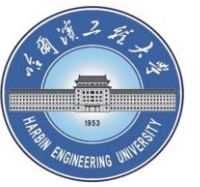
- 最小工作模式是一个以8086为主体的**单处理器系统**，所有控制信号均由CPU直接提供。
- 最小工作模式系统包括硬件如下：
 - 一片时钟发生器8284A；
 - 三片地址锁存器8282；
 - 两片数据收发器8286/8287；

8086/8088CPU外部引脚

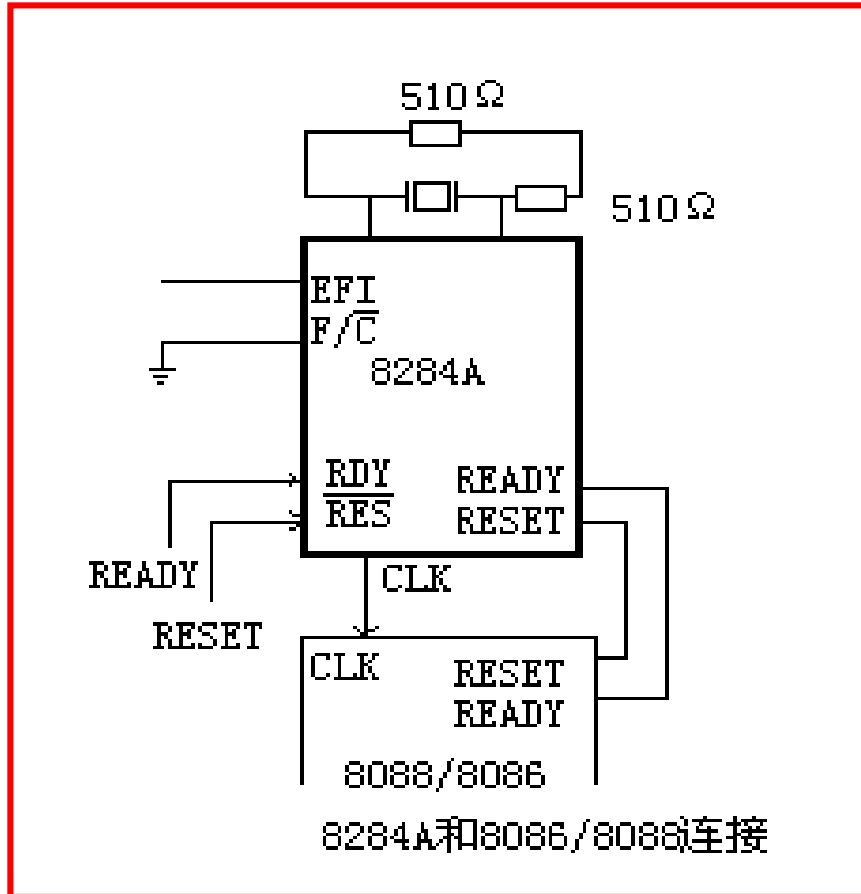
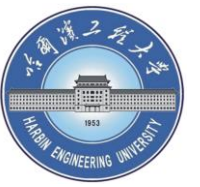


CPU系统

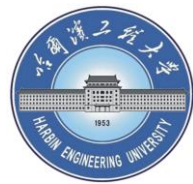
8086/8088CPU外部引脚



8086/8088CPU外部引脚



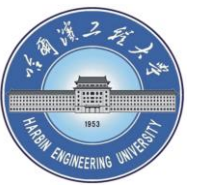
标志寄存器	清零
IP	0000H
CS	FFFFH
DS	0000H
SS	0000H
ES	0000H
指令队列	空
其它寄存器	0000H



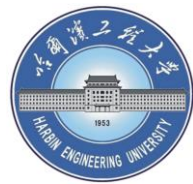
8086/8088CPU外部引脚

- 8088的外部数据总线只有**8位**，**内部总线也是16位**，称之为**准16位CPU**；
- 8088的低8位地址与数据复用， $AD_7—AD_0$ ，高8位地址为专用引脚， $A_{15}—A_8$ ；
- 8088的该引脚定义为 $\overline{IO/M}$ ，电平属性与8086的 $\overline{M/IO}$ 相反，为与8085CPU兼容；
- 8088只有8位数线，不需要 \overline{BHE} 信号

第 2 章 8086微处理器



- 2-1 8086 CPU结构
- 2-2 8086 CPU的引脚及其功能
- **2-3 8086 存储器组织**
- 2-4 8086 系统配置
- 2-5 8086 CPU时序



