



微机原理与接口技术

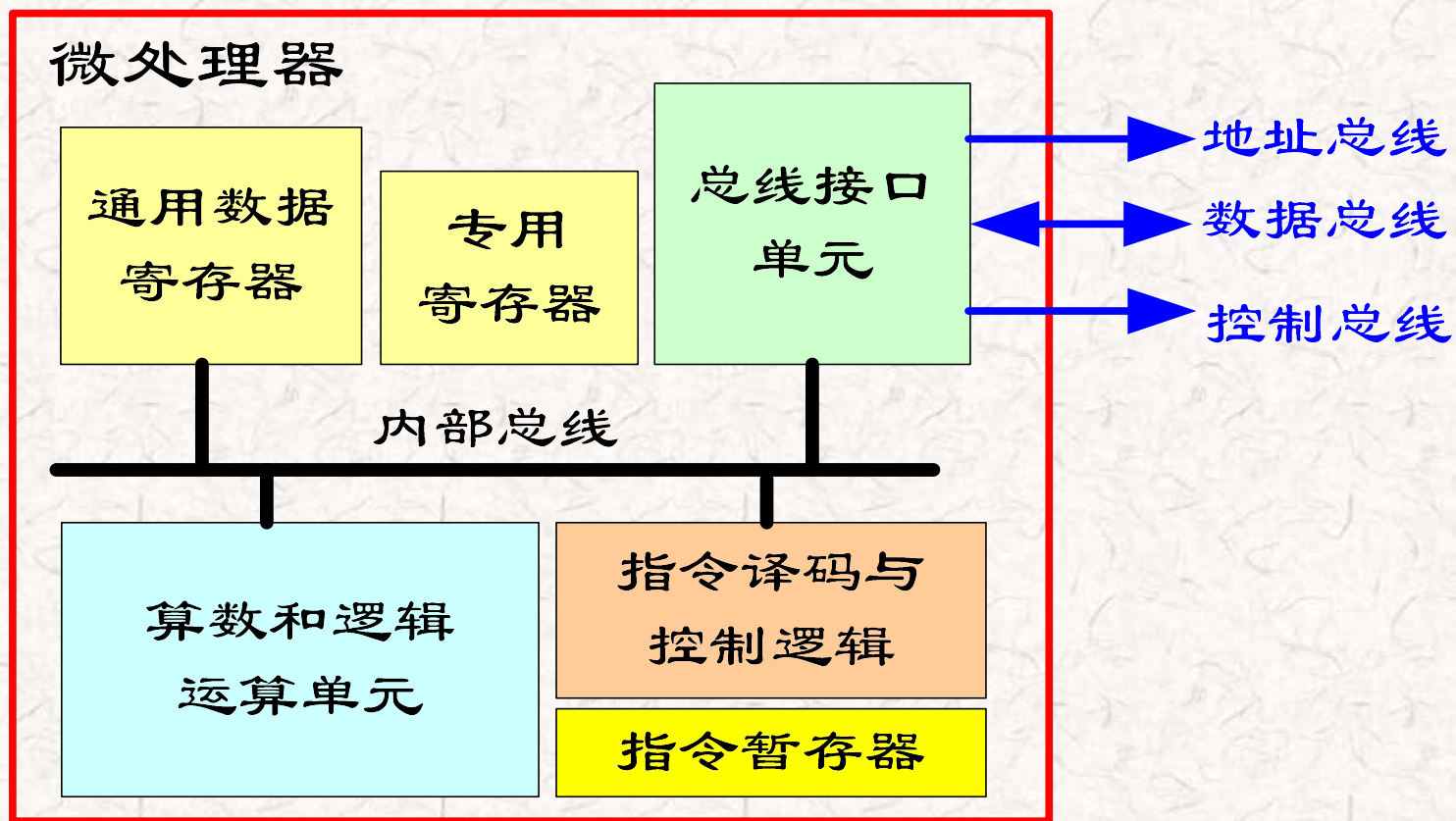
总复习-回顾一下这门课的重点知识

我们学到了什么？

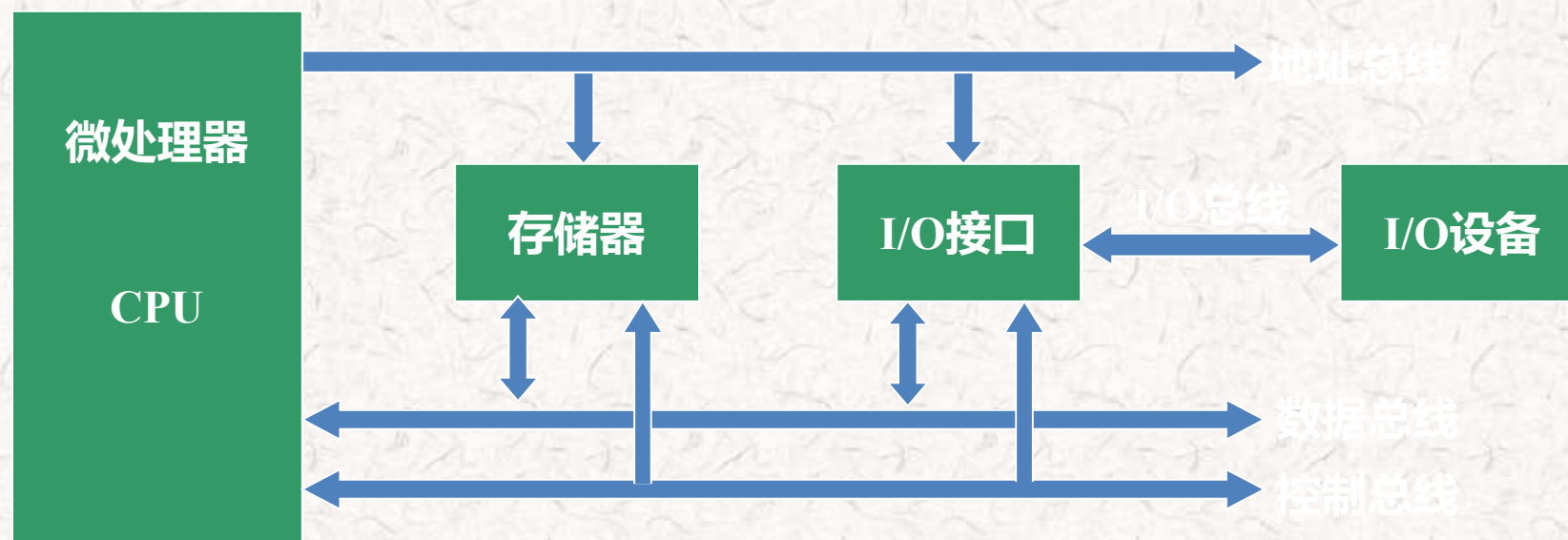
- 微处理器系统构成
- 汇编语言程序设计
- 微处理器系统硬件电路设计与编程

微处理器系统组成

微处理器的内部结构



微处理器硬件系统组成



三总线结构：地址总线、数据总线、控制总线

地址总线：单向，由CPU发出，指明要访问的存储器或I/O端口

数据总线：双向，传送读取的数据和写出的数据

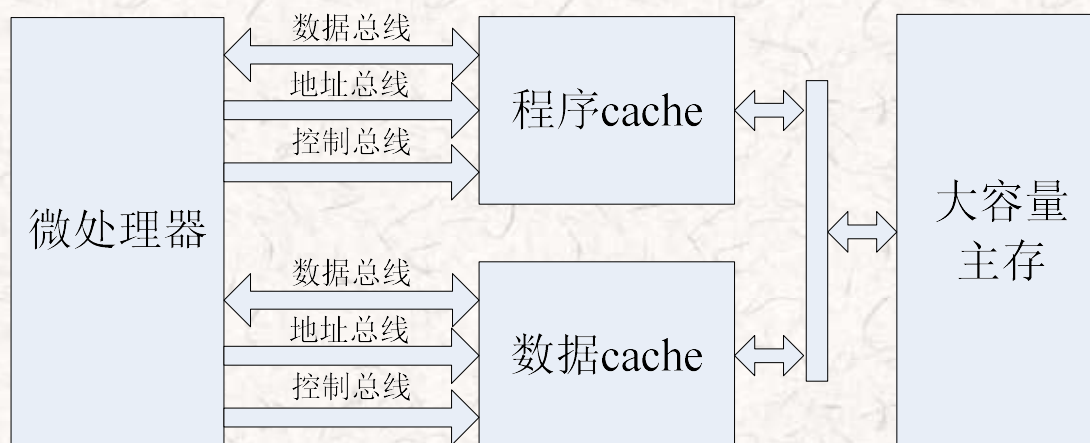
控制总线：单向，由CPU或外部电路发出，表明当前进行的读或写

冯诺依曼结构与哈佛结构

- 冯诺依曼结构（单总线）：地址、数据、控制（三总线）
- 哈佛结构（双总线）：程序总线、数据总线
- 解决“存储墙”的方法之一



哈佛结构



现代应用更广泛的总线和存储结构



汇编程序设计



汇编程序设计需要注意什么？

- 区分不同的操作数
- 寄存器、立即数、存储器、IO端口
- 存储器：变量名或采用方括号表示的操作数
- **MOV AX,[2000H] MOV AX,[BX] MOV AX[BX+SI]**
- **MOV AX,2[BP][DI]**
- 理解指令中的存储器地址（逻辑地址和物理地址）
- 方括号中是什么？存储器的偏移地址
- 存储器操作数的段地址在哪里？预先设置好的**DS**或**SS**中
- 哪些寄存器可以放在方括号中？（即：寄存器间接寻址）

