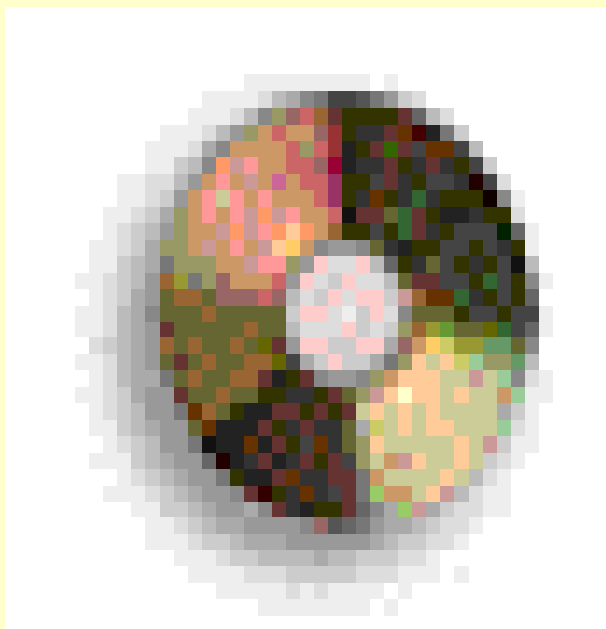
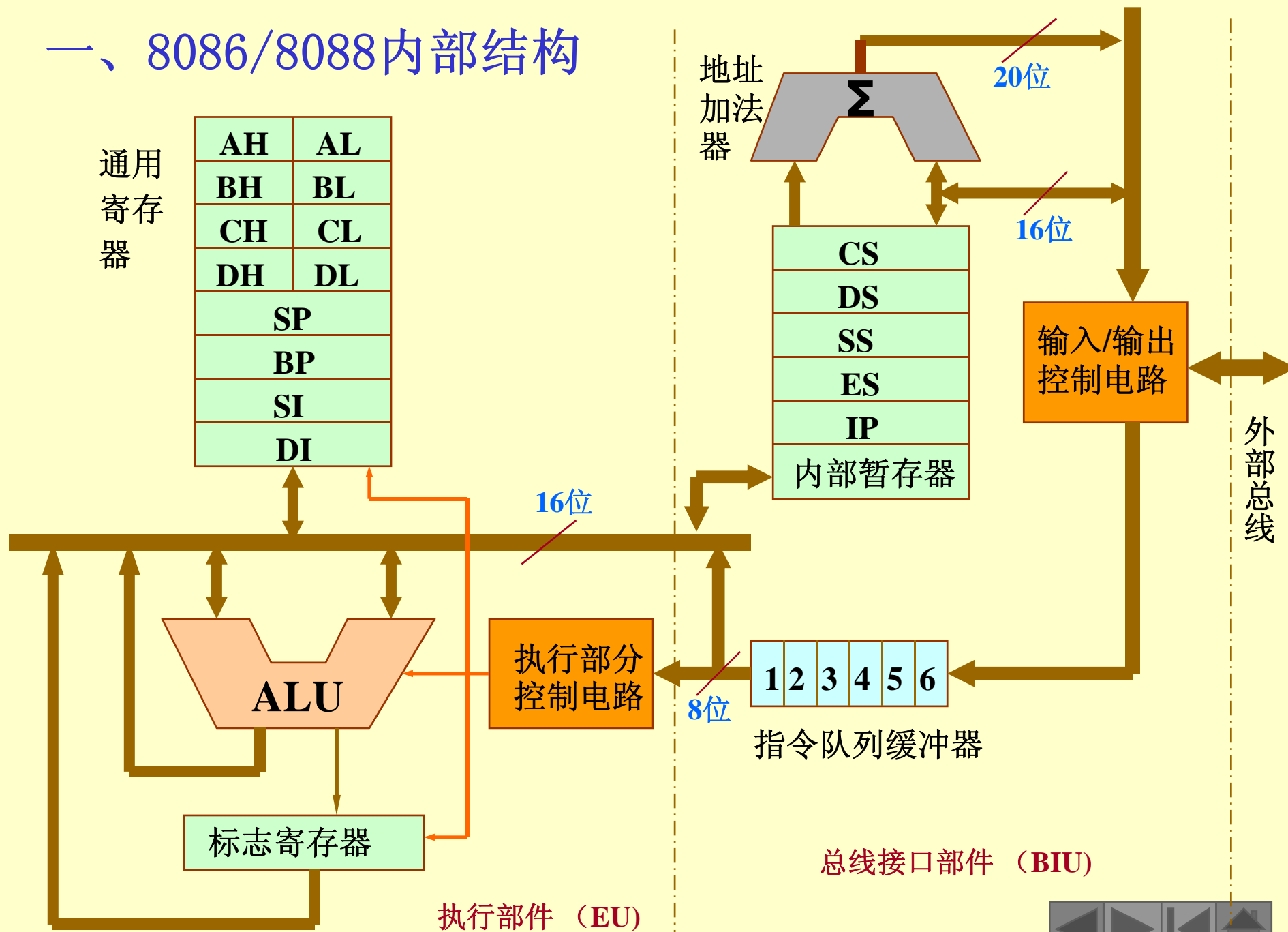


## 第二章 Intel 8086/8088 微机系统结构



- 一、Intel 8086/8088 内部结构
- 二、功能结构
- 三、存储器组织
- 四、指令运行
- 五、指令执行时间

# 一、8086/8088内部结构



# 1. 结构

- 寄存器结构

8位寄存器

AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

(A)

累加器  
基址寄存器  
计数器  
数据寄存器

16位寄存器

SP
BP
SI
DI

(SP)

堆栈指针寄存器  
基址寄存器  
源变址寄存器  
目的变址寄存器

IP
FLAG

(PC)