2-3 8086 存储器组织

8086存储器分体结构：

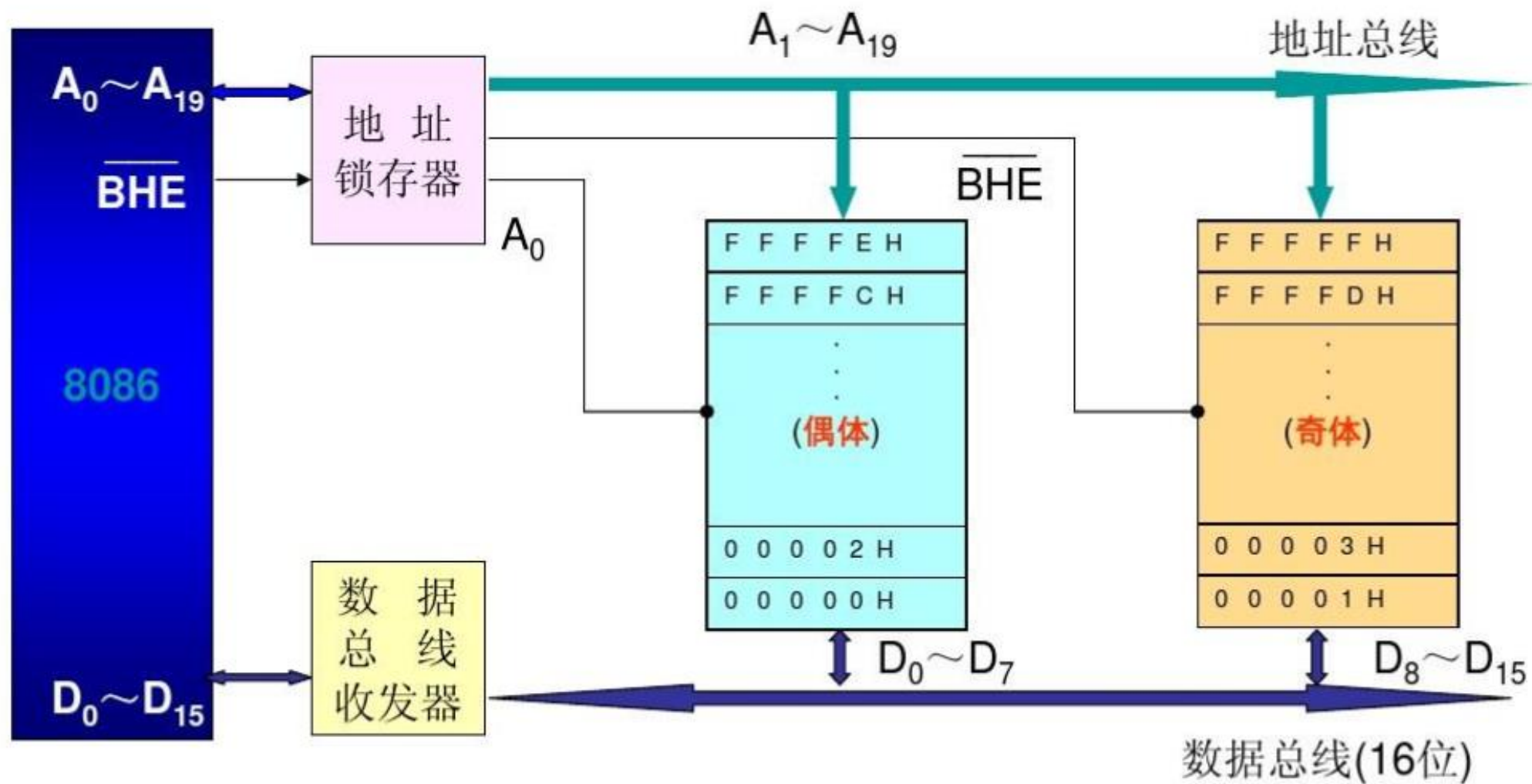
- 常用的存储器芯片是字节结构，只有8位数据线，8086有16位数据线。

如何用8位的存储器芯片构成16位的存储器？？

1MB空间分为两个512KB的存储体：

- 偶地址存储体：连接D7~D0，A0=0时选通；
- 奇地址存储体：连接D15~D8，BHE=0时选通。
- 8086可以传送一个字节，也可以传送二个字节。

2-3 8086 存储器组织





2-3 8086 存储器组织

存储器分体结构注意要点:

- **BHE**控制连接CPU高8位的存储体——奇存储体的片选信号
- **A0**控制连接CPU低8位的存储体——偶存储体的片选信号
- CPU的地址线A1~A19连接两个存储体的地址信号A0~A18



2-3 8086 存储器组织

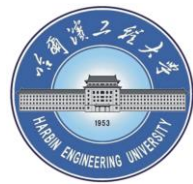
存储器分体结构注意要点：

- 一个字数据包含地址相邻的两个字节，一个字节放在奇存储体中，另一个字节放在偶存储体中；
- **字数据的地址定义为其低字节所在单元的地址**，可以是偶地址字数据（对准存放），也可以是奇地址字数据；
- 8088只有8位数据总线，不存在分体结构。

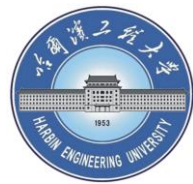
8086 总线操作列表

操作	$\overline{\text{BHE}}$	A0	总线使用
从偶地址开始读写一个字节	1	0	AD7~AD0
从奇地址开始读写一个字节	0	1	AD15~AD8
从偶地址开始读写一个字，一次总线操作就完成	0	0	AD15~AD0
从奇地址开始读写一个字，分两次进行	0 1	1 0	AD15~AD8 AD7~AD0

第 2 章 8086微处理器



- 2-1 8086 CPU结构
- 2-2 8086 CPU的引脚及其功能
- 2-3 8086 存储器组织
- **2-4 8086 系统配置**
- 2-5 8086 CPU时序



2-4 8086 系统配置

一、最小模式系统

8086在最小模式下的典型配置：

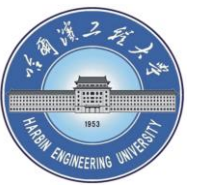
1、 $\overline{MN}/\overline{MX}$ 接+5v

2、一片8284作为时钟发生器；

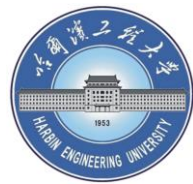
3、三片8282或74LS373作地址锁存器；

4、二片8286/8287作总线驱动器，8286是同相输出，8287是反相输出；

第 2 章 8086微处理器



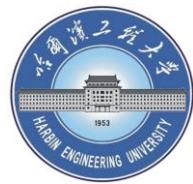
- 2-1 8086 CPU结构
- 2-2 8086 CPU的引脚及其功能
- 2-3 8086 存储器组织
- **2-4 8086 系统配置**
- 2-5 8086 CPU时序



2-5 8086的操作和时序

• 主要操作

- ① 系统复位与启动
- ② 暂停
- ③ 总线读/写操作
- ④ 总线保持
- ⑤ 中断操作



一、系统复位与启动

- 通过RESET引脚上的触发信号来执行。

标志寄存器：	清零
指令指针（IP）：	0000H
CS：	FFFFH
DS、ES、SS：	0000H
指令队列：	空
其它寄存器：	0000H



一、系统复位与启动

- 复位后，第一条指令的地址：CS左移四位为FFFF0H。

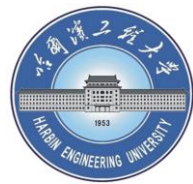
物理地址为 $\text{FFFF0} + \text{0000H}$ (IP中)
 $= \text{FFFF0H}$

- 一般在FFFFF0中，存放一条段交叉直接JMP指令，转移到系统程序实际开始处。



第 2 章 8086微处理器

- 2-1 8086 CPU结构
- 2-2 8086 CPU的引脚及其功能
- 2-3 8086 存储器组织
- 2-4 8086 系统配置
- 2-5 8086 CPU时序



最小模式下的总线操作

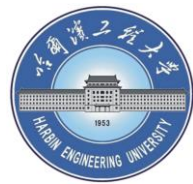
与时间有关的几个概念

时钟周期：CPU的CLK引脚所输入时钟信号的周期长度，**是计算机系统的最小时间单位**，常称为**T状态**；

总线周期：BIU完成一次总线操作，即访问一次存储器或I/O端口所需要的时间，**一个基本总线周期包含4个T状态**；

机器周期：CPU完成一个基本操作所需要的时间，通常**包含若干个时钟周期**。

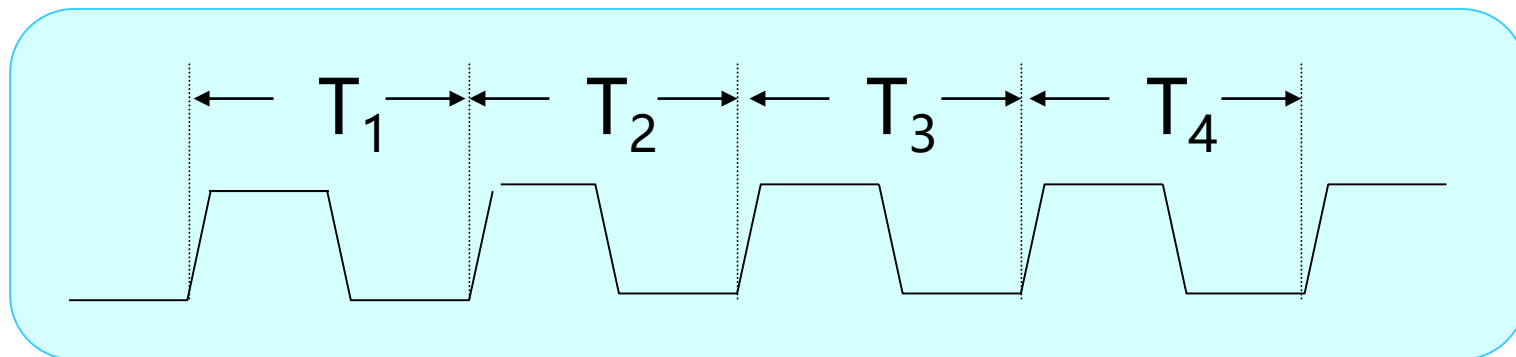
指令周期：CPU取出并执行一条指令所需要的时间，不同指令的指令周期长短一般是不同的，通常**包含若干个机器周期**；