汇编程序设计需要注意什么？

- 必须有明确的位宽信息（字节 或 字）
 - 由指令中寄存器操作数的位宽决定（注意 寄存器 与 寄存器寻址的存储器 的区别）
 - 属性修饰符说明位宽
- 注意计算防止溢出
 - 加法提前扩展位宽
 - 乘法的乘积位宽是两个乘数位宽之和
 - 除法根据期望得到的结果位宽扩展被除数位宽
 - 区分有符号数和无符号数的位宽扩展方法

汇编程序设计需要注意什么？

- 条件跳转指令符合条件跳转，不符合条件顺序执行
- 子程序调用、返回和中断服务程序的进入、退出存在堆栈操作
- 过程（子程序和中断服务程序）的编写跳转目的地一定在本程序内
- 退出过程的最后一条指令一定是**RET**或**IRET**
- 一个过程内任何执行路径必须产生相同次数的进栈和出栈操作
- 进出栈均为**16**位并且顺序相反

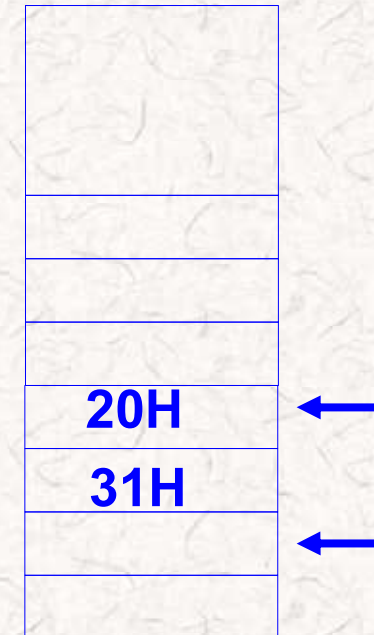
专门讲一讲堆栈

- 8086为硬件堆栈
- 进出栈均为2字节, 16bit
- 进栈举例 **PUSH BX**
- $SP < SP - 2$
- 准备进栈的数据低字节 $\rightarrow SP$
- 准备进栈的数据高字节 $\rightarrow SP + 1$
- 出栈举例 **POP AX**
- $SP \rightarrow$ 目的操作数的低字节
- $SP + 1 \rightarrow$ 目的操作数的高字节
- $SP < SP + 2$
- 低字节在低地址（偶数地址），高字节在高地址（奇数地址）
- 由上可知：初始化后实际堆栈数据写入空间为 $0 \sim SP - 1$

2000:0000H

2000:003EH

2000:0040H



AX = 3120H

BX = 3120H

什么时候产生堆栈操作？

- 进出栈指令：**PUSH XX**、**POP XX** (**XX: reg or mem**)
- 子程序调用**CALL XXX**指令：**(CS)**、**IP**进栈
- 子程序返回**RET**指令：栈顶弹出到**IP**、**(CS)**
- 进入中断服务程序（**硬件自动**）：**CS**、**IP**、**FLAGS**进栈
- 退出中断服务程序 **IRET**：栈顶弹出到**FALGS**、**IP**、**CS**

专门讲一讲标志寄存器

- 为特殊功能寄存器
- 6个标志位的含义 **CF OF ZF SF PF AF**
- 上述标志位由刚刚执行的**影响标志位的指令**的运算结果决定
- 与标志位/控制位相关的操作
- **CF**清除和置位指令 **CLC STC**
- **IF**清除和置位指令 **CLI STI**
- 复位后标志寄存器的值为0
- 中断响应过程中标志寄存器**TF**和**IF**被清除

专门讲一讲CS和IP

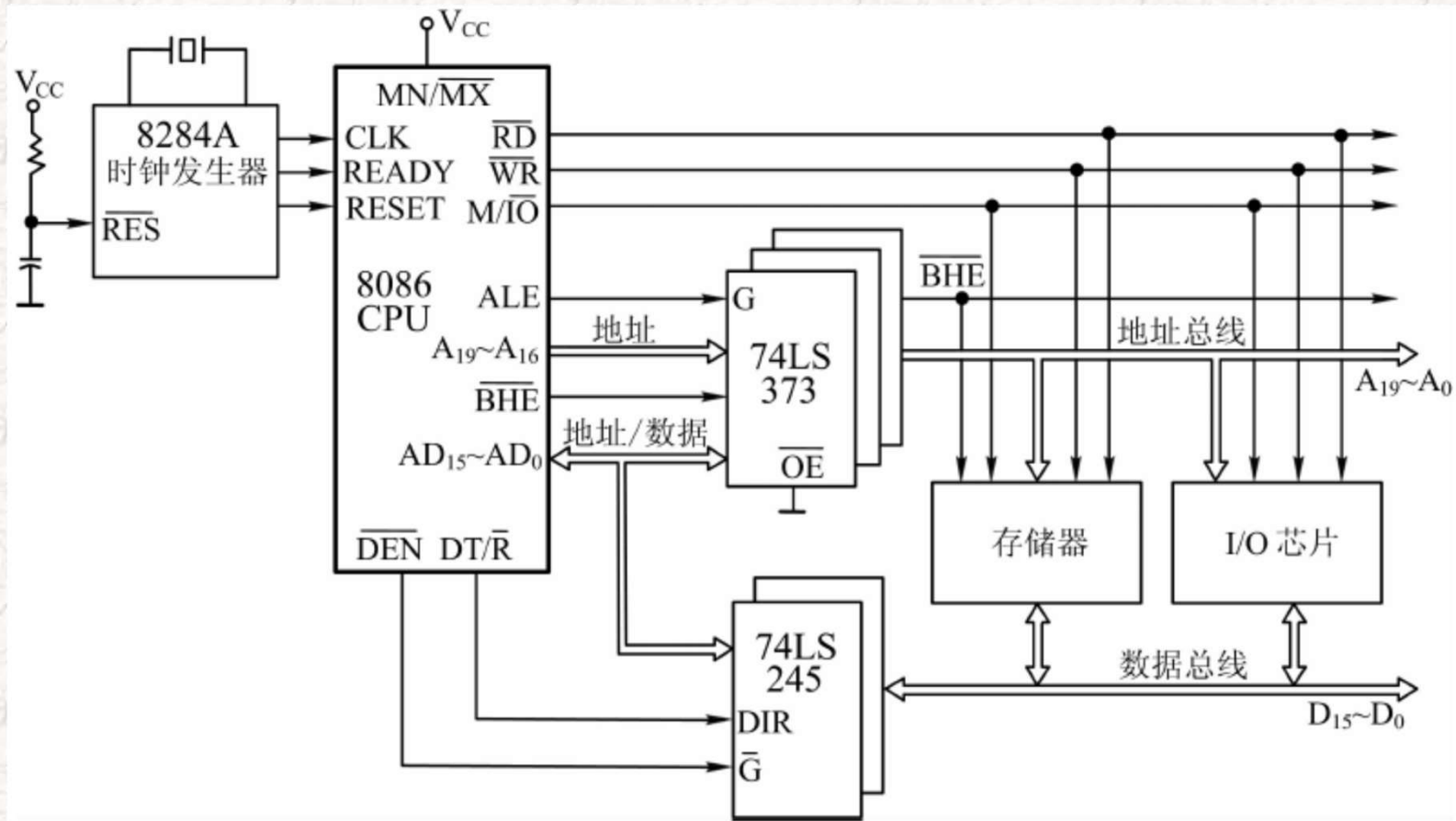
- 用于实现读取指令的操作
- 复位后**CS=0FFFFH**, **IP=0**
- 正常情况总是指向下一条指令, 不转移则始终顺序加
- **跳转指令** (不返回): 近跳转将修改**IP**的值, 远跳转修改**CS**和**IP**
- **CALL指令** (需要返回): 近调用**IP**进栈, 再修改**IP**; 远调用**CS**和**IP**进栈, 再修改**CS**和**IP**; 返回执行对应的出栈操作
- **中断响应** (需要返回): **CS**、**IP**、**FLAG**都要进栈, 并且清除**IF**和**TF**位, 再修改**CS**、**IP**为中断服务程序地址, 退出时一次出栈



