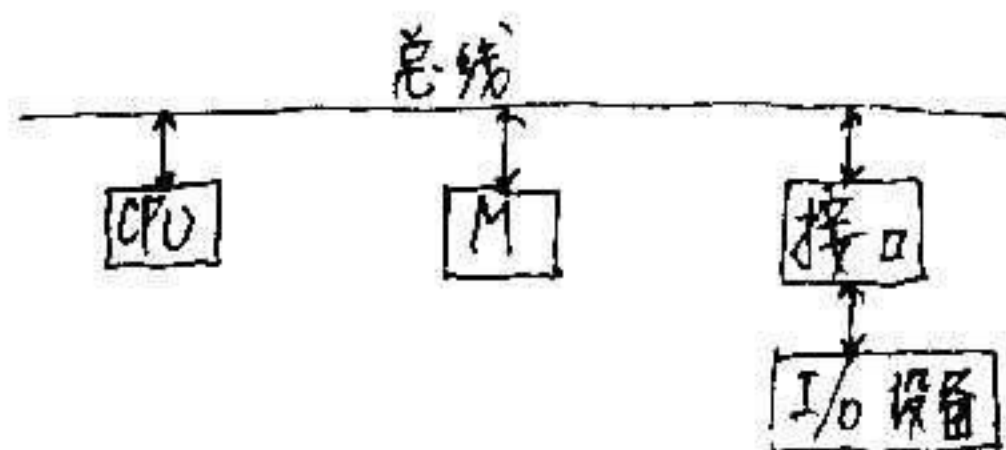


## 计算机组成原理 考研 补习



计算机概念

- 两个层次
  - CPU 整机概念
  - 硬件系统整机概念
- 两个方面
  - 逻辑组成
  - 工作机制

主要内容:

## 1. CPU

## (1) 逻辑组成

寄存器、ALU 设置、数据通路结构

## (2) 工作机制

指令的执行过程

- 寄存器传送级 · 各类指令的流程
- 微操作控制级 · 微命令序列

微命令序列

- 微命令产生方式: { 组合逻辑控制, 微程序控制 }
- 时序控制方式: 同步控制 (CPU 内部) 异步 (CPU 外部系统总线)

## 2. 常用运算方法规则和进位链

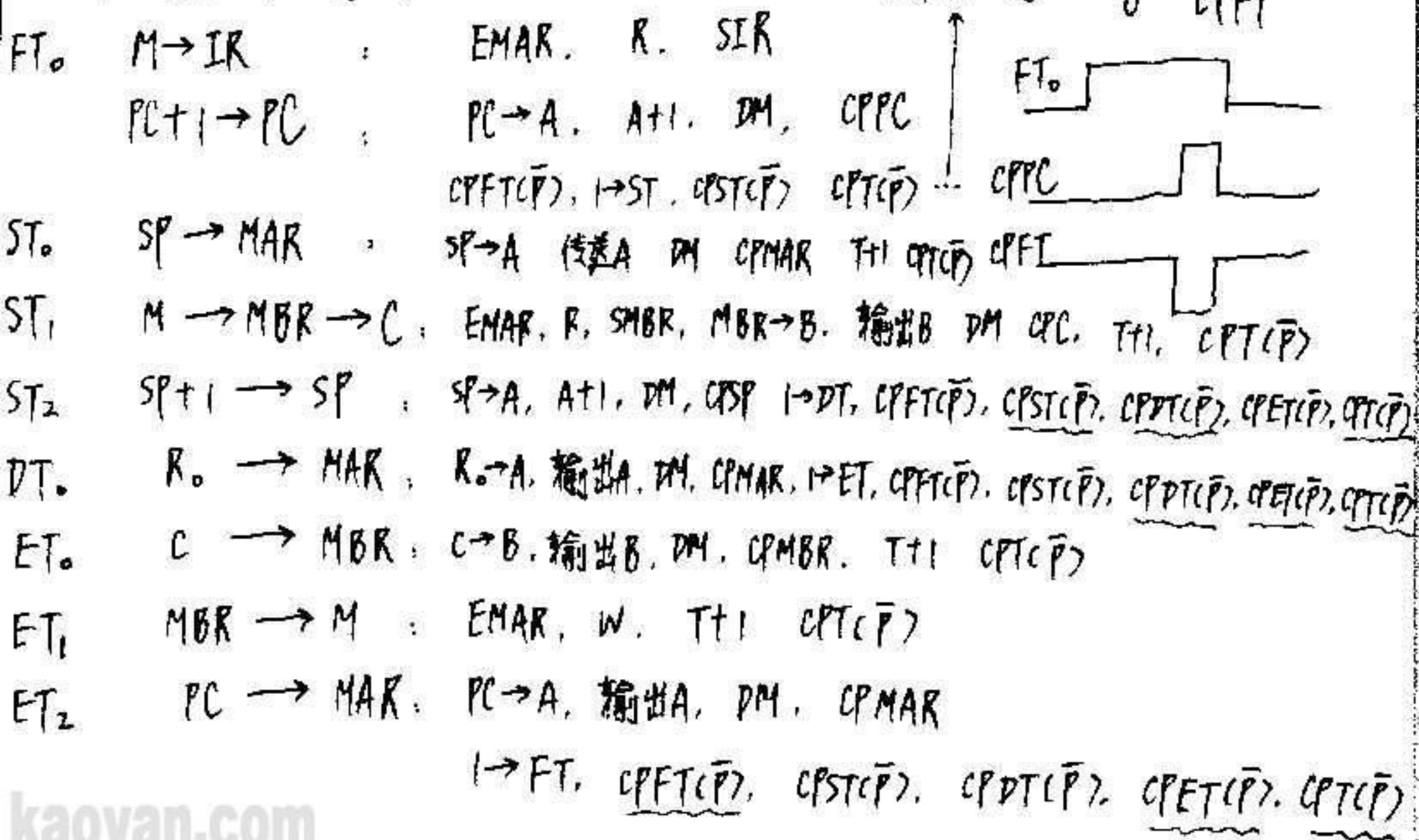
(1) 原码、补码一位乘法, 原码两位乘法, 原码、补码不恢复余数除法, 浮点运算法

(2) 串行进位、并行进位、行级同时进位

## 3. 存储器

## 2. 例题.

MOV (SP)+, (R0).



## 1.2.3 微命令的产生方式

## 1. 组合逻辑控制方式

## (1) 基本思想:

综合化简产生微命令的条件, 形成相应逻辑式, 用组合逻辑电路实现.

执行指令时, 由组合逻辑电路(微命令发生器)在相应时间发出所需微命令, 控制有关操作.