(PSW)

指令指针寄存器  
状态标志寄存器

CS
DS
SS
ES

代码段寄存器  
数据段寄存器  
堆栈段寄存器  
附加段寄存器

通用寄存器

控制寄存器

段寄存器



## 状态标志寄存器

	O	D	I	T	S	Z		A		P		C
--	---	---	---	---	---	---	--	---	--	---	--	---

状态标志

进位标志C (Carry Flag) —— 当结果的最高位产生一个进位或借位, 则C=1, 否则C=0。

溢出标志O (Overflow Flag) —— 在算术运算中, 带符号数的运算结果超出了8位或16位带符号数能表达的范围, 则O=1, 否则O=0。

8位(字节)运算 -128 ~ +127

16位(字)运算 -32768 ~ +32767

符号标志S (Sign Flag) —— 结果的最高位(D<sub>15</sub>或D<sub>7</sub>)为1, 则S=1, 否则S=0。

零标志Z (Zero Flag) —— 若运算的结果为0, 则Z=1, 否则Z=0。

奇偶标志P (Parity Flag) —— 若结果中‘1’的个数为偶数, 则P=1, 否则, P=0。

辅助进位标志A (Auxiliary Flag) —— 在字节操作时, 由低半字节(第3位)向高半字节, 字操作时低字节向高字节有进位或借位, 则A=1, 否则A=0。

控制标志

方向标志(Direction Flag) —— D=1, 串操作时地址自动减量, D=0, 串操作时地址自动增量。

中断允许标志(Interrupt—enable Flag) —— I=1, 则允许CPU接收外部的中断请求, I=0, 则屏蔽外部中断请求。

追踪标志(Trace Flag) —— T=1, 使处理进入单步方式, 以便于调试。



例：两个带符号数 64 h , 64 h 相加。

$$\begin{array}{r} 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0 \\ +\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0 \\ \hline 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0 \end{array}$$

**O=1**（运算结果超过127）；

**C=0**；

**Z=0**；

**S=1**；

**P=0**；

**A=0**。



- 算术逻辑单元

负责各种算术和逻辑运算。

- 地址加法器

将左移 4 位的段寄存器的内容与偏移地址相加，形成 20 位的物理地址，以便对 1 兆空间的存储器寻址。

- 指令队列

指令队列中包含若干个（8086—6个，8088—4个）8 位寄存器，用于顺序存放从存储器中取出的指令，供执行单元执行。

- 输入/输出控制电路

提供系统总线的控制信号，实现数据、地址和状态信息的分时传送。

- 执行单元控制电路

是控制、定时与状态逻辑电路。用于控制执行单元中各部件按制定的要求协调工作。



## 二、 功能结构

8088从功能上来说分成两大部分，总线接口单元 BIU (Bus Interface Unit)和执行单元 EU (Execution Unit).

- BIU: 负责与存储器接口，即8088与存储器之间的信息传送，都是由BIU负责进行的，即：
  - (1) BIU从内存的指定部分取出指令，送至指令队列排队。
  - (2) 从内存的指定部分取出执行指令时所需的操作数，送至EU部分。
- EU: 负责指令的执行。

特点： 将CPU分为两个单元，可以使取指令和执行指令同时进行，减少了CPU为取指令而等待的时间，从而提高了CPU的利用率，提高了系统的运行速度。



### 三、存储器组织

8088有20条地址线，它的直接寻址能力为  $2^{20}=1\text{M}$ ，所以在一个8088组成的系统中，可以有多达1M字节的存储器，这1M字节在逻辑上可以组成一个线性矩阵，地址从00000到FFFFFF。给定一个20位的地址，就可以从中取出所需的指令和数据。

在8088内部，ALU 能进行16位的运算，有关地址寄存器如：IP、SP、BX、BP、SI、DI等都是16位的，因而对地址的运算也只能是16位。其寻址的范围最多是  $2^{16}=64\text{K}$ 。