微处理器系统硬件电路 设计



如何生成8086系统总线

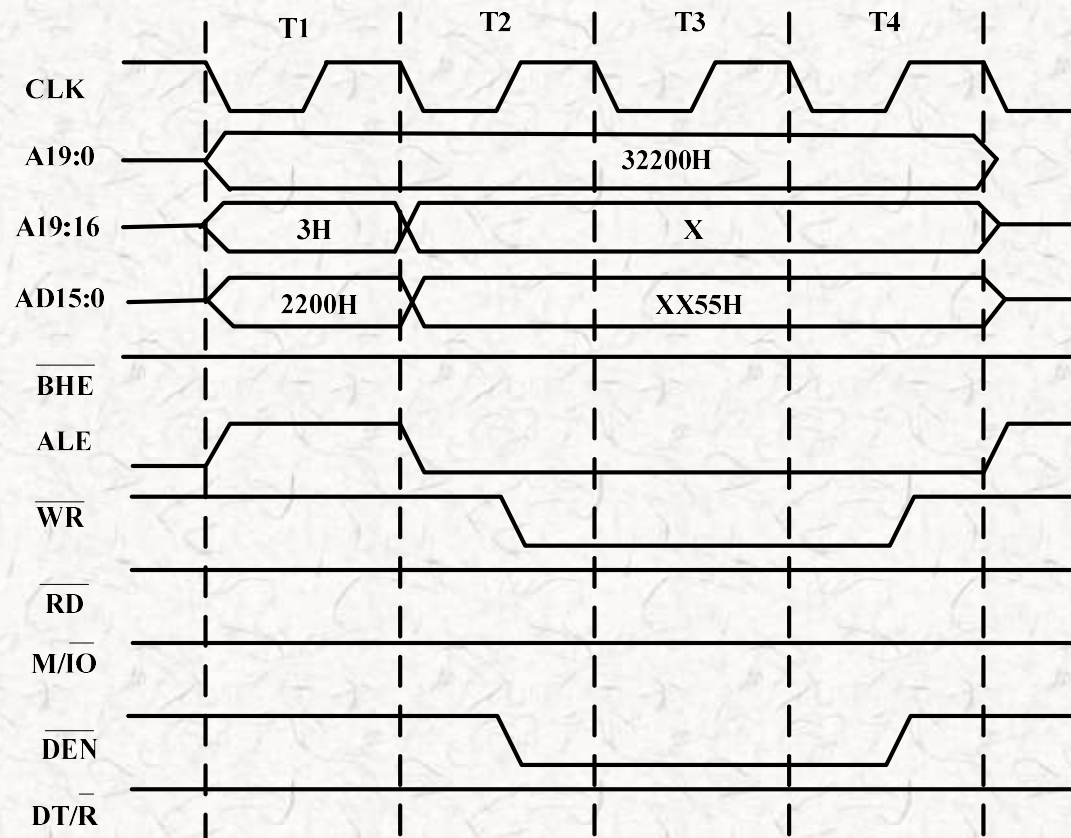


如何理解微处理器硬件系统工作方式？

- 微处理器系统中的存储器与外设都连接于微处理器系统总线
- 来自微处理器、存储器、输入端口的数据在数据总线上传递
- 数据传送的方向：由WR、RD指明
- 允许谁的数据在总线上传递：通过地址总线和M/IO指明
- CPU给出的地址能够选中谁？由地址分配的译码逻辑决定
- 什么时候产生与CPU之间的数据传输？由总线周期决定
- 总线周期什么时候产生？执行存储器和IO的访问指令时
- 硬件电路的操作在CPU总线信号的控制之下