## (2) 优缺点:

优点: 速度快



缺点：设计不规整，结构零乱，不易修改，扩充指令系统功能。

(3) 应用。

用于高速计算机或小规模机器中

## 2. 微程序控制方式。

(1) 基本思想。

1) 将微命令以代码形式编成微指令，控制一步操作。

2) 若干微指令编成一段微程序，解释执行一条机器指令

3) 微程序事先存放在控制存储器(CM)中，执行机器指令时再取出。

注意区分：

微指令：产生微命令，控制完成机器指令功能的一步操作

机器指令：实现指令系统功能所规定的一种操作

微程序：包含若干微指令，解释执行一条机器指令

工作程序：包含若干机器指令，完成某一特定任务

CM：存放微程序，位于CPU之内

主存：存放工作程序，位于CPU之外。

(2) 优缺点

优点：结构规整，设计效率高，性价比高，可靠性高，易于修改、扩展指令系统功能

缺点：速度较慢，执行效率受影响。

(3) 应用

用于速度要求不是很高，功能复杂的机器中，特别适用于系列机。

## 3. 微指令

拟定微指令格式，了解微指令编码方法和微地址形成方式。

格式：字段，各段功能含义

模型机分段原则：按操作类型分段，同类互斥微命令放同一字段。

3	3	5	2	2	3	1	1	1	2	4
AI	BI	SM	Co	S	ZO	EMAR	R	W	ST	SC

### (1) CPU内部数据通路操作控制

AI: A输入选择

BI: B输入选择

SM: ALU功能选择

Co: 初始进位选择

S: 移位功能选择

ZO: 结果归零

### (2) 访存操作控制

EMAR: 地址使能

R: 读

W: 写

### (3) 辅助操作控制

ST: 开中断、关中断、SIR

### (4) 顺序控制

SC: 提供后继微地址形成方式

增量方式  
断定方式

### 1.2.4 时序控制方式

操作与时序信号的关系

同步控制方式  
异步控制方式

掌握定义、特点、应用场合。

#### 1. 同步控制方式

(1) 定义: 各项操作与统一时序信号同步。



(2) 特点: 1) 有明显的时序时间划分

2) 时钟周期(节拍)时间固定

3) 各步操作的衔接, 各部件之间的数据传送受严格同步定时控制。

(3) 应用场合: 用于CPU内部, 设备内部, 总线操作(各挂接部件速度差异小, 传送时间确定, 传送距离较近)。

## 2. 异步控制方式:

(1) 定义: 各项操作按需要安排不同时间, 不受统一时序约束。

(2) 特点: 1) 无严格时钟周期划分

2) 各操作间的衔接, 各部件之间的数据传送采取异步应答方式。

(3) 应用场合: 用于异步总线操作(各挂接部件速度差异大, 传送时间不确定, 传送距离较远)。

主设备: 申请并控制总线的设备

从设备: 响应主设备请求的设备。

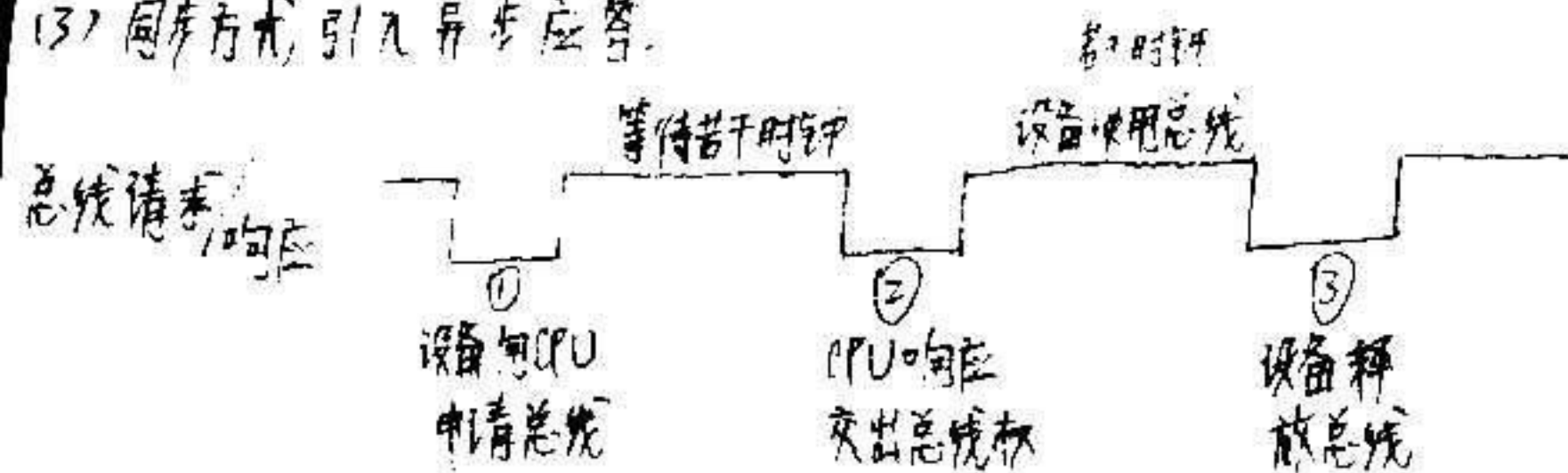
## 3. 同步方式在实际应用中的变化

(1) 不同指令安排不同时钟周期数。

(2) 总线周期中插入延长周期。



1.3 同步方式引入异步应答。



### 1.3 运算方法与运算器

#### 1.3.1 运算方法

##### 1. 原码一位乘法

##### (1) 原码运算

操作数和结果用原码表示，绝对值参加运算，符号单独处理

##### (2) 算法规则。

用乘数末位作判断位

##### 2. 补码一位乘法

##### (1) 补码运算

操作数和结果用补码表示，符号位参加运算

##### (2) 算法规则

乘数末位设置附加位，两位判断位。

##### 3. 原码两位乘法

如何处理  $\div 3 \times$  操作

##### 4. 原码不恢复余数除法

算法规则：根据余数的正负决定上商及下一步操作。



## 5. 补码不恢复余数除法

算法规则：根据余数与除数同号或异号决定上商及下一步操作。

## 6. 浮点运算

加减法运算过程：

(1) 判操作数是否为0

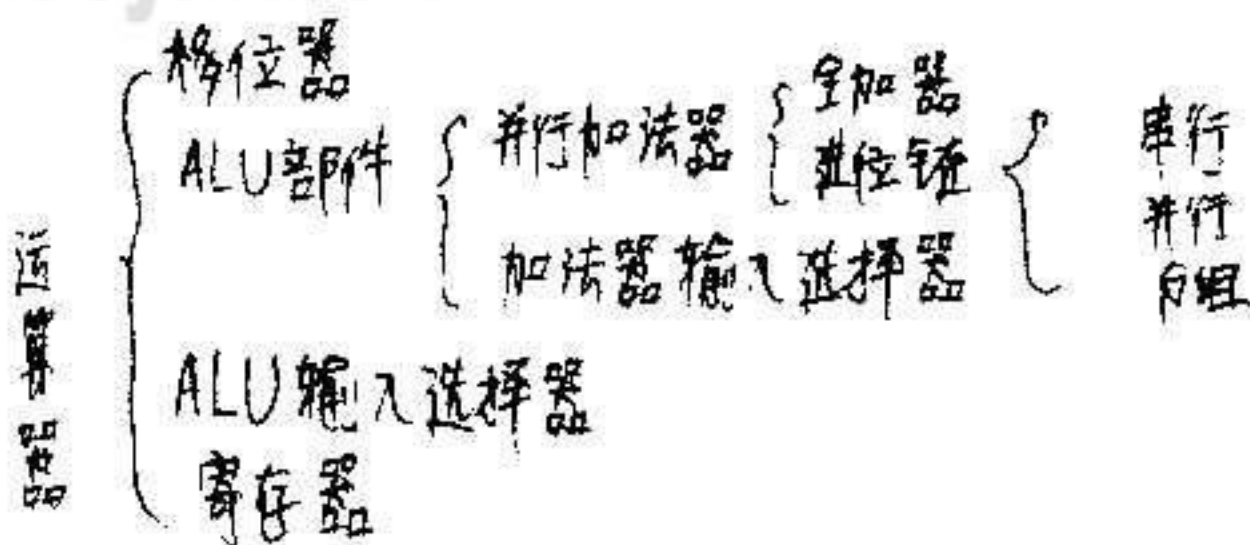
(2) 对阶 { 原则：小阶向大阶对齐  
操作：小阶增大，尾数右移

(3) 尾数加、减

(4) 结果规格化

{ 左规 (尾数绝对值小于  $\frac{1}{2}$ )：尾数左移，阶码-1  
右规 (尾数绝对值大于  $1$ )：尾数右移，阶码+1

## 1.3.2 运算器



## 2. 进位逻辑

写出任一位  $C_i$  的进位逻辑

(1) 串行进位

(2) 并行进位

(3) 数组、各级同时进位 (求级写，直至  $A_i, A_i, C_0$ )