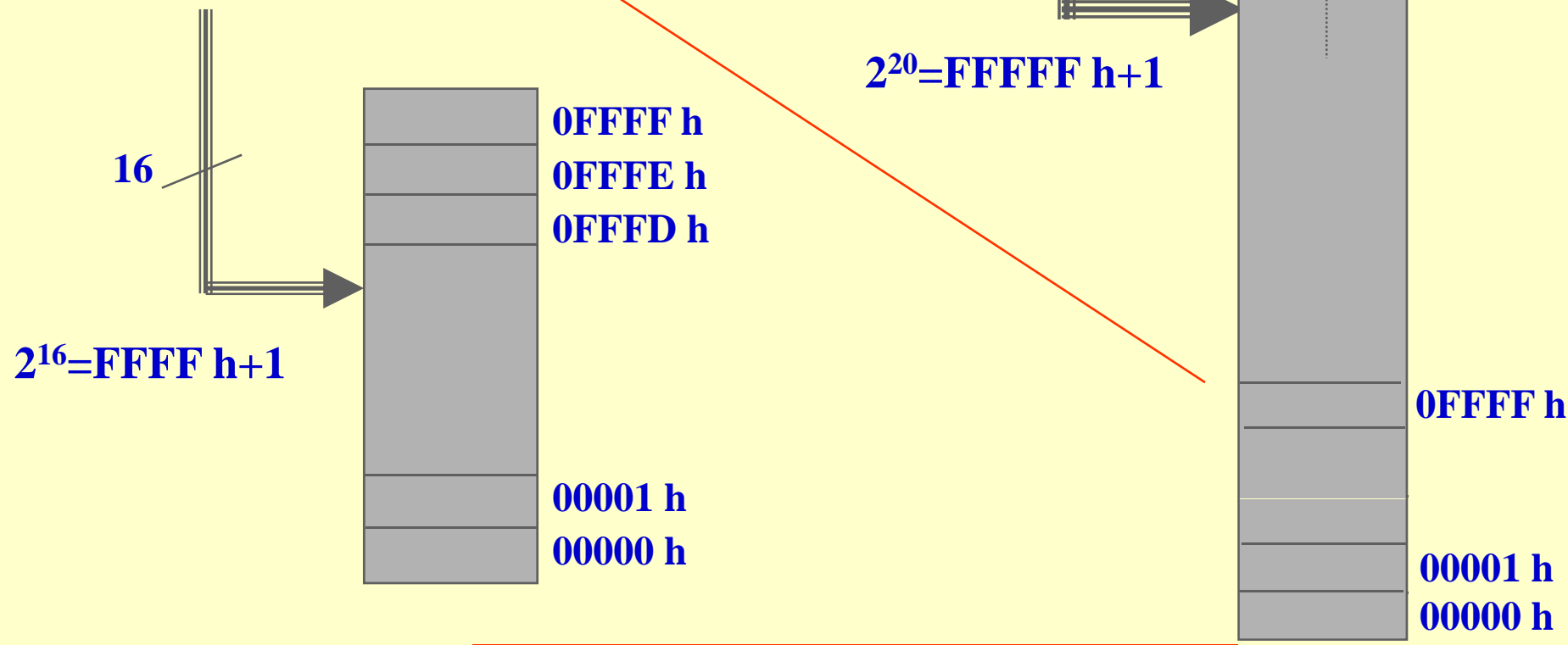
在8088中怎样形成这20位的地址呢？——分段！

将内存的1M字节以64K为范围，分成若干段；在8088中设置4个段寄存器CS、DS、SS、ES，在寻址一个具体的内存单元（物理地址）时，由一个段寄存器中保存的16位基地址，加上由IP（或SP，或BP，或BX，或SI，或DI）中保存的、可由CPU处理的16位偏移量，来形成20位的物理地址。

在形成物理地址时，地址加法器将段寄存器中的16位数左移4位，然后与16位偏移地址相加，形成20位的物理地址。







## 四、指令运行

操作： 将两个数 相加.

指令： **mov al, [2000]**

**add al, 02**

**hlt**

机器指令： **1010 0000**      **A0h**      **mov al,[2000]**

**0000 0000**      **00h**

**0010 0000**      **20h**

**0000 0100**      **04h**      **add al,02**

**0000 1010**      **02h**

**1111 0100**      **F4h**      **hlt**

(运算过程展示见动画)