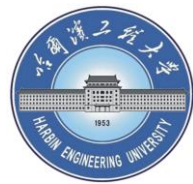


最小模式下的总线操作

1. 时钟周期 (T状态周期Clock Cycle)

微处理器内部操作的最基本时钟单元，宽度为时钟信号相邻两上升(下降)沿之间的时钟间隔。

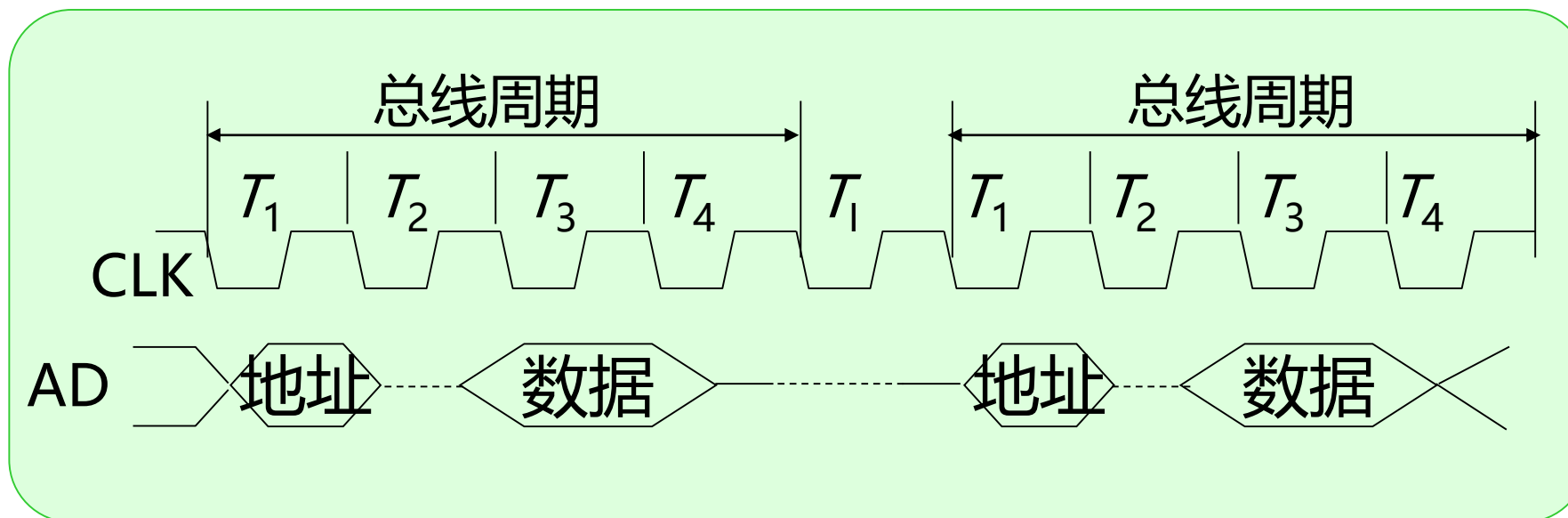


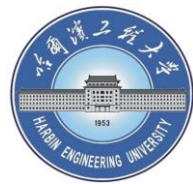


二、最小模式下的总线操作

2. 总线周期 (Bus Cycle)

微机处理器BIU与外部电路之间进行一次数据传送操作所占用的时间，包含若干个时钟周期。



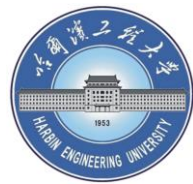


二、最小模式下的总线操作

3.指令周期(Instruction Cycle)

执行一条指令所需要的时间，**由一至若干个机器周期组成。**

机器周期与总线周期的区别？？



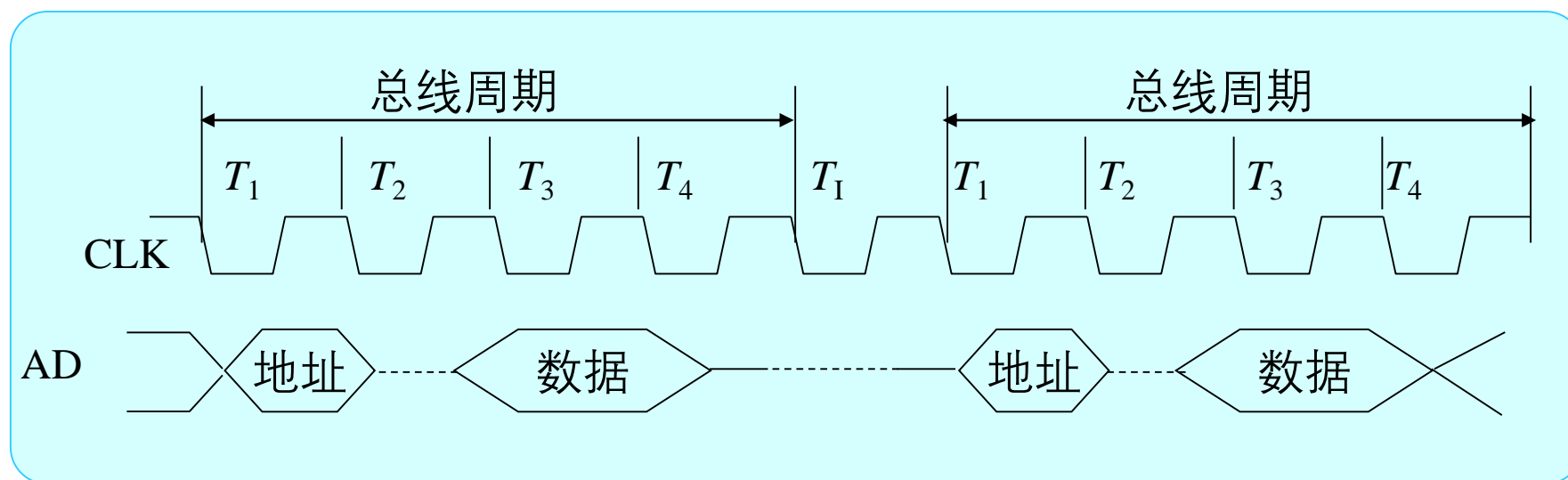
二、最小模式下的总线操作

几个周期的区别：

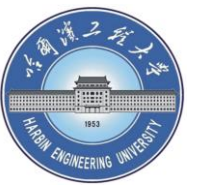
- ① **时钟周期**是最小单位 (CPU晶振工作频率的倒数)。
- ② **机器周期**需要1个或多个时钟周期。
- ③ **指令周期**需要1个或多个机器周期。
- ④ **机器周期**因涉及一个基本操作时间, 可能**操作总线**, 因此可能会包含**总线周期**, 也可能不包含。