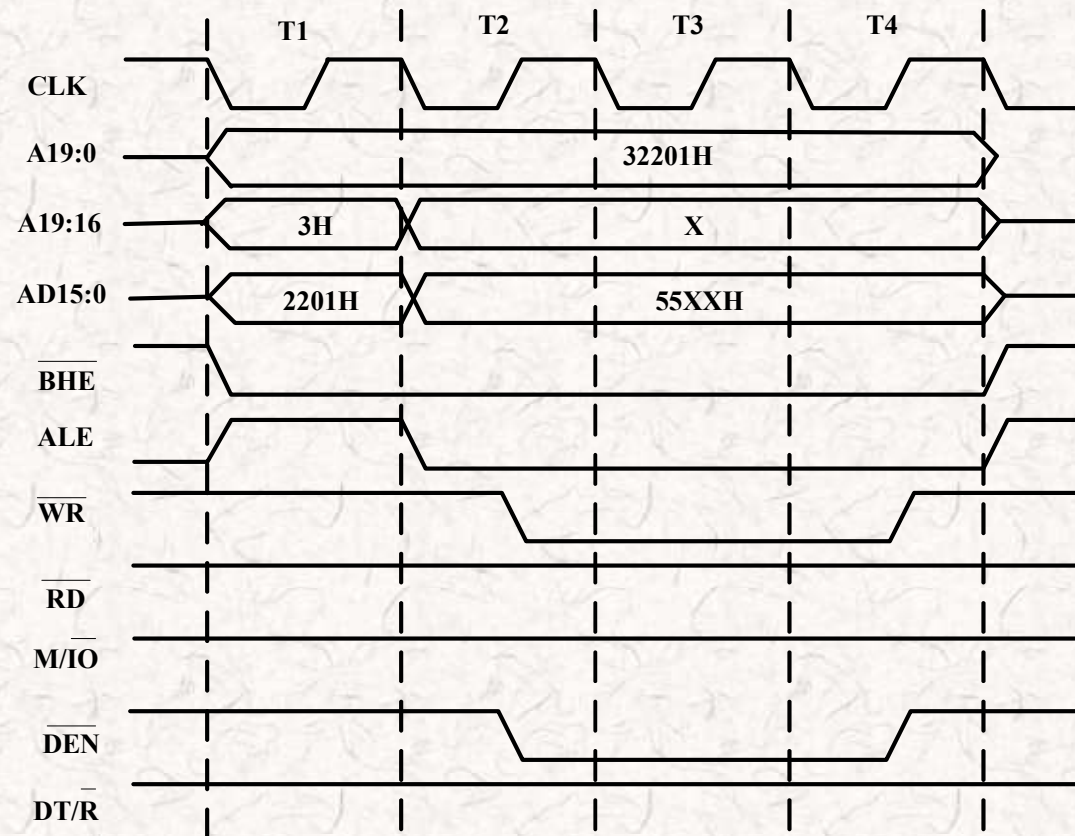
写总线操作举例1-写偶地址字节

若DS=3000H, AL=55H, 执行 MOV [2200H], AL时产生如下总线周期



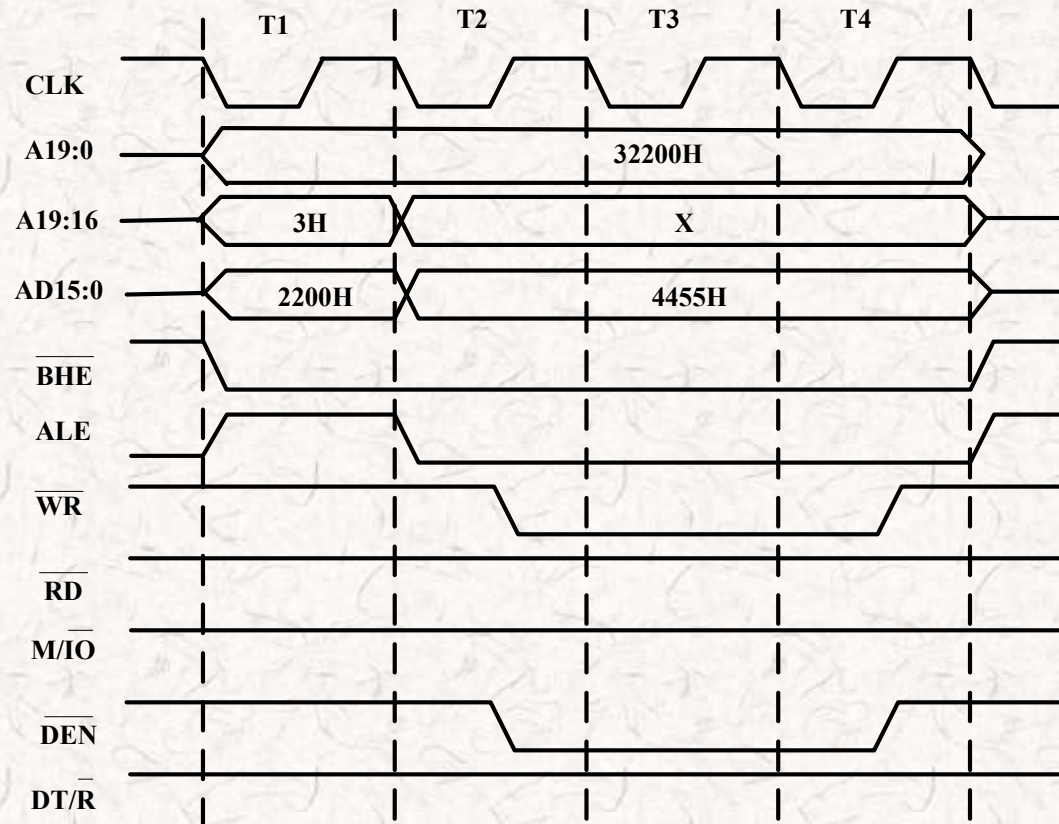
写总线操作举例2-写奇地址字节

若DS=3000H, AL=55H, 执行 MOV [2201H], AL时产生如下总线周期



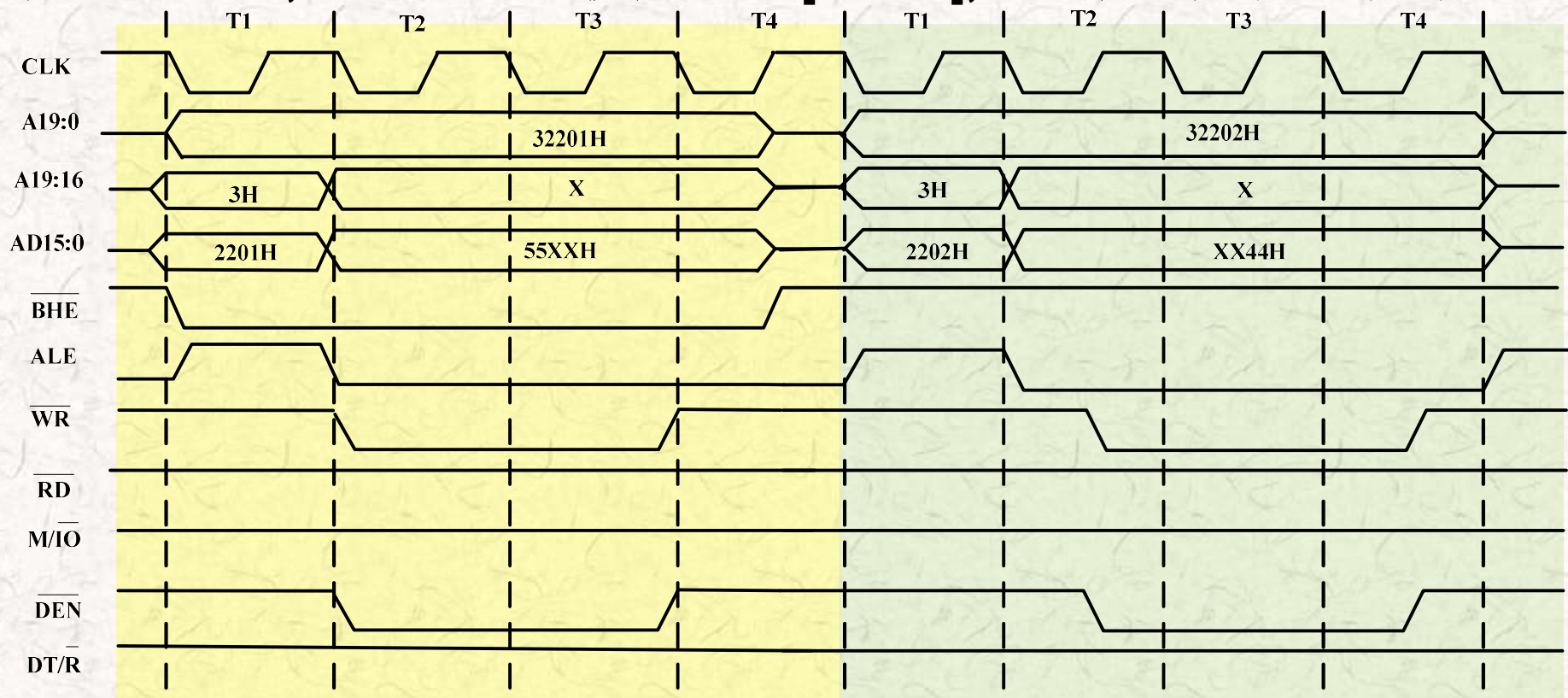
写总线操作举例3-写偶地址字

若DS=3000H, AX=4455H, 执行 MOV [2200H], AX时产生如下总线周期



写总线操作举例4-写奇地址字