## 五、指令执行时间

- 指令周期 —— 执行一条指令所需的时间称为指令周期。

在8088/8086中，由于它的并行处理的特点，一条指令的执行时间为：

取操作数+执行指令+传送结果      进一步说：

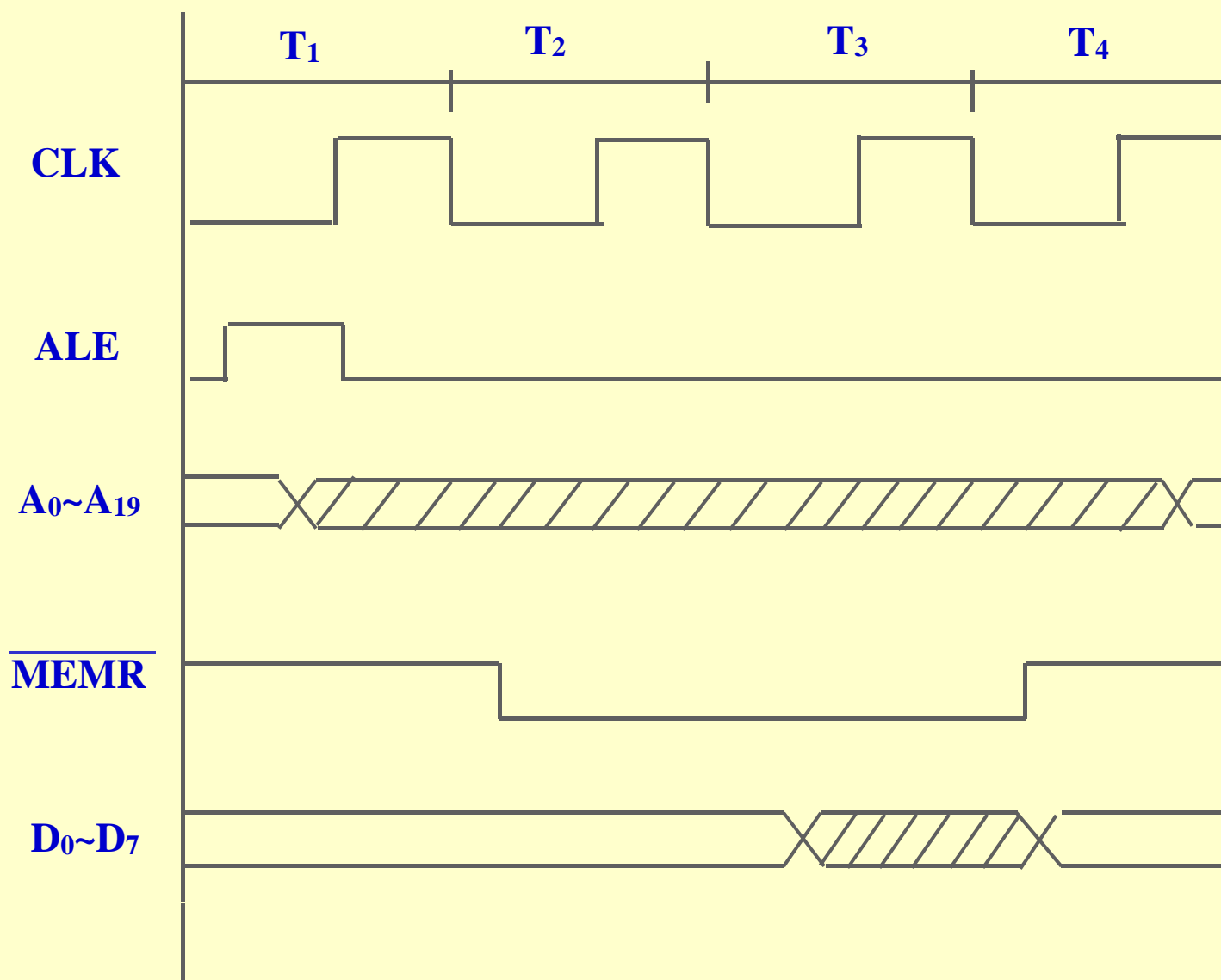
基本时间+计算有效地址的时间+为了读取操作数和存放操作结果而访问内存的时间

（但是，在8086/8088中不同的指令的指令周期是不同的，因此我们

再将其细划，把一个指令周期划分为一个个总线周期。）

- 总线周期——CPU从存储器或输入输出端口，存（或取）一个字节所需的时间，即为总线周期。
- T 状态 —— 每个总线周期通常包含4个T状态。T状态是8088中处理动作的最小时间单位，它就是时钟周期。（8088的时钟频率为5MHz, 故时钟周期，也即一个T状态为200ns。）





存储器读总线周期

8086的总线周期有4个时钟周期组成：T<sub>1</sub>、T<sub>2</sub>、T<sub>3</sub>、T<sub>4</sub>（四个状态）

- T<sub>1</sub>状态：CPU向总线上发出地址信息，指出要寻址的内存单元或外设端口地址（内存为20位A<sub>19</sub>~A<sub>0</sub>，外设为8位地址A<sub>7</sub>~A<sub>0</sub>）
- T<sub>2</sub>状态：地址的低16位呈高阻状态，最高4位（A<sub>19</sub>~A<sub>16</sub>）输出本总线周期的状态信息（中断允许状态、当前正在使用的寄存器名等）
- T<sub>3</sub>状态：最高4位继续提供状态，低16位出现要读写的数据（8088CPU则位低8位）
- T<sub>w</sub>状态：当外设或存储器速度不能跟上CPU的速度时，外设或存储器会通过“READY”信号线在T<sub>3</sub>状态开始前向CPU发出一个“数据未准备好”信号，则CPU会在T<sub>3</sub>状态之后插入一个或多个T<sub>w</sub>状态，当READY线变低，表明数据准备好后，CPU自动脱离T<sub>w</sub>状态，进入T<sub>4</sub>状态
- T<sub>4</sub>状态：总线周期结束
- 空闲周期：只有在取指、内存、I/O访问时才执行总线周期，如果不马上执行下一个总线周期，则在总线进入空闲状态，执行空闲周期

在空闲周期中，高4位仍是上一总线的状态信息，如果上一总线周期为写周期，低16位上仍是写数据，如位读周期，则低16位呈高阻状态。



# 8086的引脚信号和工作模式

- 最小模式和最大模式的概念
    - 最小模式：系统中只有一个8086/8088微处理器，所有的总线控制信号由它产生
    - 最大模式：系统中有多个微处理器，其中一个主处理器为8086/8088，其它的称为协处理器。
      - 8087数值运算协处理器：高精度的整数和浮点运算，超越函数的计算
      - 8089输入/输出协处理器：两个DMA通道的管理
  - 8086/8088的引脚信号和功能（见下页）
    - 8086/8088的数据线和地址线是复用的。
    - 第28脚和34脚8086和8088的定义不一样，其它一样。
    - 第28脚的定义电平意义刚好相反，8088：M/IO；8086：M/IO
    - 第34脚：8086：BHE/S<sub>7</sub>，因为8086可以传送16位、高8位、低8位的数据，而8088只是8位的数据宽度传送，故派作状态用（SS<sub>0</sub>）
- 1、GND/VCC：地/电源，8086和8088都用+5V电源。
  - 2、AD<sub>15</sub> ~ AD<sub>0</sub> 地址/数据复用，在T<sub>1</sub> 状态为地址，T<sub>2</sub>、T<sub>3</sub> 状态读为高阻，写为数据（双向）

AD<sub>0</sub> 为低电平，则表明AD<sub>7</sub> ~ AD<sub>0</sub> 和偶地址交换数据



# 8086微处理器引脚图

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

括号中为最大  
模式时引脚名

地	1	40	Vcc(5V)
A14	2	39	A15
A13	3	38	A16/S3
A12	4	37	A17/S4
A11	5	36	A18/S5
A10	6	35	A19/S6
A9	7	34	<u>SS0</u>
A8	8	33	MN/ <u>MX</u>
AD7	9	32	<u>RD</u>
AD6	10	31	HOLD ( <u>RQ/GT0</u> )
AD5	11	30	HLDA ( <u>RQ/GT1</u> )
AD4	12	29	<u>WR</u> ( <u>LOCK</u> )
AD3	13	28	<u>M IO</u> ( <u>S2</u> )
AD2	14	27	DT/ <u>R</u> ( <u>S1</u> )
AD1	15	26	<u>DEN</u> ( <u>S0</u> )
AD0	16	25	ALE (QS0)
NMI	17	24	<u>INTA</u> (QS1)
INTR	18	23	<u>TEST</u>
CLK	19	22	READY
地	20	21	RESET