总线操作与时序

Intel 8086微处理器采用**总线复用**操作方式，8086的16位数据总线和地址总线的**低16位**是**共用**的，典型的总线周期如图。

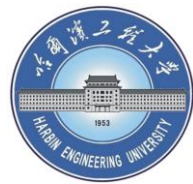


基本概念

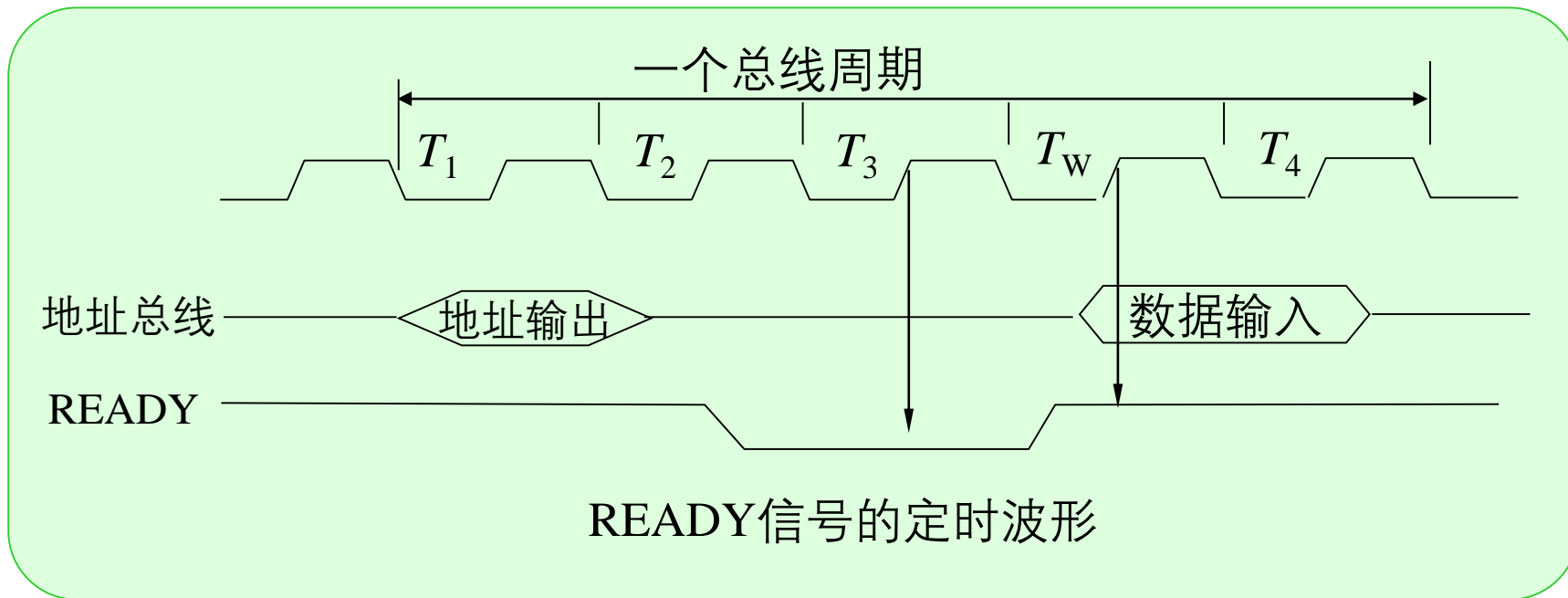


- 在没有插入等待时钟周期 T_w 的情况下，总线周期由4个时钟周期组成，即图中 T_1 、 T_2 、 T_3 、 T_4
- 在 T_1 期间CPU把存储器或外设的地址放到总线上，这些地址信息由**ALE**控制锁存到地址锁存器中，以便使总线上可以传送数据信息。
- T_2 期间 $AD_{15} \sim AD_0$ 上的地址已消失，在数据还未到达前，分时复用的地址/数据总线处于高阻态，以便为读入或写出数据作准备。
- 在 T_3 和 T_4 期间，读或写的数据出现在总线上，以使完成读或写的操作。

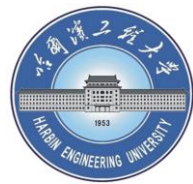
基本概念



等待周期 T_w :



如果在 **T_3 周期结束之前**，存储器或外设未准备好数据传送，就要启动输入CPU的**READY线**使之变低电平，从而在 **T_3 和 T_4 之间**插入一个或多个 **T_w 等待周期**，直到**READY变高**，转入 **T_4 周期**，完成读写操作。