串:  $C_6 = A_6 + F_6$

并:  $C_6 = A_6 + F_6 A_5 + F_6 F_5 A_4 + \dots$

数组:  $C_6 = A_6 + F_6 A_5 + F_6 F_5 A_4$

$$C_I = A_I + P_I C_0$$

$$A_I = A_4 + P_4 A_3 + P_4 P_3 A_2 + P_4 P_3 P_2 A_1$$

$$P_I = P_4 P_3 P_2 P_1$$

$$A_i = A_i B_i \quad P_i = A_i \oplus B_i$$

#### 1.4 其他基本概念

(1) 溢出及判断方法、扩展操作码、地址结构、隐地址、显地址等。

(2) I/O 指令的设置

{ 显式 I/O 指令  
隐式 I/O 指令

(3) I/O 指令的功能扩展

{ I/O 指令中留有扩展余地

{ 用通用 I/O 指令或 MOV 指令访问接口中的控制/状态寄存器

(4) 主机对外设的寻址方式

{ 单独编址 (为接口寄存器分配端口地址)

{ 统一编址 (为接口寄存器分配总线地址)

## 第二章 存储子系统

### 2.1 半导体存储器逻辑设计

1. 主要解决: 芯片的选用, 地址分配, 片选逻辑, 信号线的连接

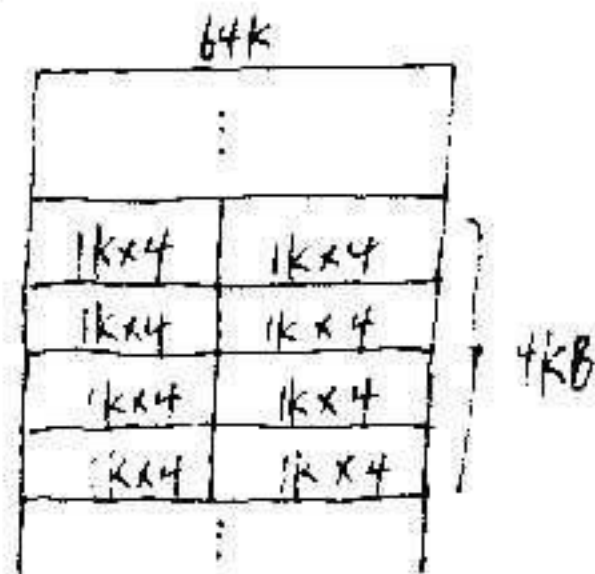


例1. 用SRAM芯片(1K×4位)组成16K存储器, 地址总线A<sub>15</sub>~A<sub>0</sub>线, 双向数据总线D<sub>7</sub>~D<sub>0</sub>线, 读/写信号线R, W.

(1) 芯片数: 8片

(2) 存储空间安排: 任意连续区间

(3) 地址分配和片选逻辑:

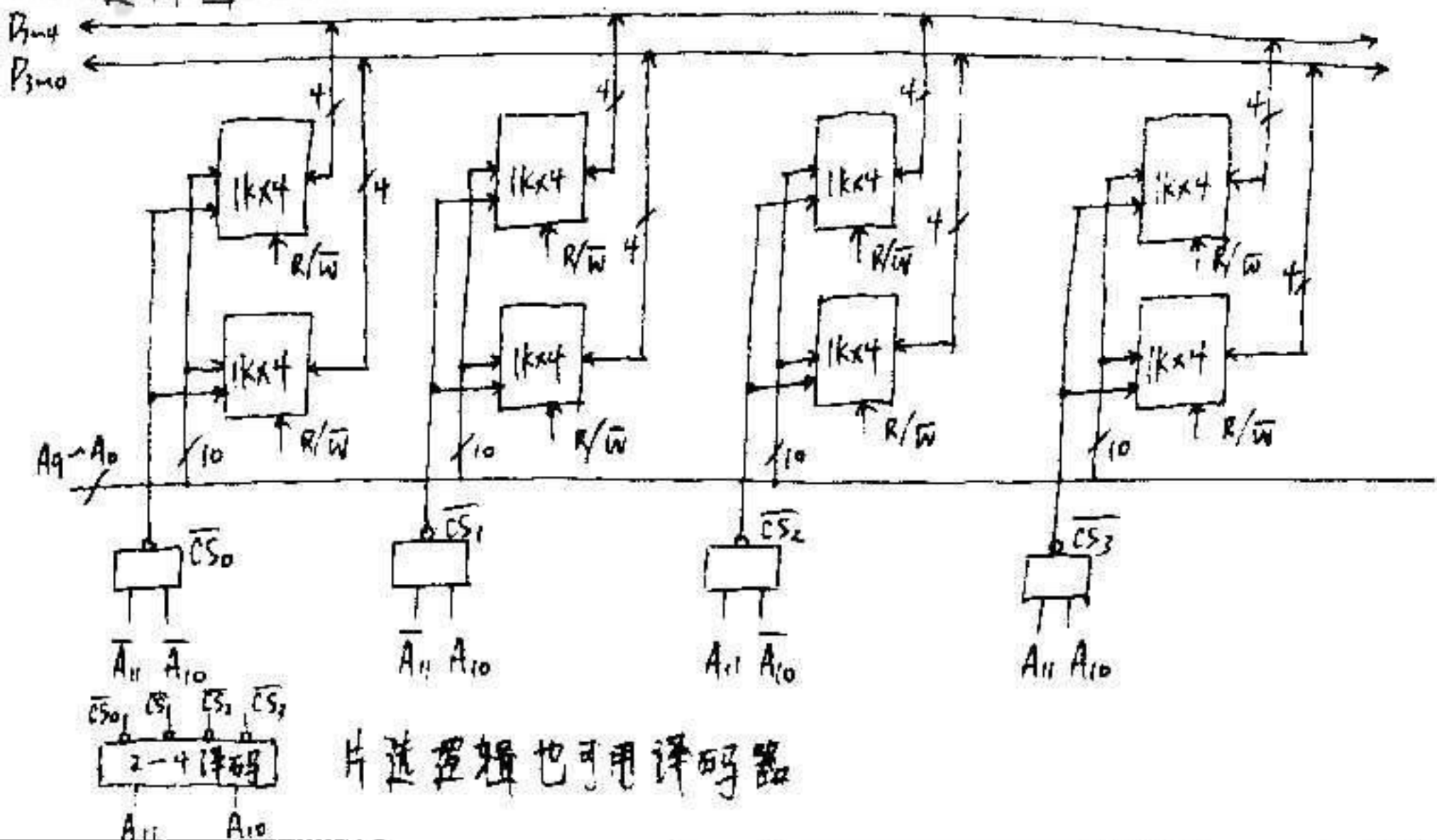


4KB: 12位地址A<sub>11</sub>~A<sub>0</sub>.