# 8086的引脚信号和工作模式

## 3、 $A_{19}/S_6 \sim A_{16}/S_3$ 地址/状态复用，输出。

- 总线周期的 $T_1$ 状态输出 $A_{19} \sim A_{16}$ 高4为地址， $T_2$ 、 $T_3$ 、 $T_w$ 、 $T_4$ 时输出状态信息。
- $S_6$ 为0用来指示8086/8088当前与总线连接（告诉其它处理器）。
- $S_5$ 中断允许标记的状态，1：允许可屏蔽中断，0：禁止可屏蔽中断。
- $S_4 \sim S_3$  指出当前的使用的段寄存器。

0	0	ES
0	1	SS
1	0	CS或者未使用任何段寄存器
1	1	DS

## 4、BHE/ $S_7$ 高8位数据允许/状态：输出 BHE和 $A_0$ 结合赋予读写操作不同的含义：

<u>BHE</u>	$A_0$	操作	数据引脚
0	0	从偶地址开始读/写一个字	$AD_{15} \sim AD_0$
1	0	从偶地址开始读/写一个字节	$AD_7 \sim AD_0$
0	1	从奇地址开始读/写一个字节	$AD_{15} \sim AD_8$
0	1	从奇地址开始读/写一个字	$AD_{15} \sim AD_8$
1	0		$AD_7 \sim AD_0$