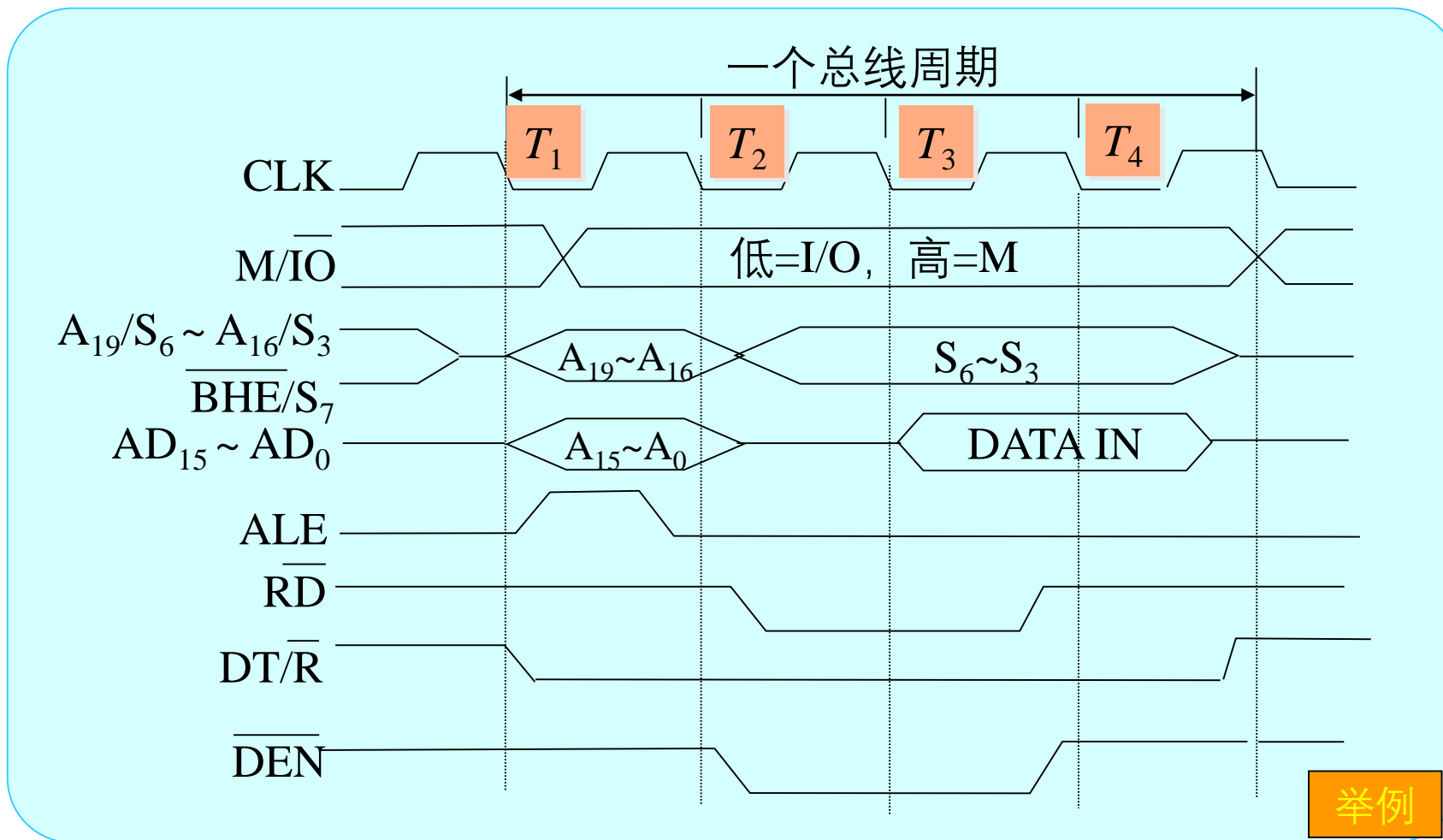


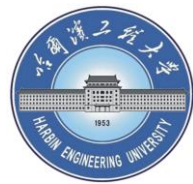
基本概念

空闲状态周期 T_I :

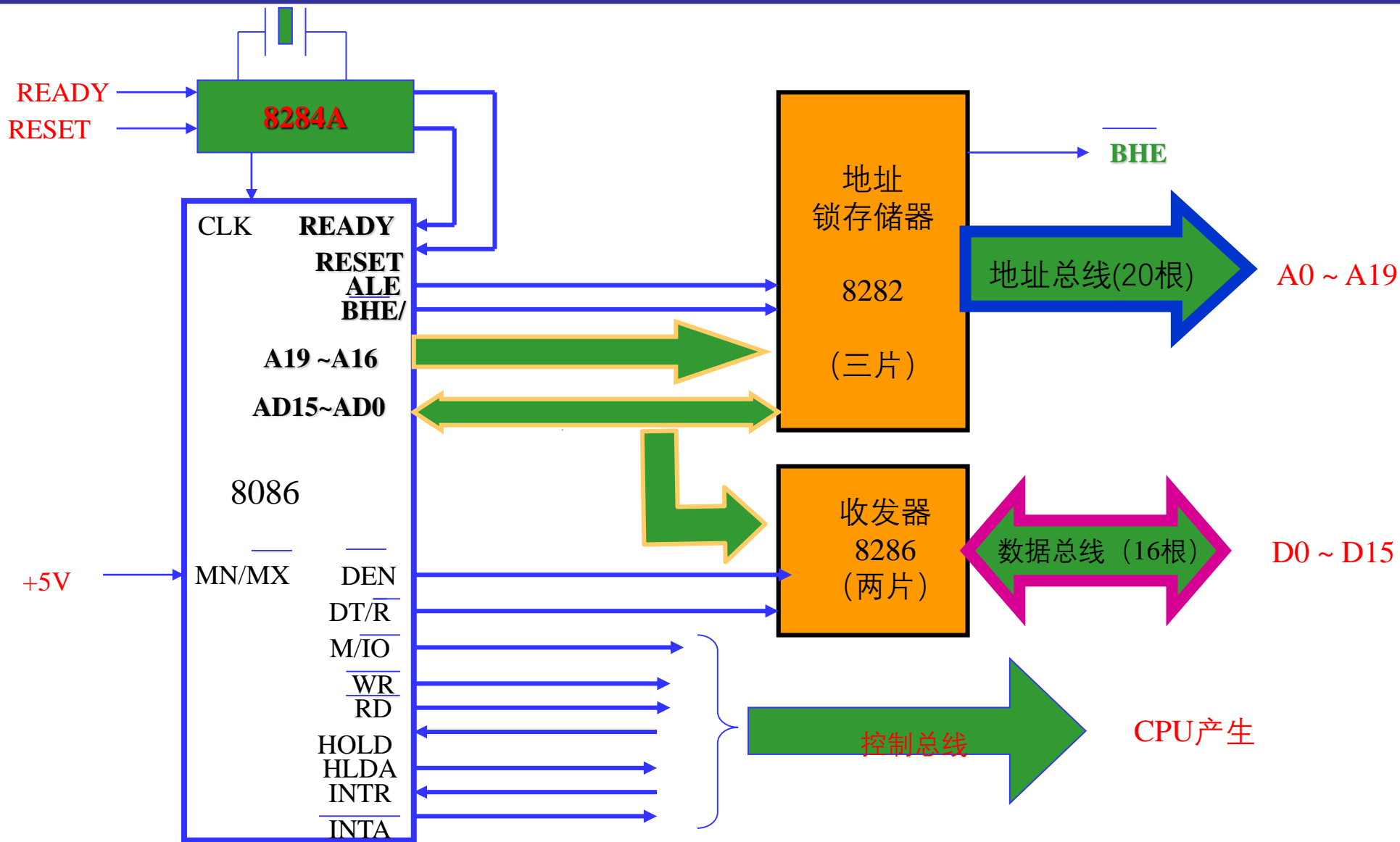
8086**执行部件EU和总线接口部件BIU在一定程度上独立并行工作**，只有当BIU为了填满指令队列或EU执行指令需要与外部交换数据时，申请一个总线周期，此时BIU才执行一个总线周期。因此在两个总线周期之间，**可能会存在一些BIU空闲时钟周期。**