哪几位分配给芯片? 哪几位形成片选逻辑?

芯片	芯片地址	片选逻辑
1KB	A <sub>9</sub> ~A <sub>0</sub>	$CS_0 = \bar{A}_{11} \bar{A}_{10}$
1KB	A <sub>9</sub> ~A <sub>0</sub>	$CS_1 = \bar{A}_{11} A_{10}$
1KB	A <sub>9</sub> ~A <sub>0</sub>	$CS_2 = A_{11} \bar{A}_{10}$
1KB	A <sub>9</sub> ~A <sub>0</sub>	$CS_3 = A_{11} A_{10}$

(4) 逻辑图



例2. 用 4KB ROM 芯片,  $2k \times 4$  位和  $1k \times 4$  位 RAM 芯片 组成 7KB 存储器。

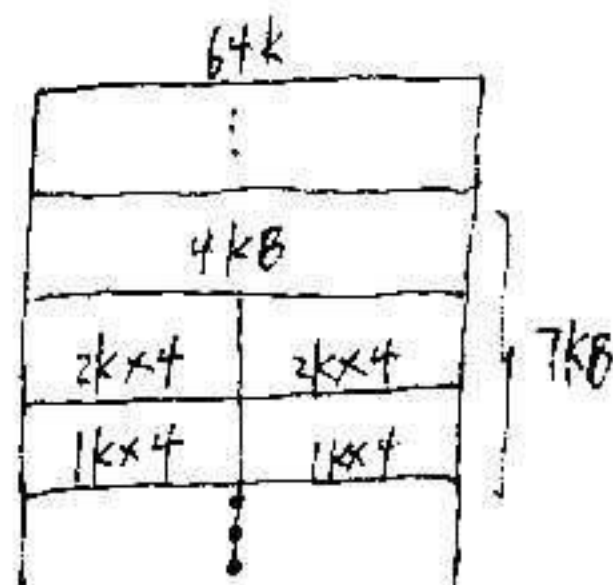
地址总线  $A_{15} \sim A_0$  16 位, 双向数据总线  $D_7 \sim D_0$  8 位, 读写信号线  $R, W$ 。

(1) 芯片数: 5 片

(2) 存储空间安排:

连续区间, 先安排大容量

芯片, 后安排小容量芯片

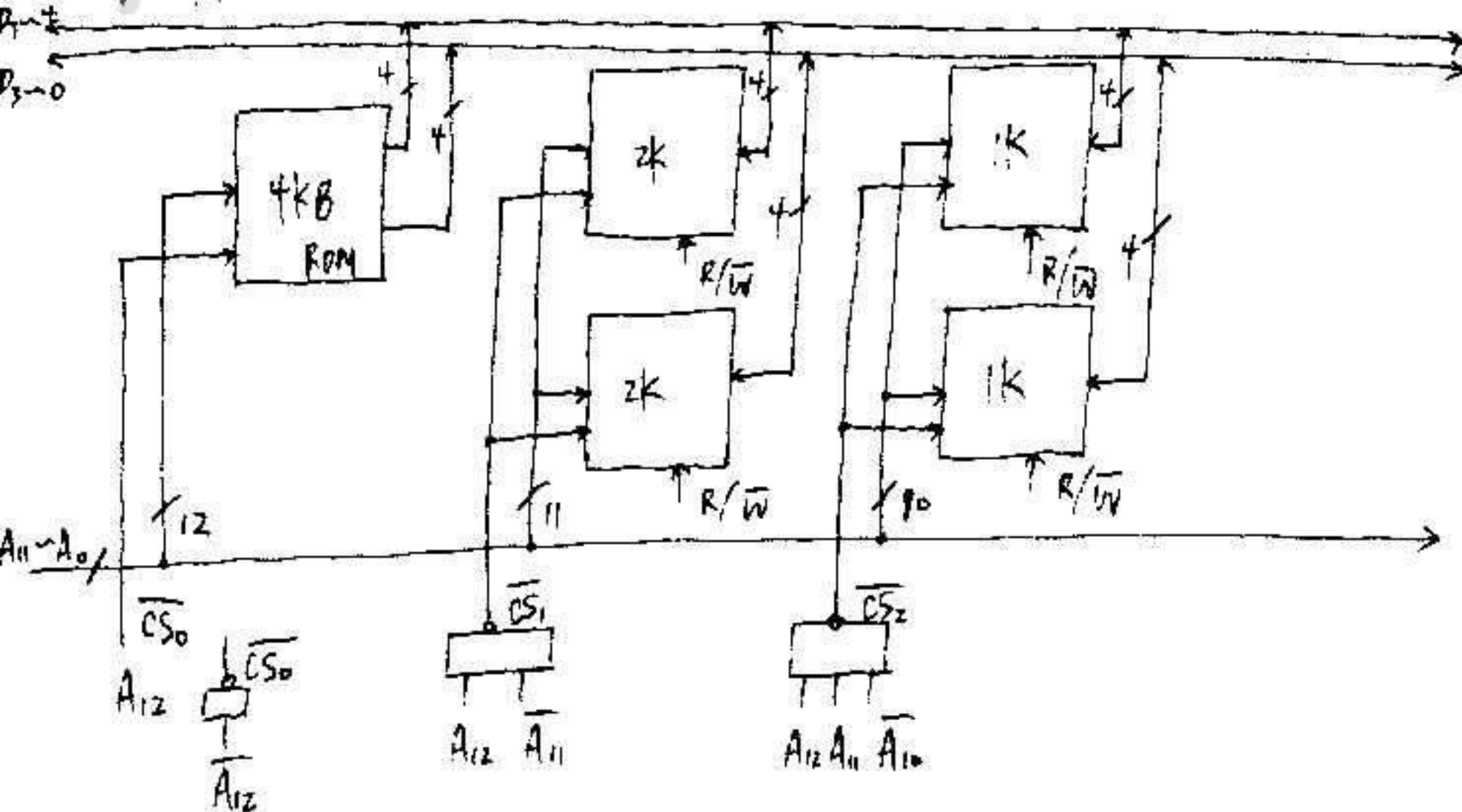


(3) 芯片地址分配与片选逻辑:

7KB: 13 位地址  $A_{12} \sim A_0$

芯片	芯片地址	片选逻辑
4KB	$A_{11} \sim A_0$	$CS_0 = \bar{A}_{12}$
2KB	$A_{10} \sim A_0$	$CS_1 = A_{12} \bar{A}_{11}$
1KB	$A_9 \sim A_0$	$CS_2 = A_{12} A_{11} \bar{A}_{10}$

(4) 逻辑图



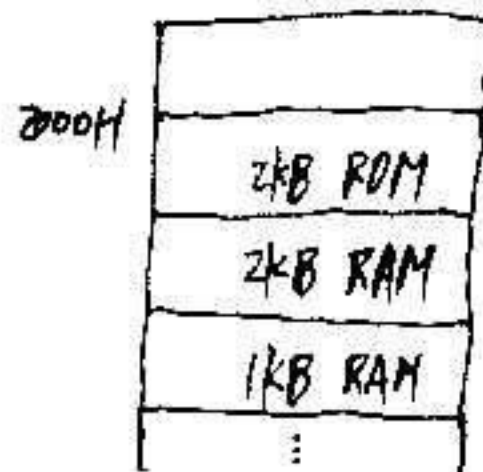


例3. 存储器空间分配: ROM区: 2000H ~ 27FFH  
(按字节编址) RAM区: 2800H ~ 33FFH

选用芯片: EPROM 2KB/片 RAM 2KB/片, 1KB/片

ROM区: 7FF:  $\underbrace{111}_{2^3} \underbrace{1111}_{2^4} \underbrace{1111}_{2^4} = 2^3 - 1 = 2^4 - 1$