若DS=3000H, AX=4455H, 执行 MOV [2201H], AX时产生如下总线周期

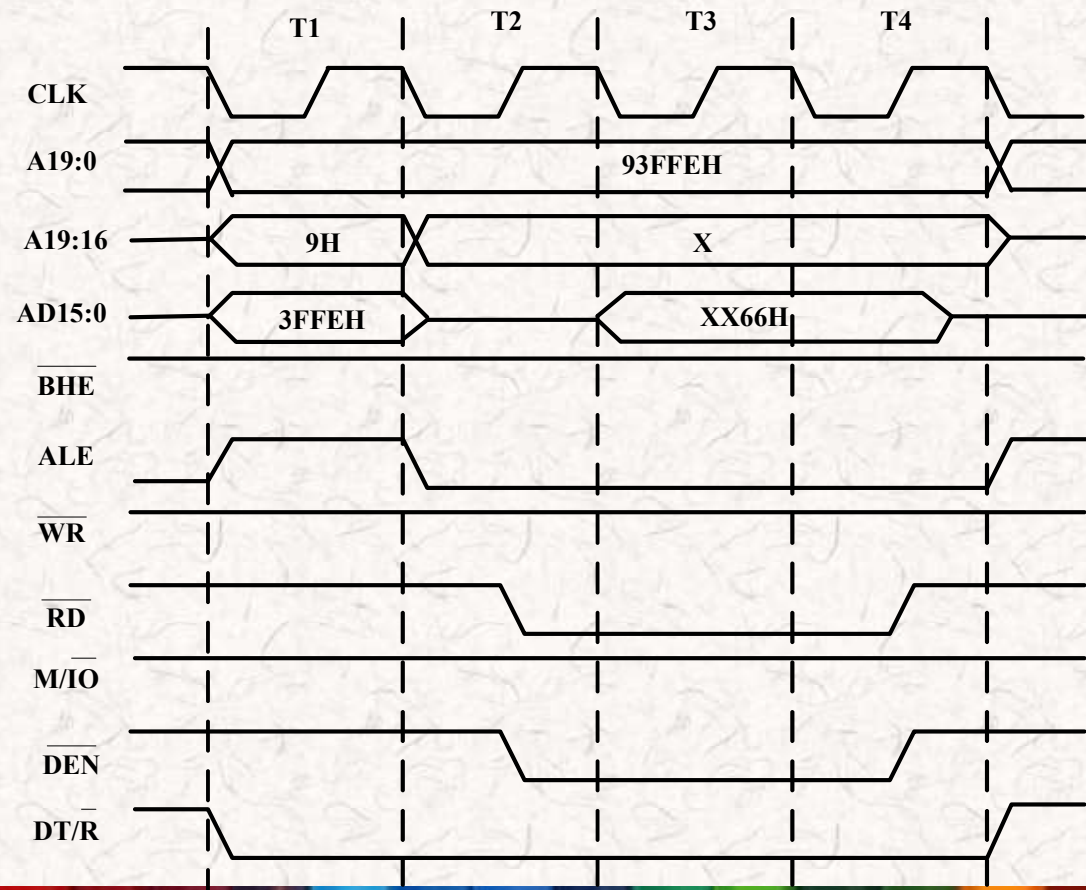


写低地址的奇地址字节

写高地址的偶地址字节

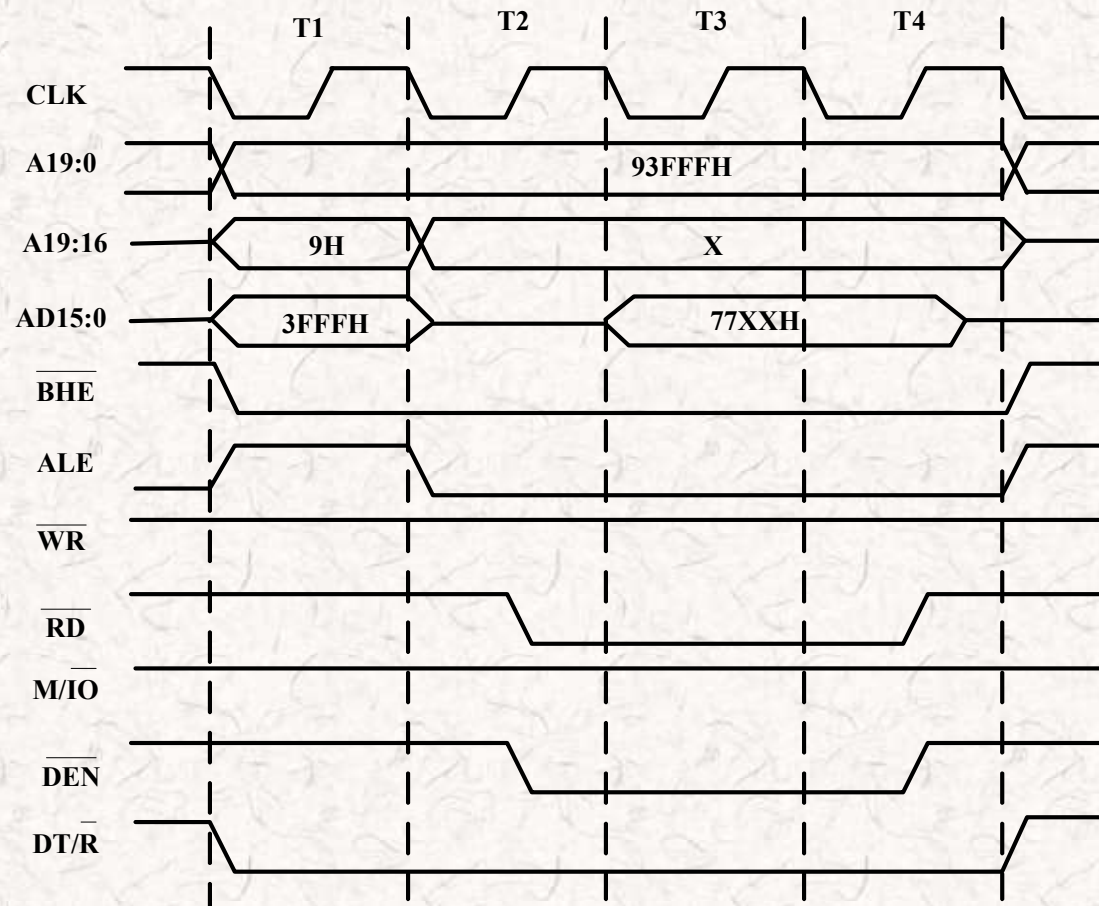
读总线操作举例1-读偶地址字节

若DS=9000H, 存储单元93FFE_H存放66H, 执行 MOV BL, [3FFE_H] 时产生如下总线周期



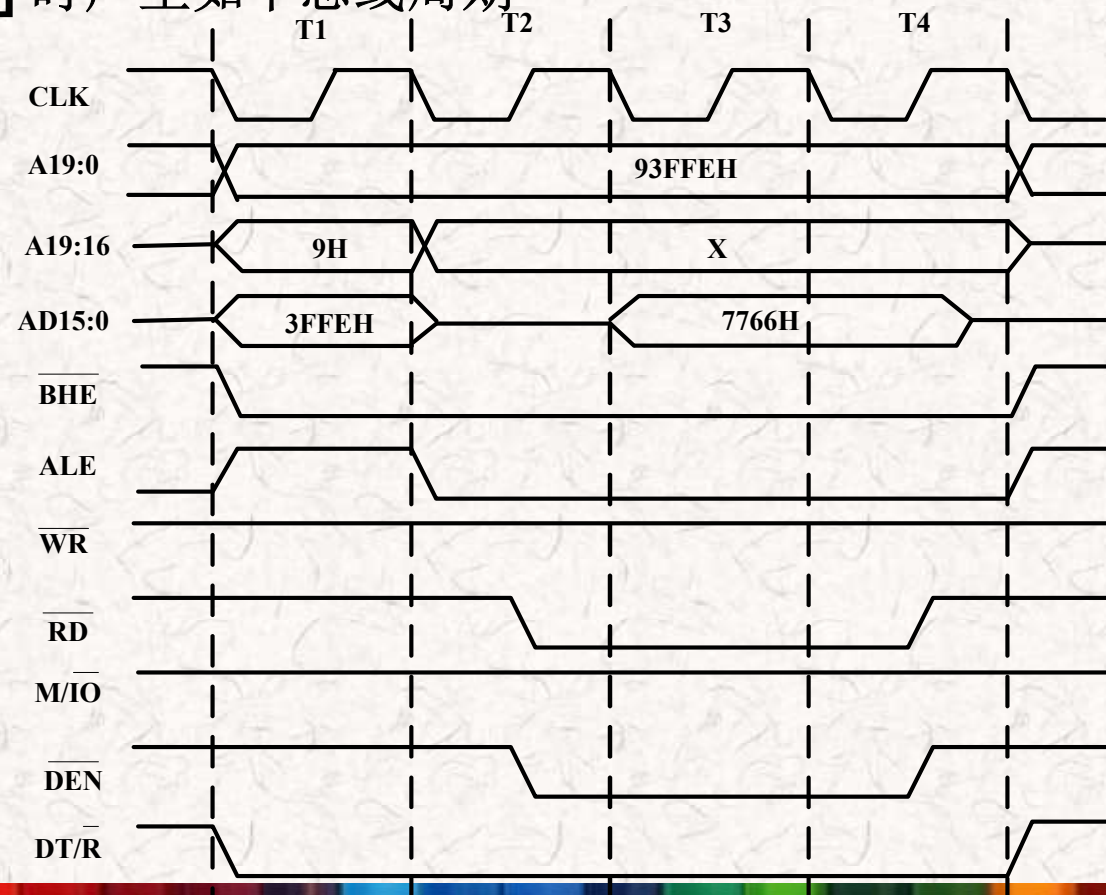
读总线操作举例2-读奇地址字节