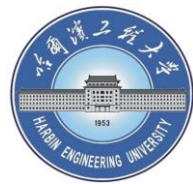
一、8086读总线周期



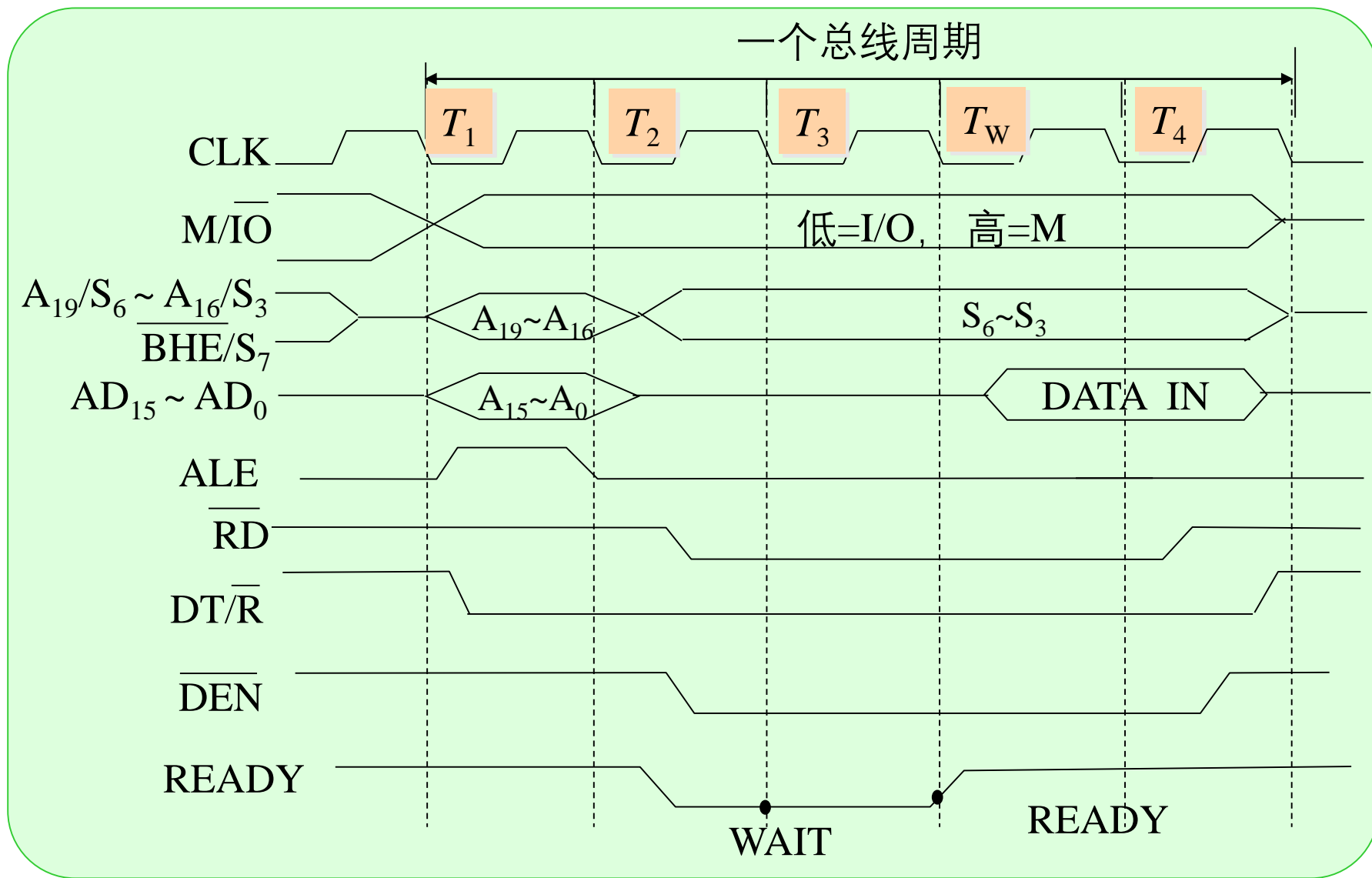


一、8086读总线周期

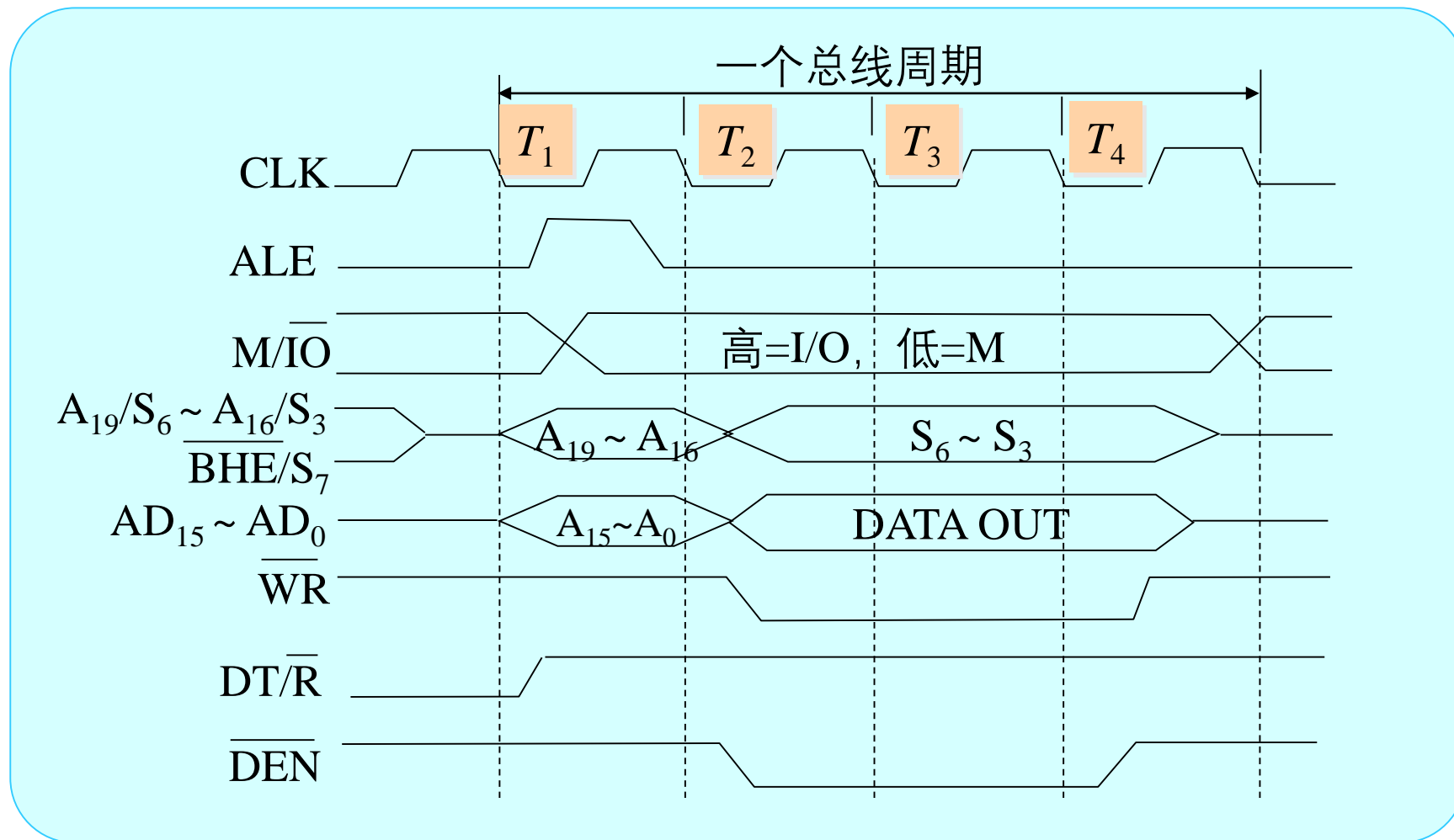




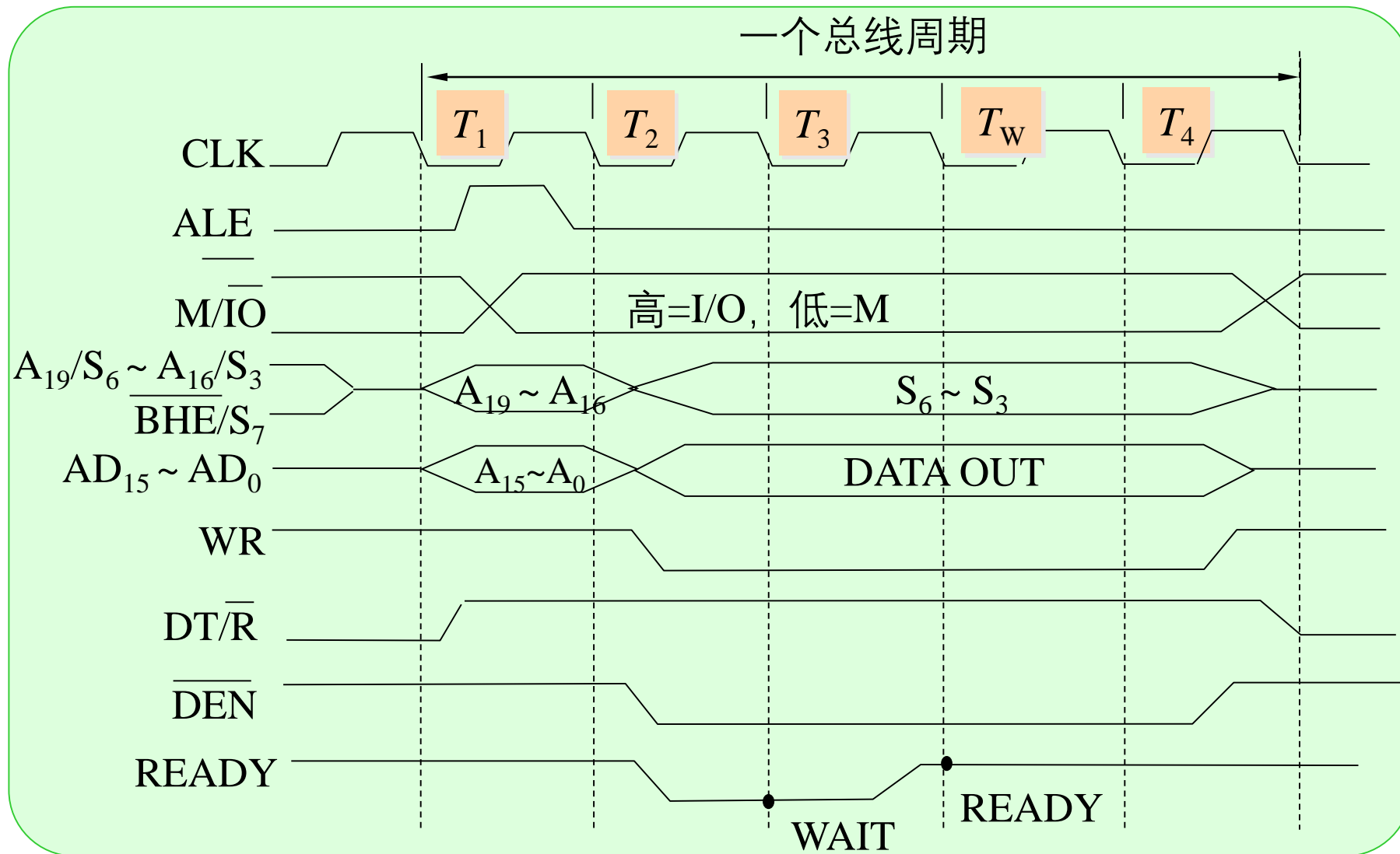
二、8086具有等待状态的读总线周期

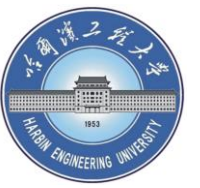


三、8086写总线周期



四、8086具有等待状态的写总线周期





四、最小模式下的总线保持

- 系统有多个总线主模块，CPU以外的其他总线主模块为了获得对总线的控制，需向CPU发出使用总线的请求；
- CPU如果同意让出总线，要向其他总线主模块发出应答信号。

HOLD: 总线保持请求信号

HLDA: 总线保持回答信号



第二章 小结

一、8086CPU内部结构

(一) 总线接口单元BIU:

功能：控制存储器与I/O端口与CPU交换信息。

1. 段寄存器:

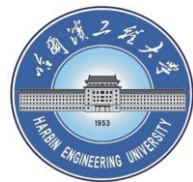
CS (代码段) 控制程序区

DS (数据段) 控制数据区

ES (附加段) 控制数据区

SS (堆栈段) 控制堆栈区

第二章 小结

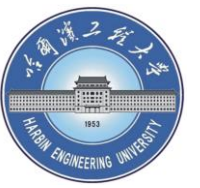


2. 16位指令指针寄存器IP

IP: 其中存放的是**下一条指令**的地址偏移量。

将CS左移四位 + IP内容 → 指令代码的**实际物理地址**。

第二章 小结



(二) 执行单元EU

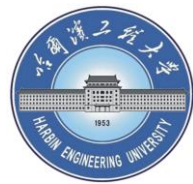