RAM区:  $\frac{33FF}{\rightarrow 2800}$  BFF:  $\underbrace{1011}_{2^4} \underbrace{1111}_{2^4} \underbrace{1111}_{2^4} = 2^4 + 2^4 = 3K$



(1) 计算容量, ROM区: 2KB RAM区: 3KB

(2) 芯片数, EPROM 1片, RAM: 2片

(3) 芯片地址分配与片选逻辑: 5KB: 13位地址  $A_{12} \sim A_0$

芯片      芯片地址

2KB       $A_{10} \sim A_0$

2KB       $A_{10} \sim A_0$

1KB       $A_9 \sim A_0$

片选逻辑

$$CS_0 = \overline{A_{12}} \overline{A_{11}}$$

$$CS_1 = \overline{A_{12}} A_{11}$$

$$CS_2 = A_{12} \overline{A_{11}} \overline{A_{10}}$$

(16根地址线)

$$A_{15} \sim A_{13} \text{ 为 } \overline{A_{15}} \overline{A_{14}} A_{13}$$

$$\text{即: } 001$$

(1) 基本概念

(2) 半导体存储器的逻辑设计

芯片地址分配、片选逻辑、框图

4. 总线

(1) 基本概念

(2) 系统总线：信号组成、时序控制方式

5. 接口

(1) I/O传送的控制机制

中断：基本概念、中断控制器与接口、中断过程

DMA：基本概念、DMA控制器与接口、DMA过程

(2) 接口设计

接口组成、状态命令字和状态字格式、扩展中断源

6. 常用外设原理

(1) 键盘：键码转换方式  $\begin{cases} \text{硬件扫描} \\ \text{软件扫描} \end{cases}$

(2) CRT显示器：VRAM与屏幕显示的对应关系。

(VRAM内容和容量、地址组织、信息转换、同步计数器的设置)

(3) 打印机：信息转换、调用过程(中断方式)

(4) 磁盘：信息分布与寻址信息、磁记录方式、调用过程(DMA方式)

速度指标和容量指标。

题型：综合设计 50%

单选 }  
简答 } 50%

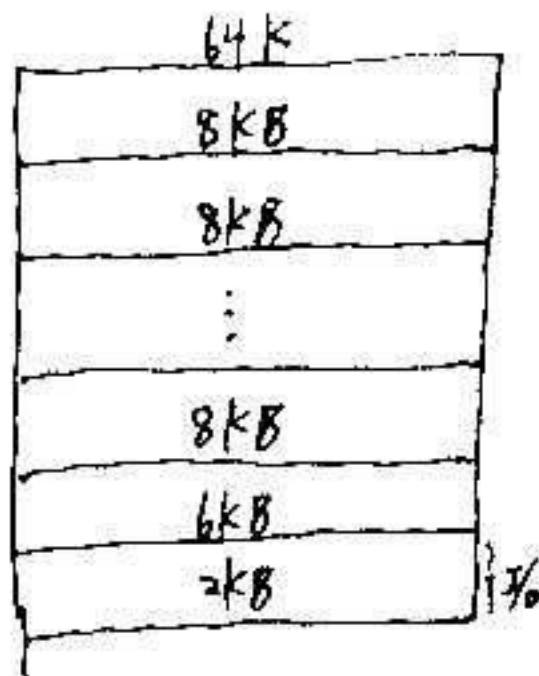


例4. 主存64KB, 其中高地址区2KB用于I/O空间, 选用芯片8KB/片.

(1) 芯片数: 8片

(2) 芯片地址分配与片选逻辑:

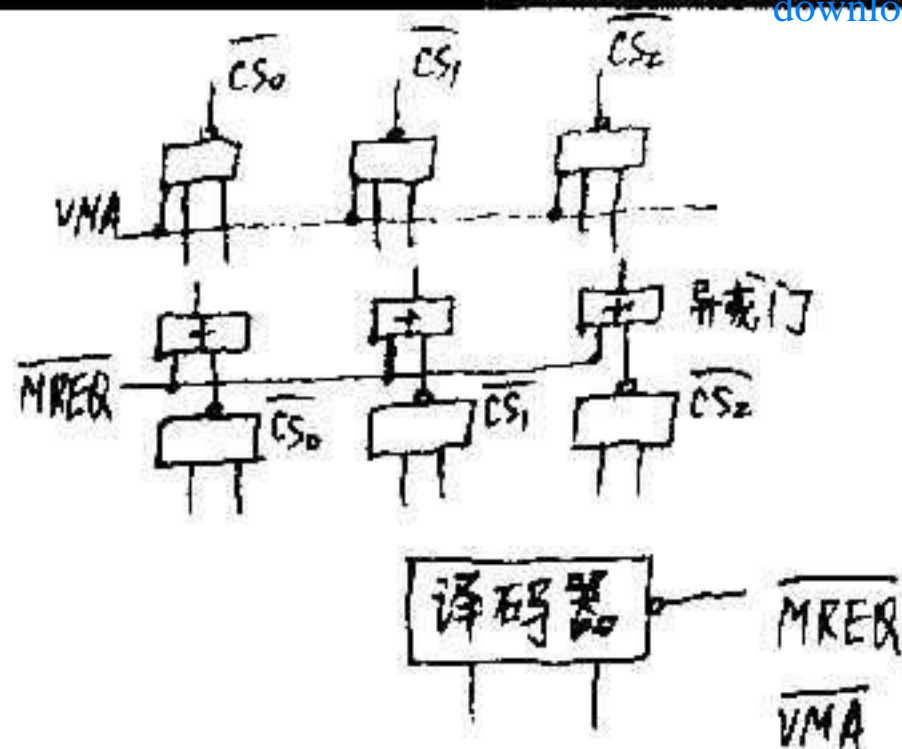
64KB: 16位地址  $A_{15} \sim A_0$



芯片	芯片地址	片选逻辑
8KB	$A_{12} \sim A_0$	$CS_0 = \bar{A}_{15} \bar{A}_{14} \bar{A}_{13}$
8KB	$A_{12} \sim A_0$	$CS_1 = \bar{A}_{15} \bar{A}_{14} A_{13}$
8KB	$A_{12} \sim A_0$	$CS_2 = \bar{A}_{15} A_{14} \bar{A}_{13}$
⋮	⋮	⋮
8KB	$A_{12} \sim A_0$	$CS_6 = A_{15} A_{14} \bar{A}_{13}$
8KB	$A_{12} \sim A_0$	$CS_7 = A_{15} A_{14} A_{13} \overline{A_{12} A_{11}} \quad (A_{12} A_{11} \text{ 不是 } 11)$

## 2. 其他问题

(1) 有关控制信号 ( $VMA$ ,  $\overline{MREQ}$  等)



(2) 地址复用技术

(3) 存储空间分配可以不连续, 相应片选逻辑也不连续

## 2-2 基本概念

### 1. 存储原理

SRAM: 利用双稳态触发器内部交叉反馈存储信息.  $\rightarrow$  功耗大

DRAM: 利用电容存储电荷存储信息.  $\rightarrow$  需动态刷新

### 2. 动态刷新

(1) 定义: 按所存信息定期向电容补充电荷.