若DS=9000H, 存储单元93FFFH存放77H, 执行 **MOV BL, [3FFFH]** 时产生如下总线周期



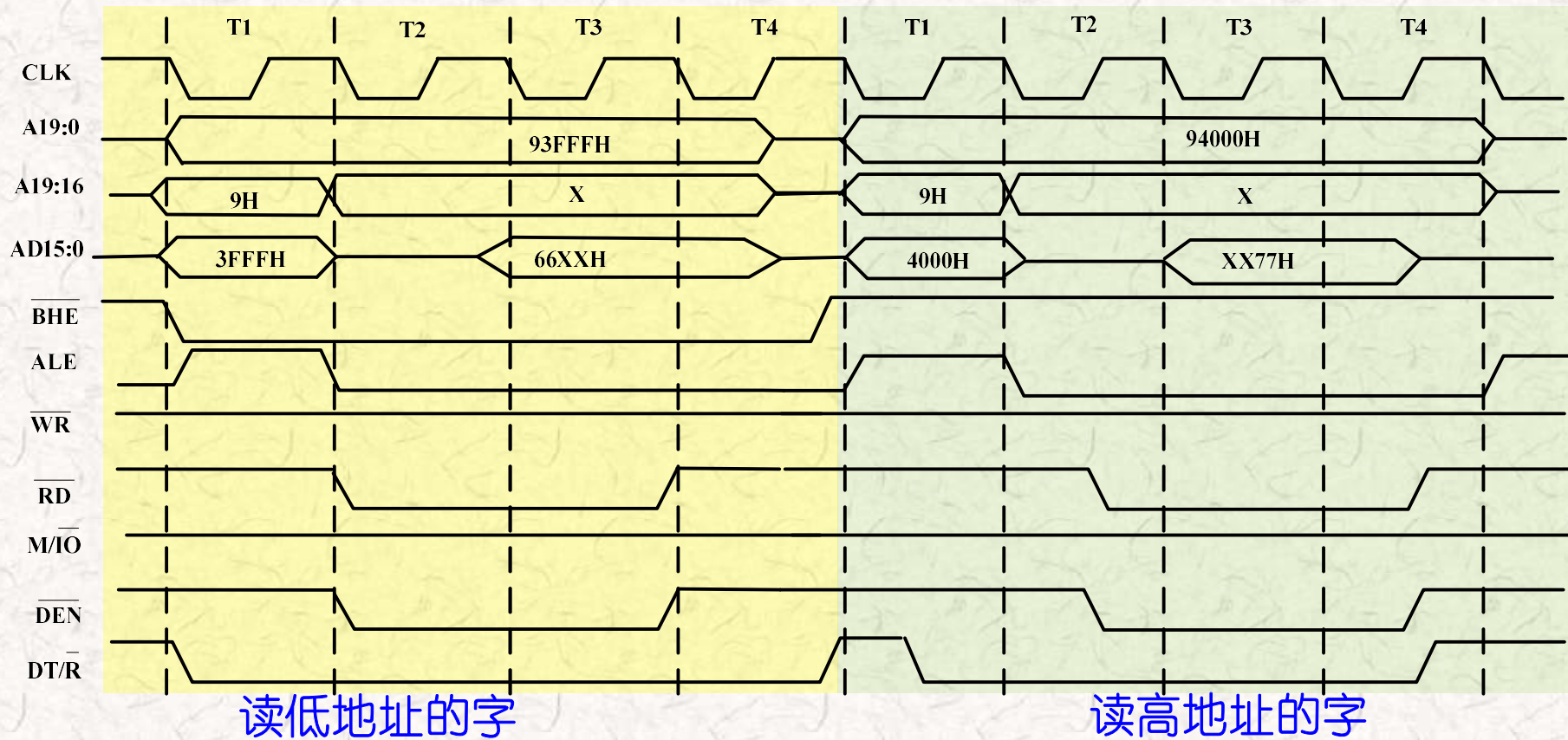
读总线操作举例3-读偶地址字

若DS=9000H, 存储单元93FFE_H存放66H, 93FFF_H存放77H, 执行MOV BX, [3FFE_H]时产生如下总线周期

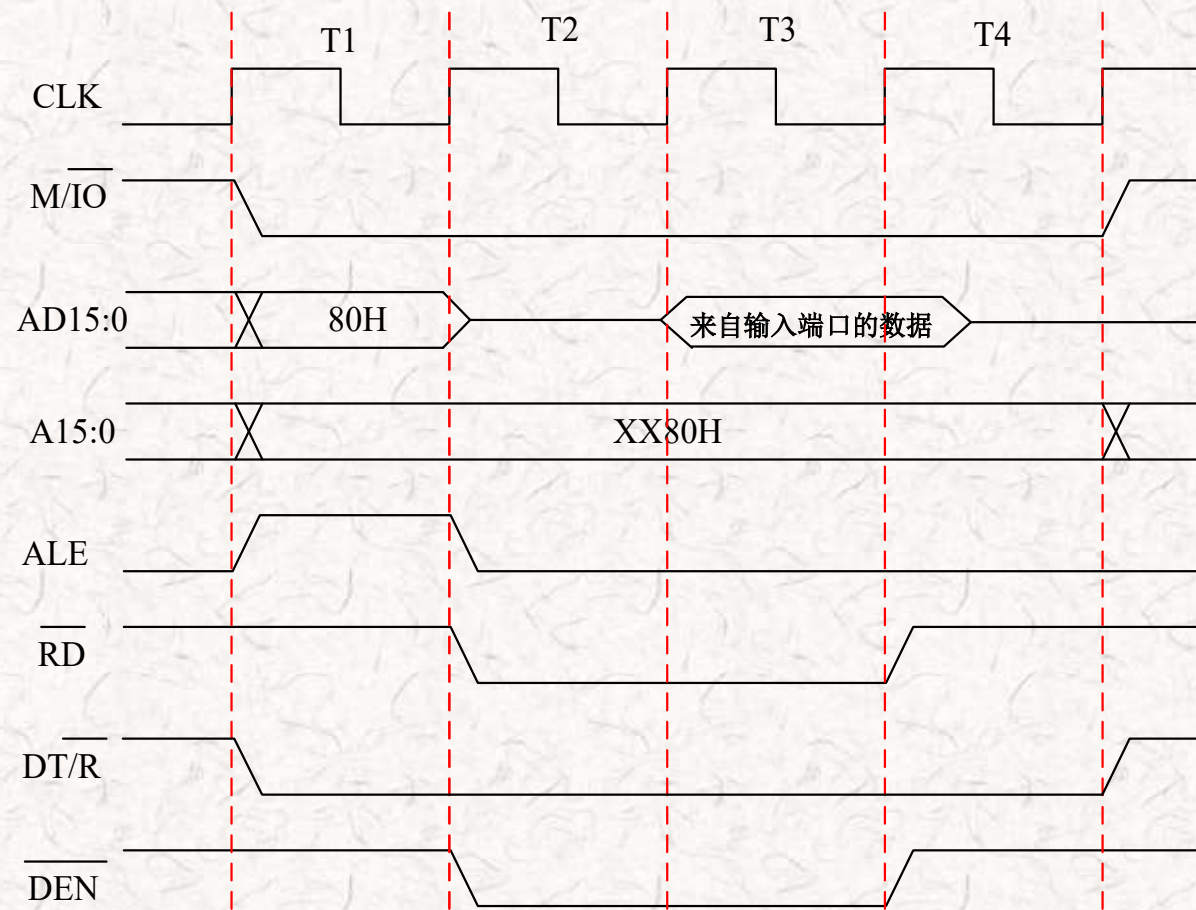


读总线操作举例4-读奇地址字

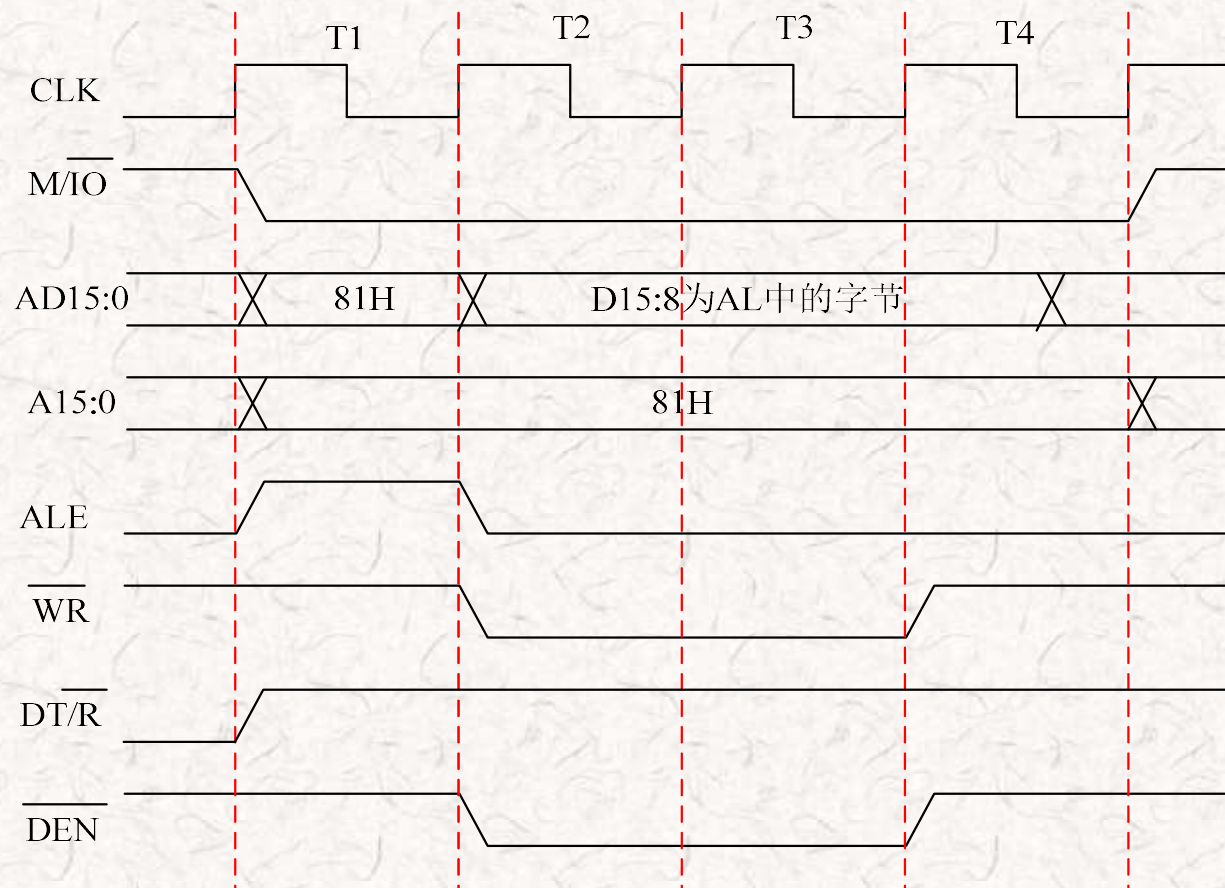
若DS=9000H, 存储单元93FFFH存放66H, 94000H存放77H, 执行MOV BX, [3FFFFH] 时产生如下总线周期