8086中从奇地址开始读写一个字需要两个总线周期





## 8086的引脚信号和工作模式

### 5、NMI 非屏蔽中断：输入

正沿触发，不受IF的影响，中断的类型号为2

### 6、INTR可屏蔽中断请求：输入

8086/8088在执行每条指令的最后一个时钟周期时采样INTR，当IF为1，则响应该中断。

### 7、RD读信号：输出

执行读内存和读I/O操作，是内存还是I/O由M/IO决定。

### 8、CLK 时钟输入

占空比为33%，1/3为高电平，2/3为低电平

### 9、RESET 复位信号输入

至少维持4个时钟周期，清除寄存器及指令队列，CS=FFFFH，当RESET变为低电平时，从FFFF0H地址开始执行复位程序。

### 10、READY 就绪或准备好信号输入

### 11、TEST 测试信号输入

该信号和WAIT指令配合使用：当执行了WAIT指令后，8086接收到该引脚信号，则结束等待状态，CPU继续往下执行。

### 12、MN/MX 最小/最大模式控制信号输入（固定）



## 8086的引脚信号在最小模式时的含义

### 1、INTA中断响应信号输出

在 $T_2$ 、 $T_3$ 和 $T_w$  INTA连续输出两个低电平，第一个负脉冲通知外设中断请求已被允许，第二个负脉冲要求外设放中断类型码。

### 2、ALE地址锁存允许信号

在 $T_1$ 周期输出有效电平

### 3、DEN数据允许信号：CPU当前准备发送或接收一个数据

### 4、DT/R数据传送方向表示：低接收，高发送

### 5、M/IO存储器/输入输出控制信号输出

### 6、WR写信号输出

### 7、HOLD总线请求信号，输入

### 8、HLDA总线响应信号，输出

8088CPU 的第34引脚称SS<sub>0</sub>，它和M/IO、DT/R组合决定当前总线周期的操作



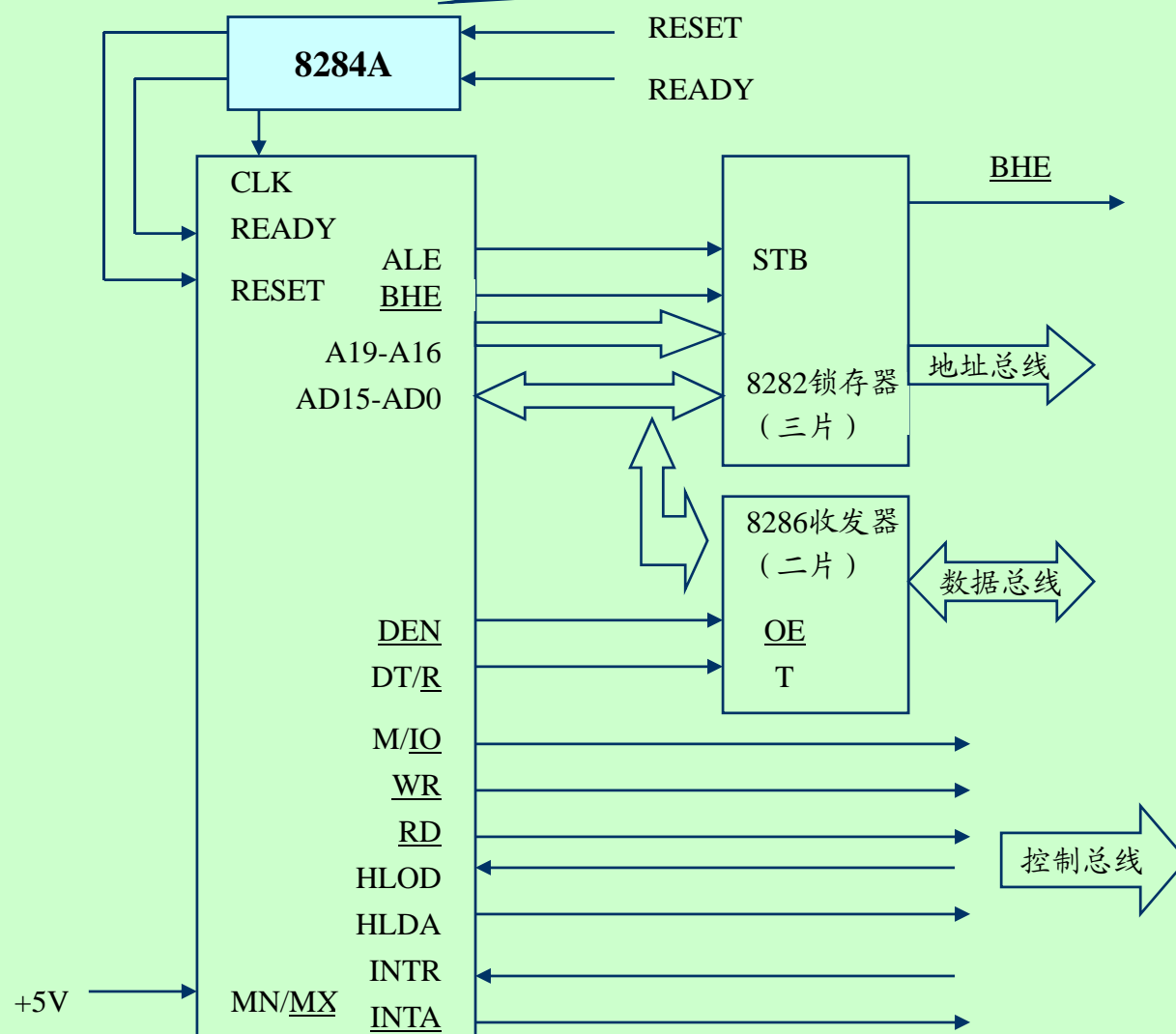
## 8088的SS<sub>0</sub>、M/IO、DT/R组合

<u>M</u> /IO	DT/ <u>R</u>	<u>SS</u> <sub>0</sub>	操作
0	0	0	取指令
0	0	1	读内存
0	1	0	写内存
0	1	1	无源状态
1	0	0	发中断响应信号
1	0	1	读I/O端口
1	1	0	写I/O端口
1	1	1	暂停



# 8086最小模式的典型配置

8284连接见后



## 8086的引脚信号在最大模式时的含义

### 1、 $QS_1$ 、 $QS_0$ 指令队列状态信号输出

0	0	无操作
0	1	从指令队列中的第一个字节取走代码
1	0	队列为空
1	1	除第一个字节外，还取走了后续字节中的代码

### 2、 $\underline{S}_2$ 、 $\underline{S}_1$ 、 $\underline{S}_0$ 总线周期状态输出信号

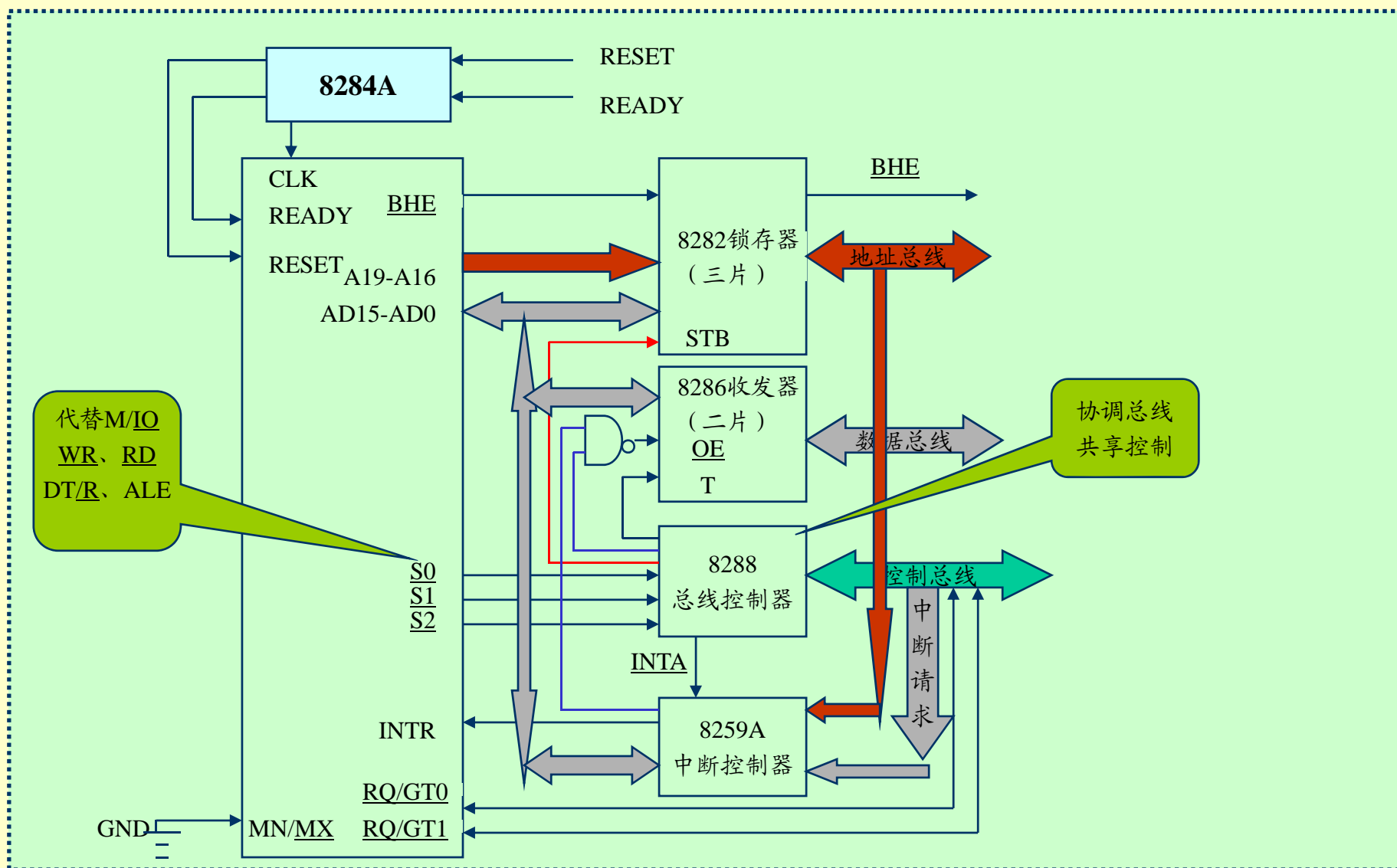
0	0	0	发中断响应信号
0	0	1	读I/O端口
0	1	0	写I/O端口
0	1	1	暂停
1	0	0	取指令
1	0	1	读内存
1	1	0	写内存
1	1	1	无源状态