(2) 方式: 按行读一遍.

(3) 刷新周期安排方式:

集中刷新, 分散刷新, 异步刷新

### 3. 存取方式

(1) 随机存取方式 (ROM属于随机存取! 堆栈不是随机存取方式!)

1) 可按地址直接访问任一单元

2) 访问时间与单元地址无关



## (2) 顺序存取方式

访问时按顺序查找, 访问时间跟数据所在位置有关。

## (3) 直接存取方式

访问时先直接指向一个小区域, 再按顺序查找, 访问时间与数据所在位置有关。

# 第三章 I/O 子系统

## 3.1 总线与接口的基本概念

### 3.1.1 总线

1. 定义: 一组能为多个部件分时共享的公共信息传送线路。

2. 分类

#### (1) 按功能分类

1) CPU 内总线: CPU 芯片内寄存器和运算部件之间互连的总线

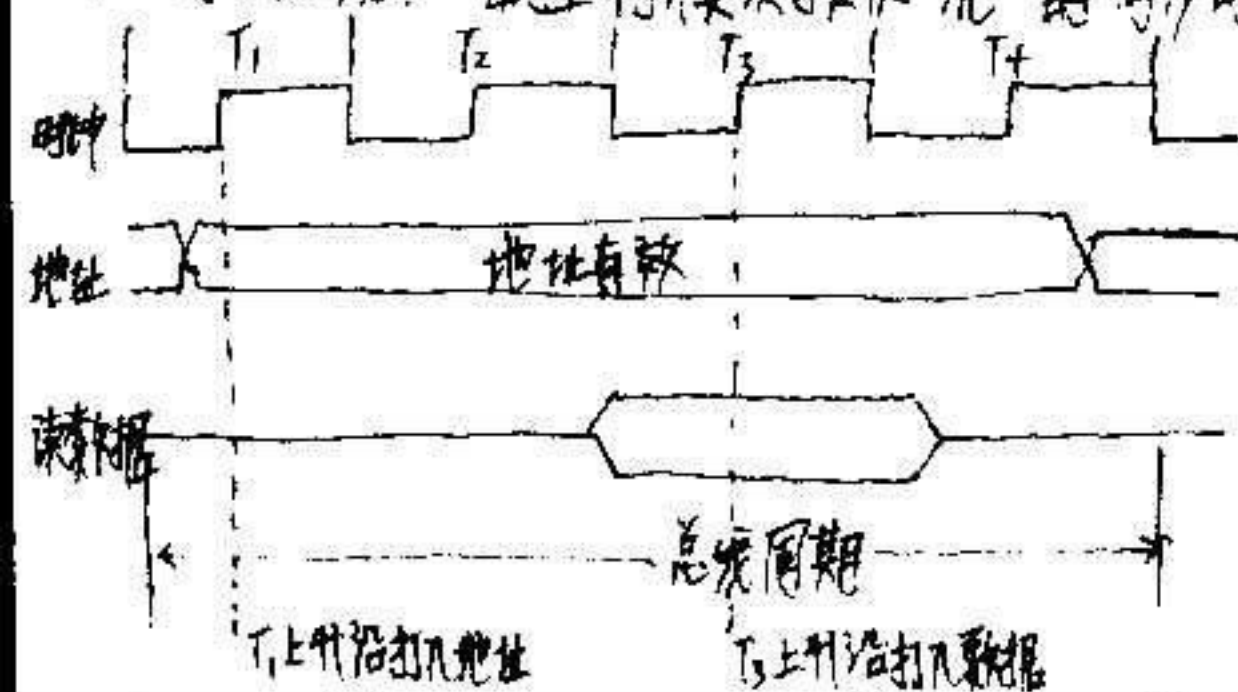
2) 部件内总线: 插件板内各芯片之间互连的总线

3) 系统总线: 计算机系统内各功能部件之间或各插件板之间互连的总线。

4) 外总线: 计算机系统之间, 或计算机系统与其他系统之间互连的总线。

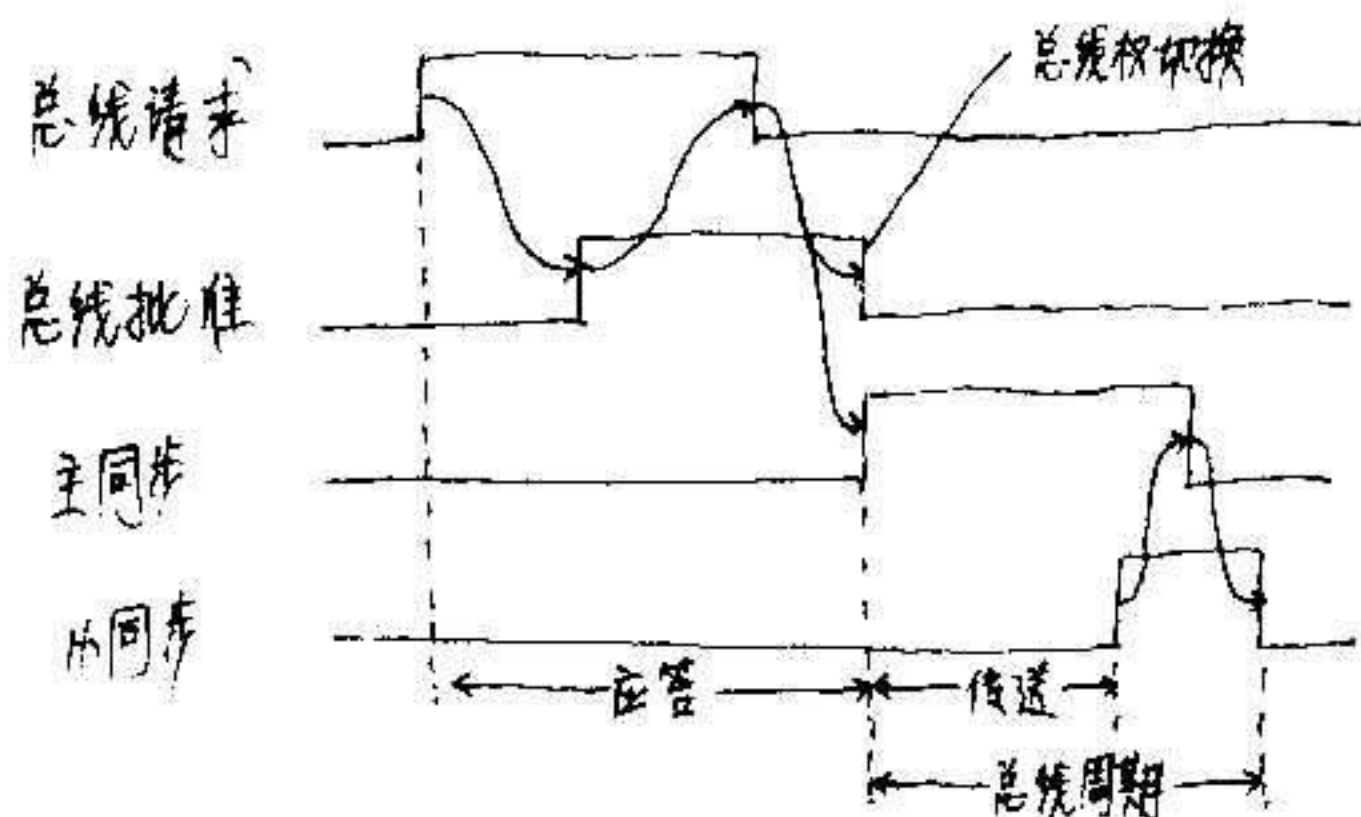
#### (2) 按时序控制方式分类

1) 同步总线: 由控制模块提供统一的同步时序信号控制总线传送操作。





2) 异步总线: 不采用统一时钟周期划分, 根据传送需要, 决定总线周期长度, 以异步应答方式控制总线传送操作。



3) 扩展同步总线: 以时钟周期为时序基础, 允许总线周期中的时钟数可变。

(3) 按数据传送格式划分

1) 并行总线: 同时传送各位信息。

2) 串行总线: 分时逐位传送各位信息。

3. 系统总线的信号组成

电源线、地址线、数据线、控制线。

时序: 时钟、定时、应答

数据传输控制: M/读/写 I/O 读/写

中断请求、响应

总线请求、响应

复位...