8086执行指令 **IN AL, 80H** 时总线周期



8086执行指令 OUT 81H,AL时总线周期





存储器系统设计

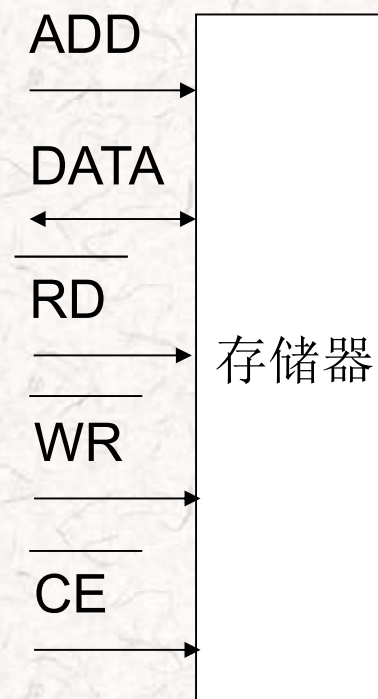


存储器接口与操作

/CE	/RD	/WR	操作
0	0	1	从ADD端口指明的单元读出nbit信息到DATA端口，DATA端口输入
0	1	0	向ADD端口指明的单元写入DATA端口输入的nbit信号，DATA端口输出驱动
1	X	X	存储器未被选中，DATA端口高阻
0	0	0	存储器无操作，DATA端口高阻