### 3、 $\underline{LOCK}$ 总线封锁信号输出，系统中其它设备不能占用总线

### 4、 $\underline{RQ}_1/\underline{GT}_1$ 、 $\underline{RQ}_0/\underline{GT}_0$ 总线请求输入/总线请求允许输出



# 8086最大模式的典型配置



# 8086操作和时序

- 系统复位和启动操作

当RESET 信号变高 (维持)

标志寄存器: 清零

指令寄存器 (IP) 0

CS=FFFFH、DS=SS=ES=0000H

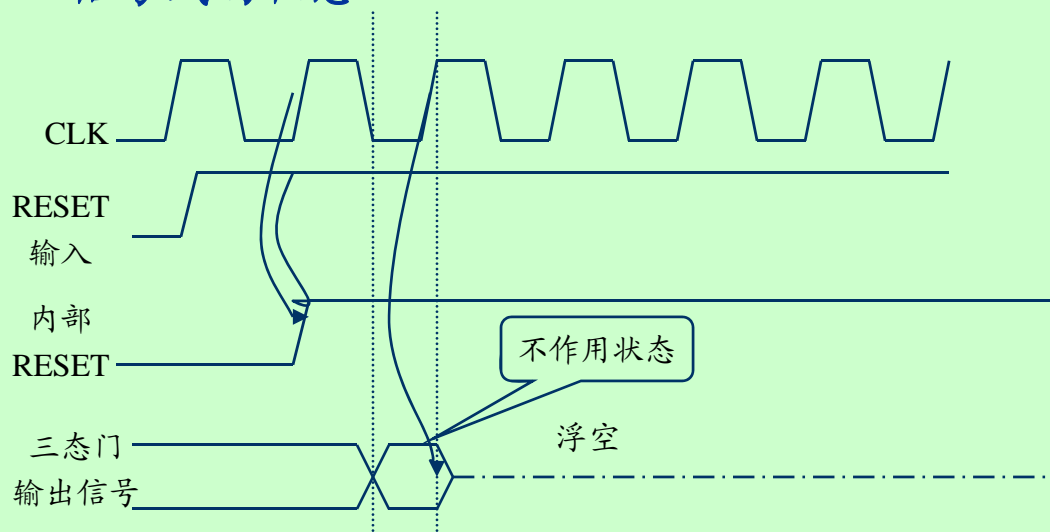
指令队列 空

其它寄存器 0000H

当RESET 信号变低 (起码4个时钟周期的高电平; 加电50us)

从FFFF0H开始执行程序, 禁止可屏蔽中断 (在适当处执行STI指令)