3.1.2 接口

1. 定义: 连接两个设备(硬、软)之间的连接部件

2. 分类





## (1) 按数据传送格式划分

### 1) 并行接口

接口与系统总线、接口与外设均按并行方式传送数据

### 2) 串行接口

接口与系统总线并行传送, 接口与外设串行传送。

## (2) 按时序控制方式划分

### 1) 同步接口

[磁盘采用串行接口, 不是同步接口]

接口与系统总线的信息传送由统一时序信号控制

### 2) 异步接口

接口与系统总线的信息传送采用异步应答方式

## (3) 按 I/O 传送控制方式划分

### 1) 直接程序传送接口

### 2) 中断接口 (可采用查询方式)

### 3) DMA 接口 (可插入中断作 DMA 善后处理)

## 3-2 I/O 传送控制机制

### 3-2-1 程序中断方式

#### 1. 定义及应用

##### (1) 定义

CPU 暂时中止现行程序的执行, 转去执行为某个随机事件服务的中断处理程序, 处理完毕后自动恢复原程序的执行。



## (2) 实质

程序切换：时间：一条指令结束时不换  
方法：保存断点、现场；恢复现场，返回断点。

## (3) 特点：

随机性 { 随机发生的事态。 (键盘)  
有意调用、随机请求处理的事态。 (打印机)  
随机插入的事态。 (软中断)

## (4) 应用

控制中、低速 I/O 操作  
处理复杂随机事态

## 2. 中断服务程序入口地址的获取

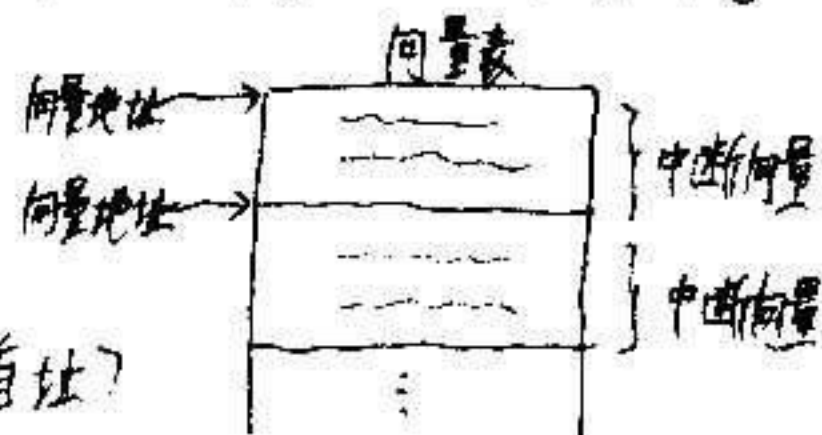
### (1) 向量中断方式

将服务程序入口地址(中断向量)组织在中断向量表中；响应中断时，由硬件直接产生对应于中断源的向量地址，访问向量表，取得相应服务程序入口，转入服务程序

中断向量：服务程序入口地址、服务程序状态字

中断向量表：存放中断向量的存储区

向量地址：访问向量表的地址(指向中断向量的首址)

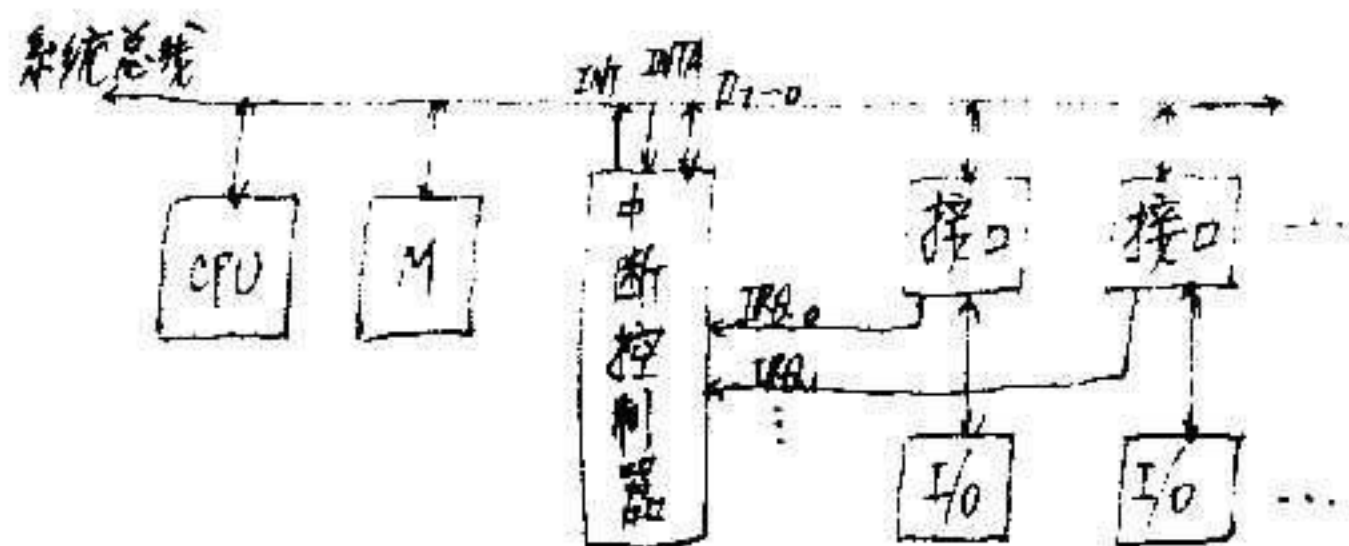


### (2) 非向量中断方式

将服务程序入口组织在查询程序中；CPU响应时执行查询程序，确定中断源，转入相应服务程序。

## 3. 中断接口功能模型 [P354]





- (1) 寄存器选择: 对接口寄存器寻址
- (2) 命令字寄存器: 接收CPU发向外设的命令
- (3) 状态字寄存器: 反映设备和接口的运行状态
- (4) 数据缓冲器: 传递数据, 实现缓冲
- (5) 控制逻辑
  - 请求信号产生逻辑
  - 电平转换逻辑
  - 串-并转换逻辑 (串口)
  - 针对设备特性的逻辑

(6) 公用中断控制器: 接收外设请求, 屏蔽、判优, 送出公共请求; 接收中断批准, 送出中断号(中断类型码)或向量地址。

4. 中断全过程 (主机与外设交换信息)

- (1) 初始化: 设置接口和中断控制器工作方式, 送屏蔽字, 送中断号。
- (2) 发启动命令 (送命令字), 启动设备
- (3) 设备完成工作, 申请中断
- (4) 中断控制器汇集各请求, 经屏蔽、判优, 形成中断号, 并向CPU送公共请求INT
- (5) CPU响应, 发批准INTA, 并关中断, 保存断点。