16位数据寄存器 : AX、BX、CX、DX

16位堆栈指针寄存器 : SP (存放的是堆栈操作地址偏移量)。

16位基址寄存器BP: 堆栈栈底指针, 用来定位栈中存储单元。

16位变址寄存器SI: 担任源地址, 目的地址的偏移量。

16位变址寄存器DI: 担任目的地址, 目的地址的偏移量。



第二章 小结

(三) 算数逻辑单元ALU

(四) 状态标志寄存器

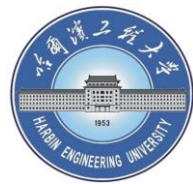
二、8086的引脚

最小模式 MN/MX为高电平，接VCC

最大模式 MN/MX接地

三、8086CPU的时序：总线读、写周期

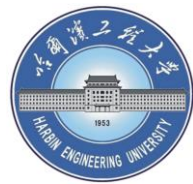
四、复位



第二章 习题

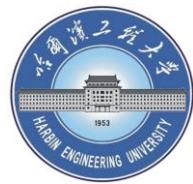
填空题

- 1、8086 CPU内部结构按功能分为两部分，即_____和_____。
- 2、8086是_____位的微处理器，其内部数据通路为_____位。其对外数据总线为_____位；8088内部数据通路为_____位，其对外数据总线为_____位。
- 3、8086中执行部件的功能是负责_____的执行。
- 4、8086 CPU中的总线接口部件BIU，根据执行部件EU的要求，完成与_____或_____间的数据传送。



第二章 习题

- 5、当对堆栈操作时，8086会自动选择_____值作为段基值，再加上由_____提供的偏移量形成物理地址。
- 6、8086总线接口部件中有四个段寄存器，它们分别是_____、_____、_____和_____。
- 7、根据功能不同，8086的标志位可分为_____标志和_____标志。
- 8、8086/8088构成的微机中，每个主存单元对应两种地址：即_____和_____。