信号线的状态



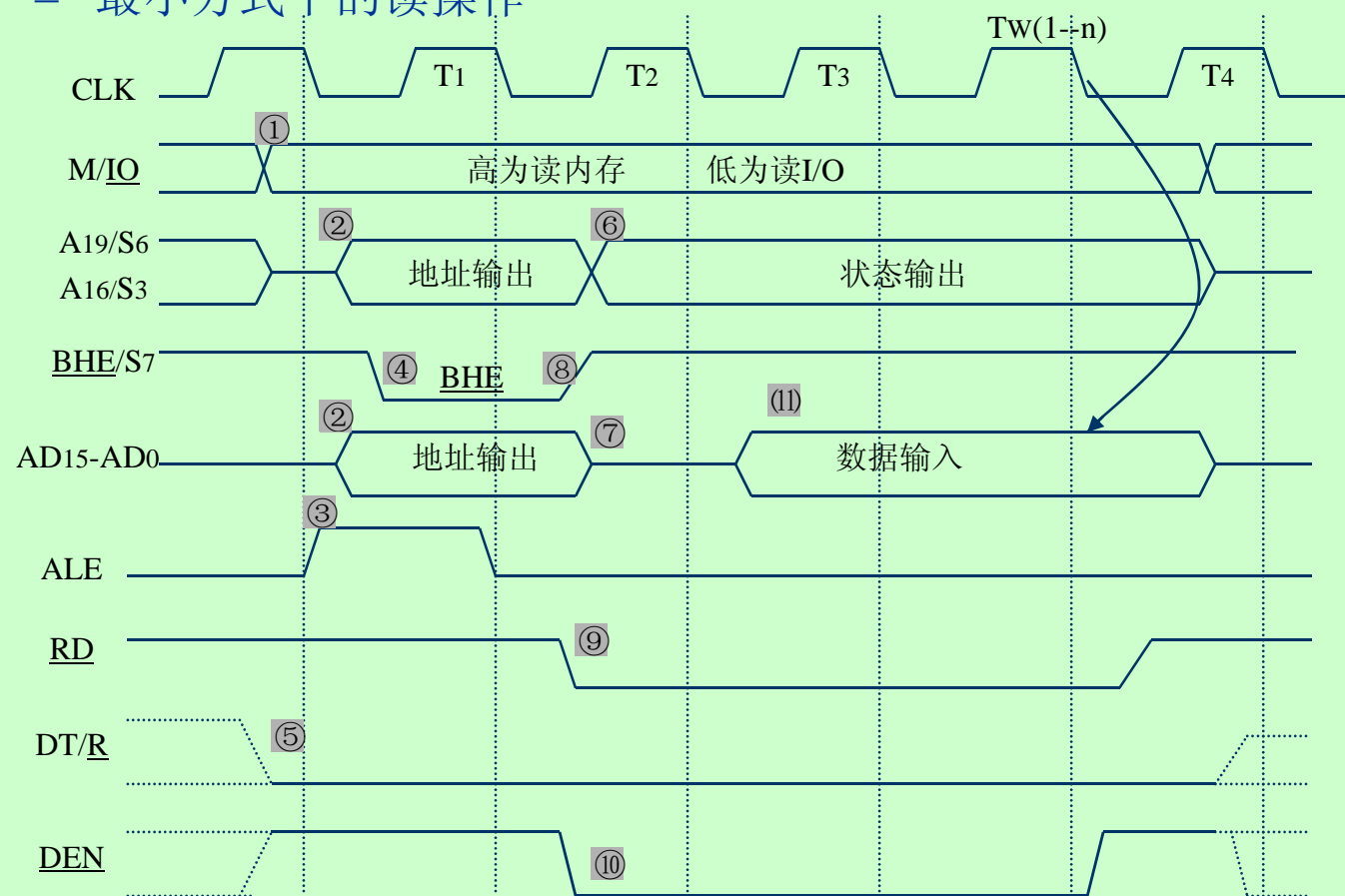
各信号状态见  
相关图表



# 8086操作和时序

- 总线操作(内存读写,I/O读写)

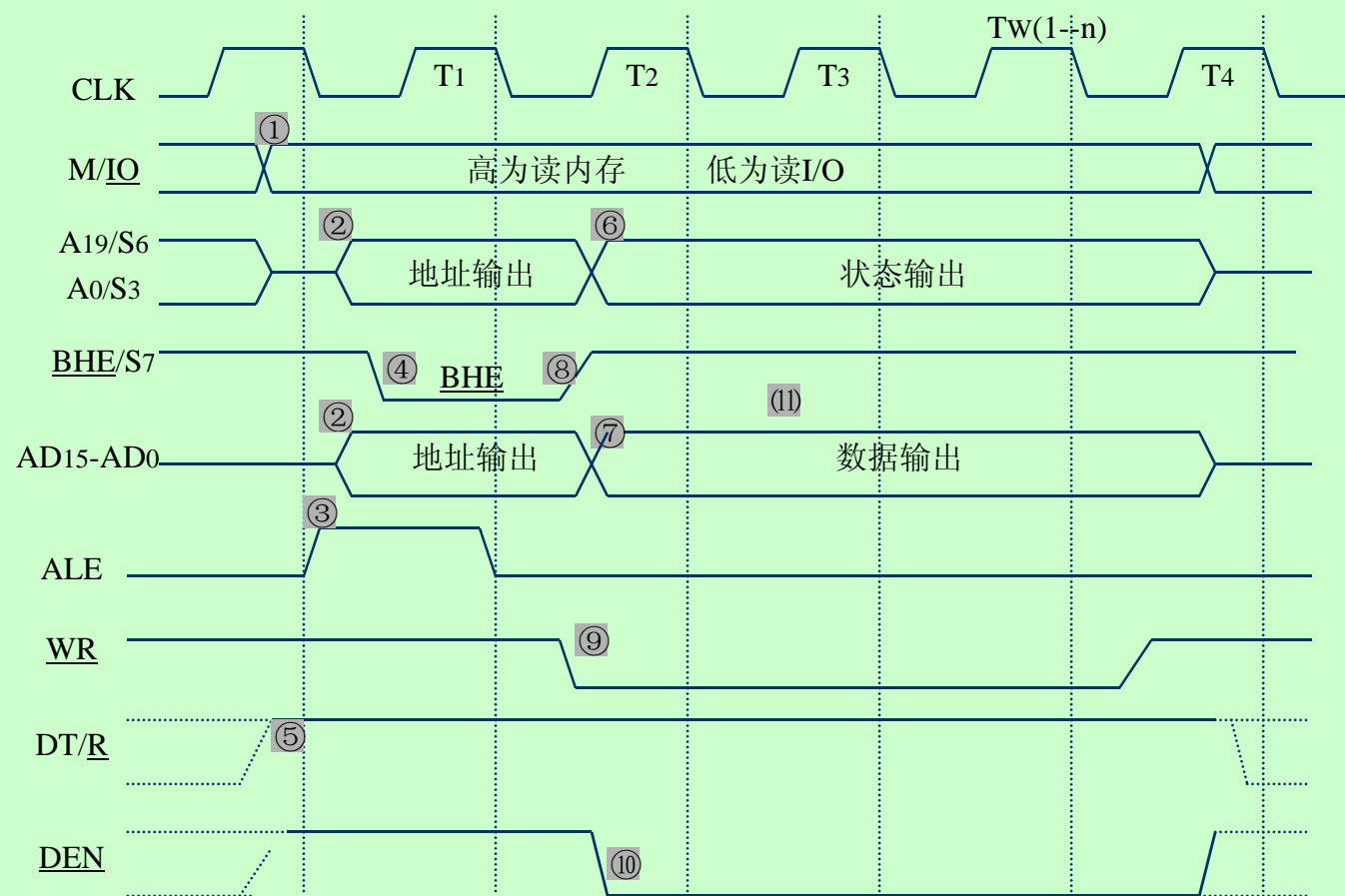
- 最小方式下的读操作





# 8086操作和时序

## – 最小方式下的写操作



## 8086操作和时序

### – 最大方式下的读操作

在最大模式下，RD和由8288产生的MRDC、IORC都在T2状态发出，但往往用后者（为什么？）

### – 最大方式下的写操作

情况和最小模式下的写操作，用8288产生的MWTC、IOWC，另外还可利用一组提前一个时钟周期的AMWC、AIOWC

### – 总线空操作

BIU不和总线打交道，进入空闲周期TI