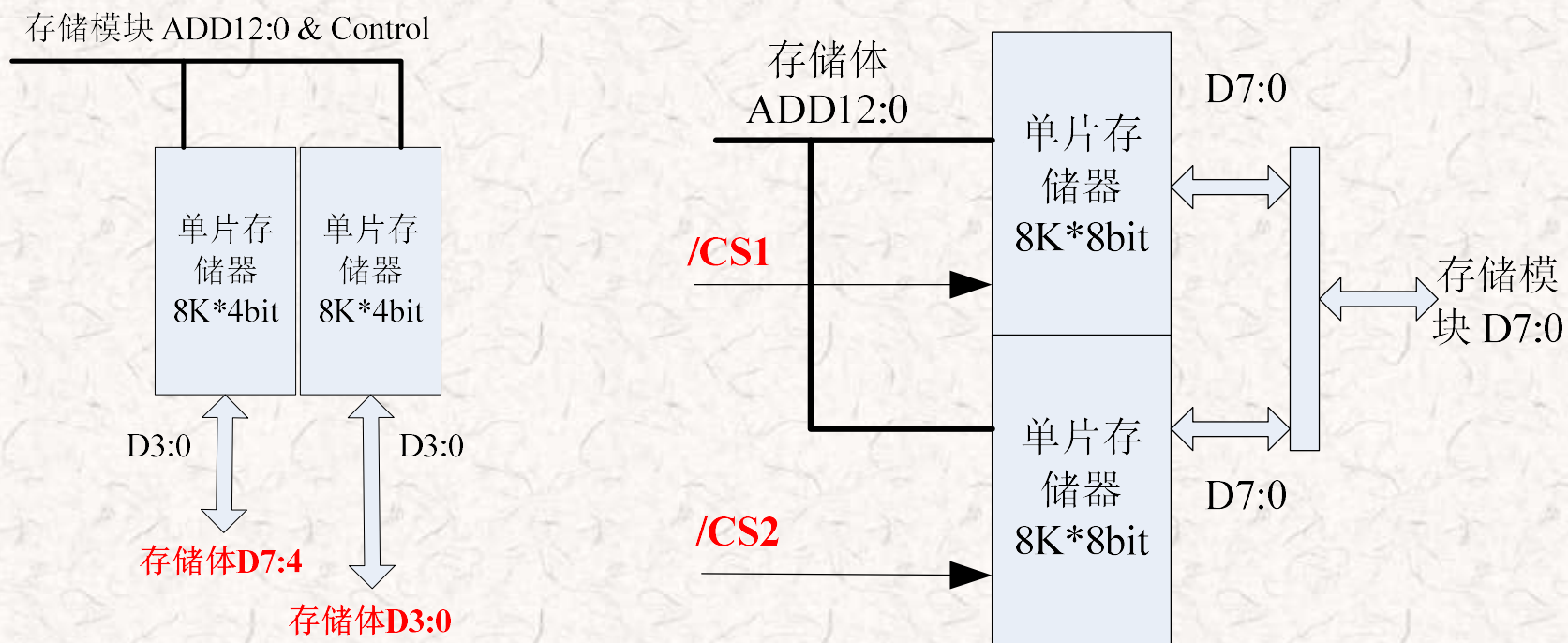
/CS连接地址译码选通逻辑，决定这片存储器可以映射在CPU的哪一段地址空间

什么时候上述信号有效：指令中有存储单元操作数，且该单元的地址位于/Cs对应地址空间时，总线上出现的总线周期将使译码电路输出有效

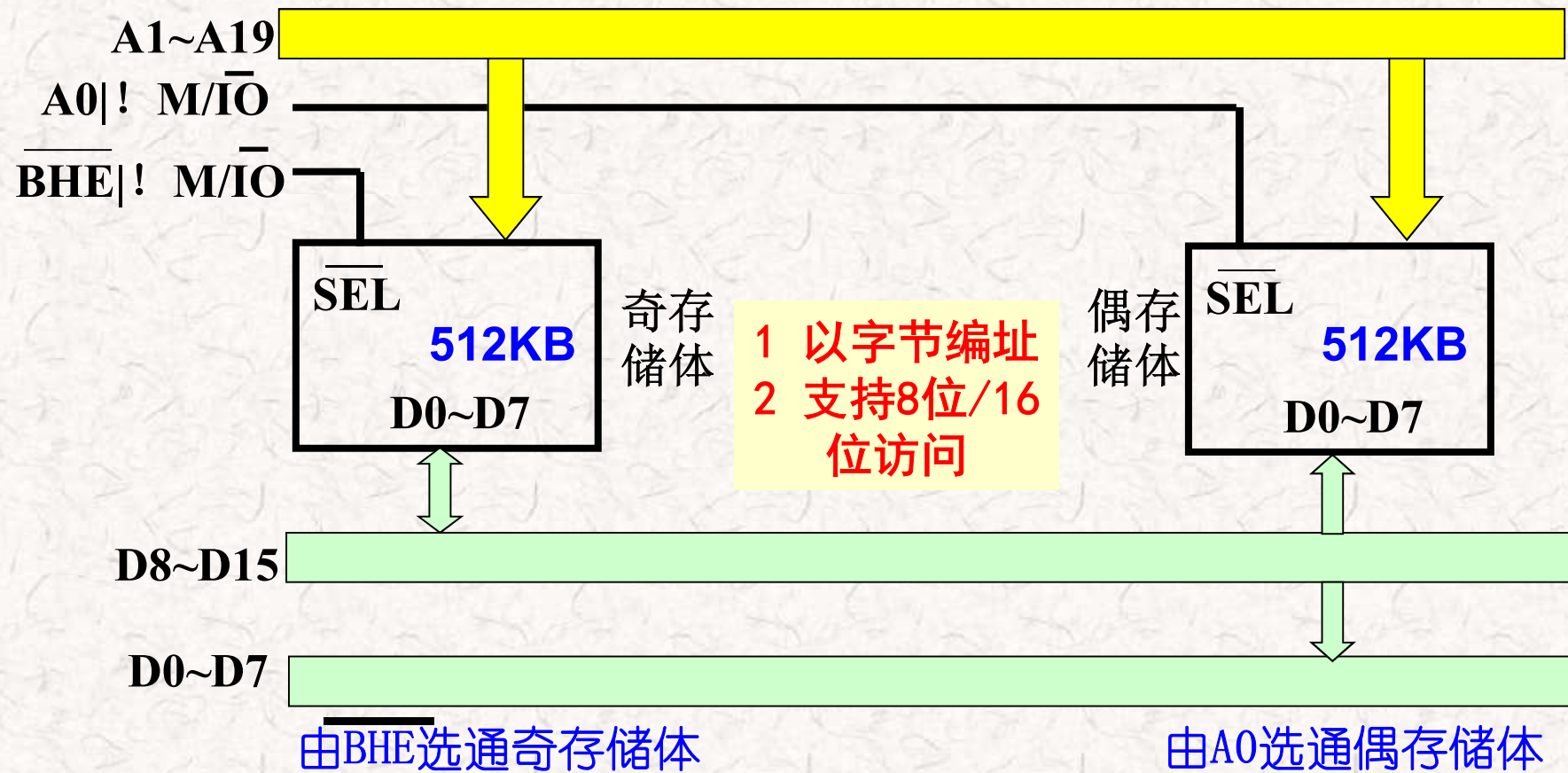


如何实现存储器的位扩展和字扩展？

- 位扩展：在相同的地址空间读写更多的位宽
- 字扩展：在不同的地址空间读写相同的位宽



8086奇偶存储体



如何为8086设计指定地址范围的 指定容量的存储器

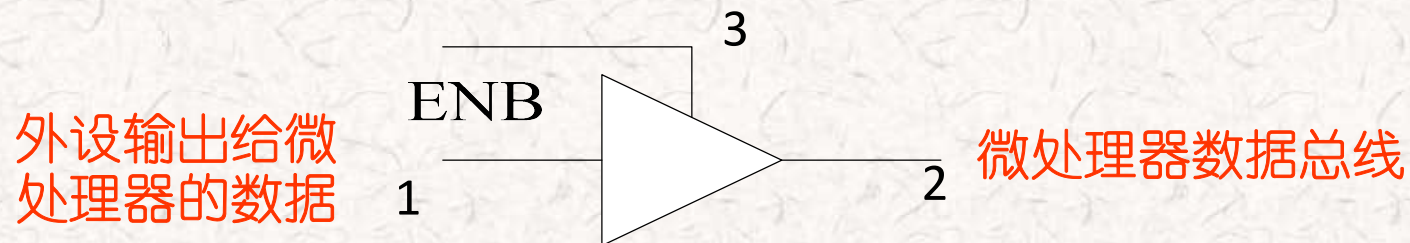
- 每个存储体容量为总容量的一半，假设为 $2^n \times 8\text{bit}$
- 确定组成单个存储体所需要的存储器数量，设单片为 $2^m \times 8\text{bit}$
- A_{m+1} 连接到每片存储器
- $A_0=0$ 和 $BHE=0$ 分别加入偶存储体和奇存储体的片选信号
- A_{n+m+1} 用于单个存储体内部字扩展
- $M/\overline{IO}=1$ & $A_{19:n+1}$ 用于地址空间分配

IO接口

输入接口

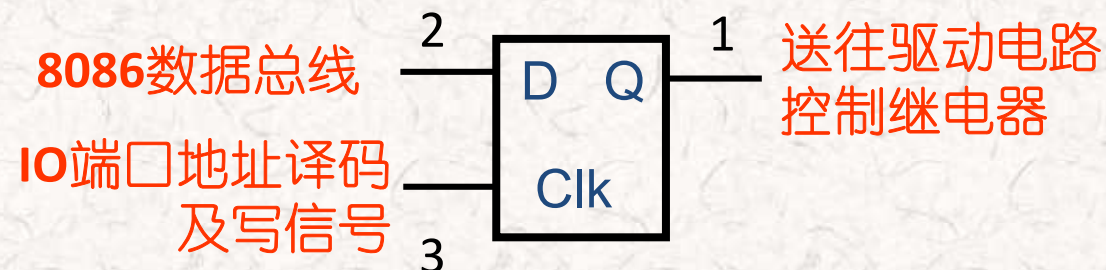
- 功能：向微处理器数据总线输出数据
- 必须由三态驱动器构成
- 访问指令：**IN AL, 端口地址**（一个字节）
- **MOV DX, 端口地址**（一个字） **IN AL, DX**

IO端口地址译码及读信号

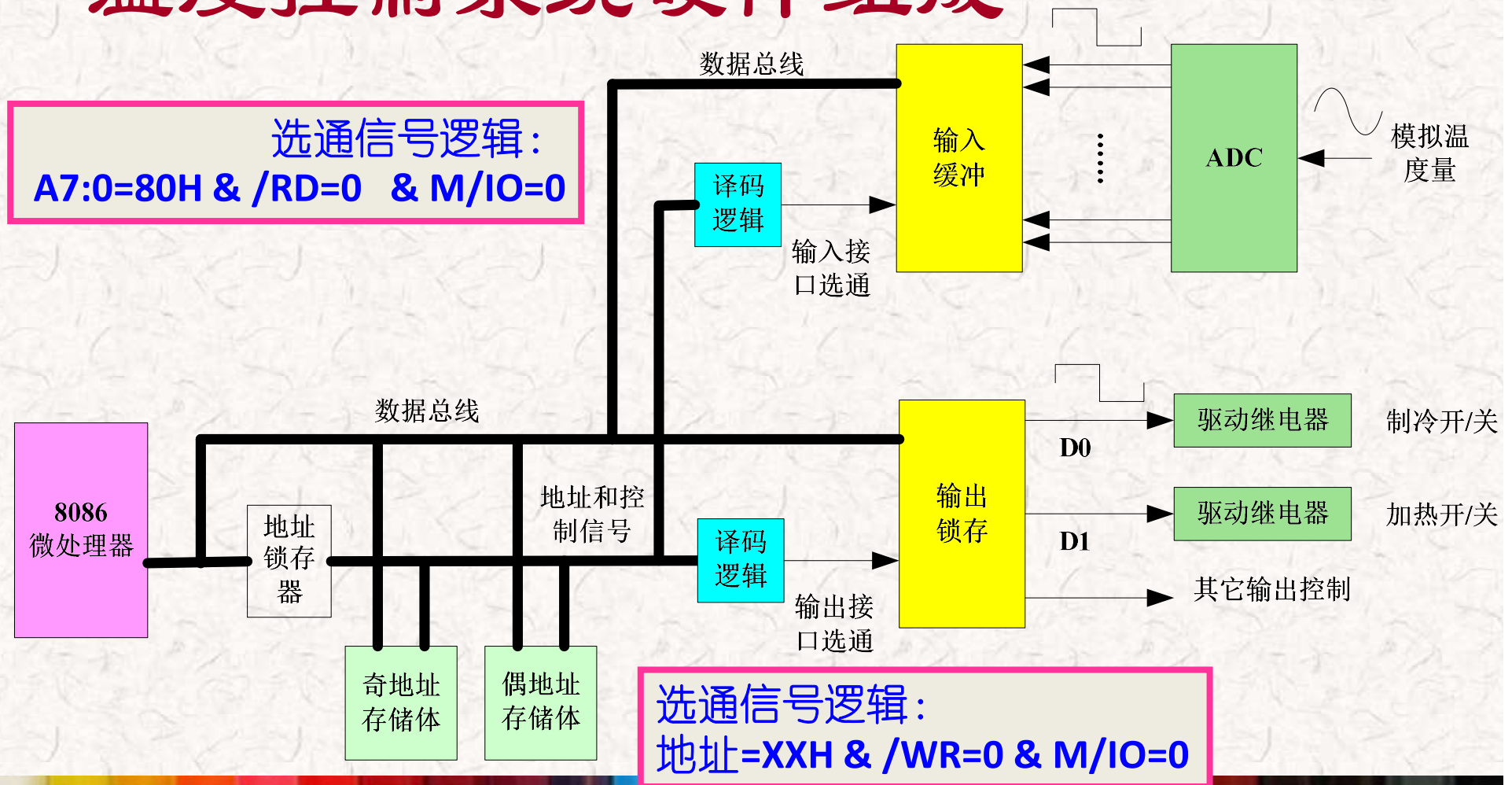


输出接口

- 功能：将微处理器希望输出到外设的数据锁存住
- 必须由锁存器、触发器等时序元件构成
- 访问指令：**OUT** 端口地址（一个字节），**AL**
- **MOV DX, 端口地址（一个字） OUT DX, AL**



温度控制系统硬件组成



中断系统

- **8086**通过中断类型号到中断向量表中寻找中断向量
- 中断类型号的来源？
- 只有**INTR**由外部电路提供中断类型号，其它都是隐含或者在软件中断指令中指定的
- **INTR**的中断类型号在何时，如何提供给**CPU**？
- 中断向量表在哪里？其中的信息如何存放？
- 如何设置中断向量表？
- 使用内中断/外中断所需要的初始化工作有哪些？
- 中断响应的过程