第二章 习题

- 9、CPU访问存储器进行读写操作时，通常在____状态去检测READY。
- 10、8086/8088CPU的数据线和地址线是以____方式轮流使用的。
- 11、在一个基本总线周期，8086引脚 $A_{19}S_6 \sim A_{16}/S_3$ 在____输出地址信息，在____输出状态信息。
- 12、在一个总线周期，8086CPU要完成与外设或存储器进行16位数据的交换，此时引脚信号 $BHE = \underline{\hspace{2cm}}$ ， $A_0 = \underline{\hspace{2cm}}$ 。
- 13、8086CPU从____引脚接收可中断请求信号，若标志位 $IF=1$ ，则CPU在结束当前指令后，从____引脚发出中断响应信号。



第二章 习题

选择题：

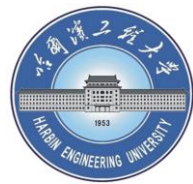
1、下面关于PC机CPU的叙述中，不正确的是___。

- A. 为了暂存中间结果，CPU中包含几十个甚至上百个寄存器，用来临时存放数据
- B. CPU是PC机中不可缺少的组成部分，它担负着运行系统软件和应用软件的任务
- C. 所有PC机的CPU都具有相同的机器指令
- D. CPU至少包含1个处理器，为了提高计算速度，CPU也可以由2个、4个、8个甚至更多个处理器组成



第二章 习题

- 3、8086有两种工作方式，当8086处于最小方式时，MN/MX接()
A.+12V B.-12V C.+5V D.地
- 4、8086CPU芯片的时序中，不加等待的一个总线周期需时钟周期数为()
A. 1 B. 2 C. 3 D. 4
- 5.PC机开机后执行的第一条指令的地址应该是()。
A.FFFFFFFH B.F0000H C.00000H D.FFFF0H



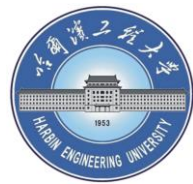
第二章 习题

6、8088/8086内部能够计算出访问内存储器的20位物理地址的附加机构是（ ）。

A.地址加法器 B.ALU C.指令队列 D.段寄存器

7、工作在最小模式下，8086CPU芯片的时序中，将地址信号锁存的信号是（ ）

A.DT/R B.DEN C.ALE D.AEN



第二章 习题

8、8086有20根地址线，直接寻址空间为_____。

A.64 MB B.1 MB C.512KB D.8 MB

9、8086有4个数据寄存器，其中AX除用做通用寄存器外，还可用做_____。

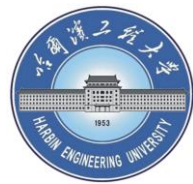
A.累加器 B.计数器 C.基址寄存器 D.段寄存器

10、微处理器8086字符串操作中，用来存放源串偏移地址的寄存器是_____。

A.BP B.SP C.SI D.DI

11、微处理器8086存放当前数据段地址的寄存器是_____。

A.CS B.DS C.ES D.SS



第二章 习题

12、8086CPU中的SP寄存器是一个（ ）位的寄存器

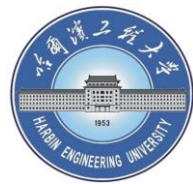
A. 8 B. 16 C. 20 D. 24

13、8086微处理器要求RESET有效维持时间至少要有（ ）。

A. 5T B. 4T C. 3T D. 2T

14、8086CPU在进行I/O写操作时，M/IO和DT/R引脚信号必须是（ ）

A. 00 B. 01 C. 10 D. 11



第二章 习题

15、取指令物理地址= ()

A. $(DS) \times 10H + \text{偏移地址}$ B. $(ES) \times 10H + \text{偏移地址}$

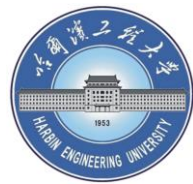
C. $(SS) \times 10H + (SP)$ D. $(CS) \times 10H + (IP)$

16、一个数据的有效地址是2140H、 $(DS) = 1016H$ ，则该数据所在内存单元的
物理地址为 ()

A. 12300H B. 122AH C. 02156H

(6) 当CPU时钟频率为5MHz，则其总线周期 ()

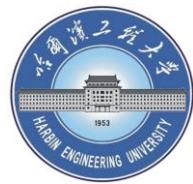
A. $0.8 \mu s$ B. 500ns C. 200ns D. $200 \mu s$



第二章 习题

判断题：

- (1) 8086中执行部件的功能是负责指令的执行。 ()
- (2) 8086 CPU中BIU和EU是互相独立、互相配合并行同步工作的，目的在于提高CPU的工作效率。 ()
- (3) 指令队列的存取原则是先进先出。 ()
- (4) 8086芯片中集成有高速缓冲存储器。 ()
- (5) 8086中，取指令和执行指令可以重叠操作。 ()



第二章 习题

- (6) 8086 CPU中的指令队列长度为4字节。()
- (7) 8086 CPU的标志寄存器有16位，每一位都有确定的含义。()
- (8) 若运算结果为0，则标志寄存器中的零标志位ZF=0。()
- (9) 若运算结果有溢出，则标志寄存器中 PF=1。()
- (10) 8086系统的存储器引入了分段的概念，每个段必须为64K，段与段之间可以是独立的。()



第二章 习题

- (11) 若运算结果为正，则标志寄存器中的 $SF=1$ 。（ ）
- (12) 总线周期就是机器周期。（ ）
- (13) 8086CPU从内存中读取一个字必须用两个总线周期。（ ）
- (14) 最小方式下，主设备向8086发出HOLD总线请求信号，若8086微处理器响应，则输出HLDA信号。（ ）
- (15) 指令周期是指CPU执行一条指令的时间。（ ）



第二章 习题

简答题：

1、设IBM PC微机内存中某个单元的物理地址是12345H，试完成下列不同的逻辑地址表示：

(1) 1234H:___H; (2) ____H:0345H

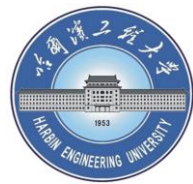
2、给定一个堆栈区，其地址为1250H: 0000H~ 1250H:0100H，(SP)=0052H，问：

①栈顶地址是什么？

②栈底地址是什么？

③在堆栈段寄存器SS中的段地址是什么？

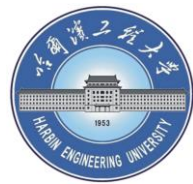
④若存入数据1000H,数据在栈中如何放置？ (SP) = ?



第二章 习题

简答题：

- 3、计算机主频为8MHz，每个机器周期平均含两个时钟周期，每条指令平均有2.5个机器周期，则该机器的平均指令执行速度为多少MIPS？
- 4、8086CPU由哪两部分组成？它们的主要功能是什么？
- 5、在8086/8088系统中，何为分时复用总线？其优点何在？试举例说明。



第二章 习题

简答题：

- 6、状态标志位和控制标志位有何不同？8086/8088的状态标志和控制标志位有哪些？
- 7、8086/8088系统用的时钟发生器产生哪些信号？
- 8、什么是地址锁存器？8086/8088系统中为什么要用地址锁存器？锁存的是什么信息？