- (16) 中断控制器送出中断号
- (17) CPU执行中断隐指令操作，将中断号转换为向量地址，查向量表，取入口，进入服务程序
- (18) CPU执行服务程序，进行中断处理（交换数据）  
 { 单级中断  
 { 多重中断
- 注意屏蔽技术的两个应用 { 标志改变优先级  
 { 实现多重中断
- (19) 返回原程序（返回前开中断）

## 5. 中断接口设计

设计关键：通用机如何针对设备的多样性、特殊性发出具体命令。

解决：(1) 命令字、状态字格式的拟定

(2) 中断源的扩展

例：某机需扩展两个外中断源。CPU发向两设备的命令字包括启动、停止、读、写等。设备状态包括忙、完成、故障等。两设备共用一个中断类型码（IR<sub>R2</sub>）。设计接口，要求两设备能同时启动，并行工作。

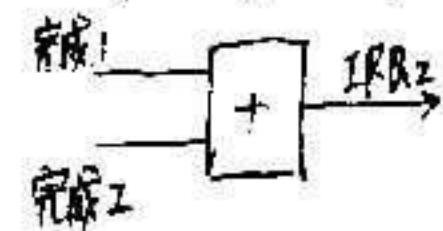
(1) 接口组成

两设备共用一个接口，设置一个命令字（分两段）一个状态字（分两段），两个缓冲器。

(2) 命令字、状态字格式

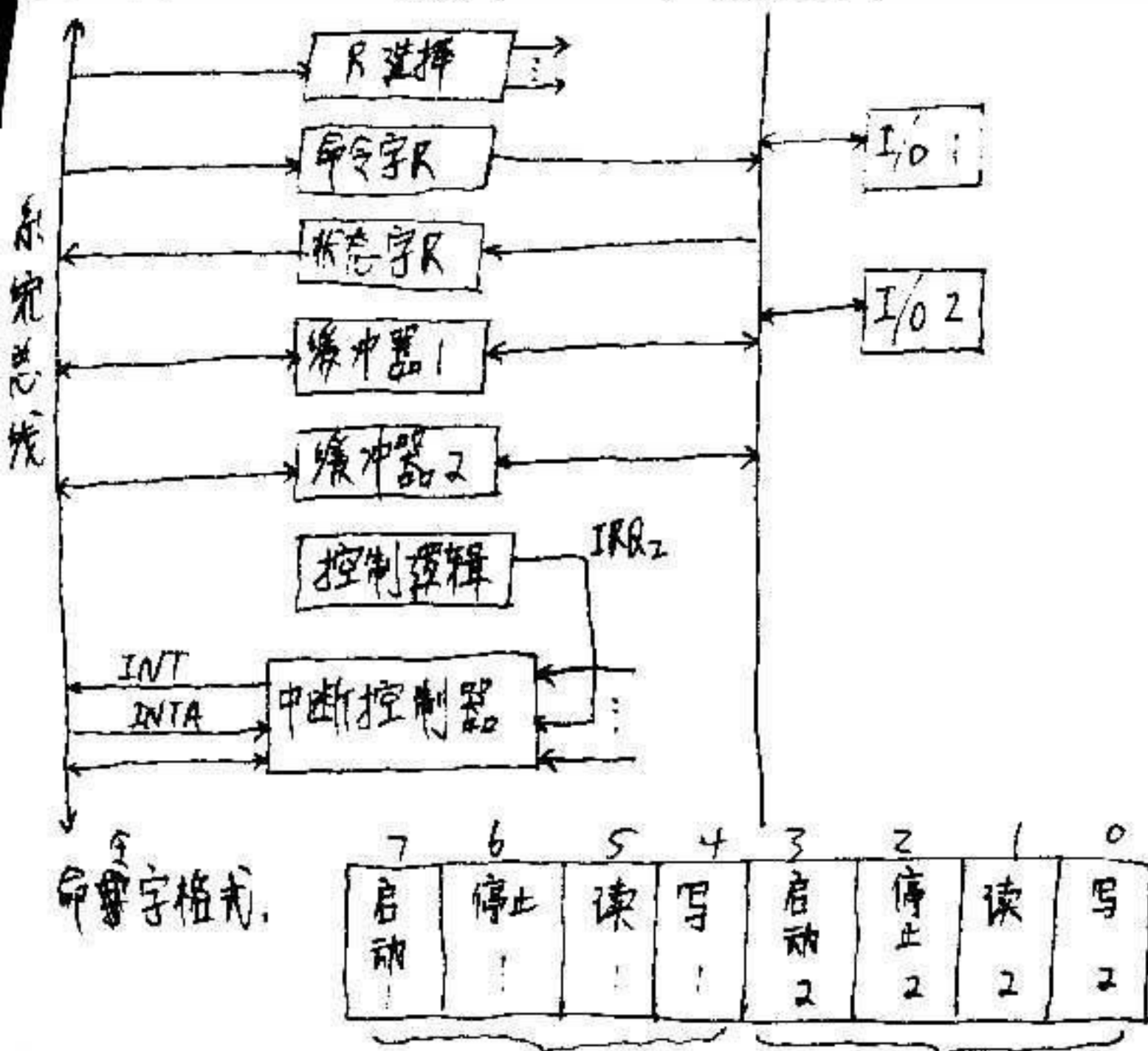
(3) 扩展方法：向量中断 + 非向量中断方式

命令字为：1000 1000



两设备公共请求送IR<sub>R2</sub>，若IR<sub>R2</sub>被响应，则转入IR<sub>R2</sub>服务程序，在该程序中设置有两设备服务程序入口。IR<sub>R2</sub>服务程序查询状态字（先查询的设备优先级高），转相应中断处理

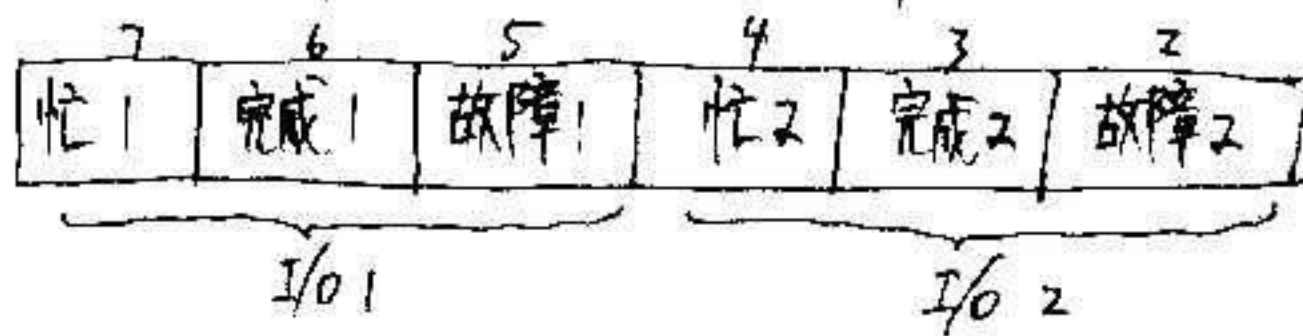




控制I/O 1

控制I/O 2

状态字格式:



## 3.2.2 DMA方式

1. 定义及应用

(1) 定义

直接依靠硬件实现主存与I/O间的数据传送, 传送期间不需CPU程序干预。

注意:

1) I/O与主存, 而不是I/O与CPU或I/O与主机交换数据。

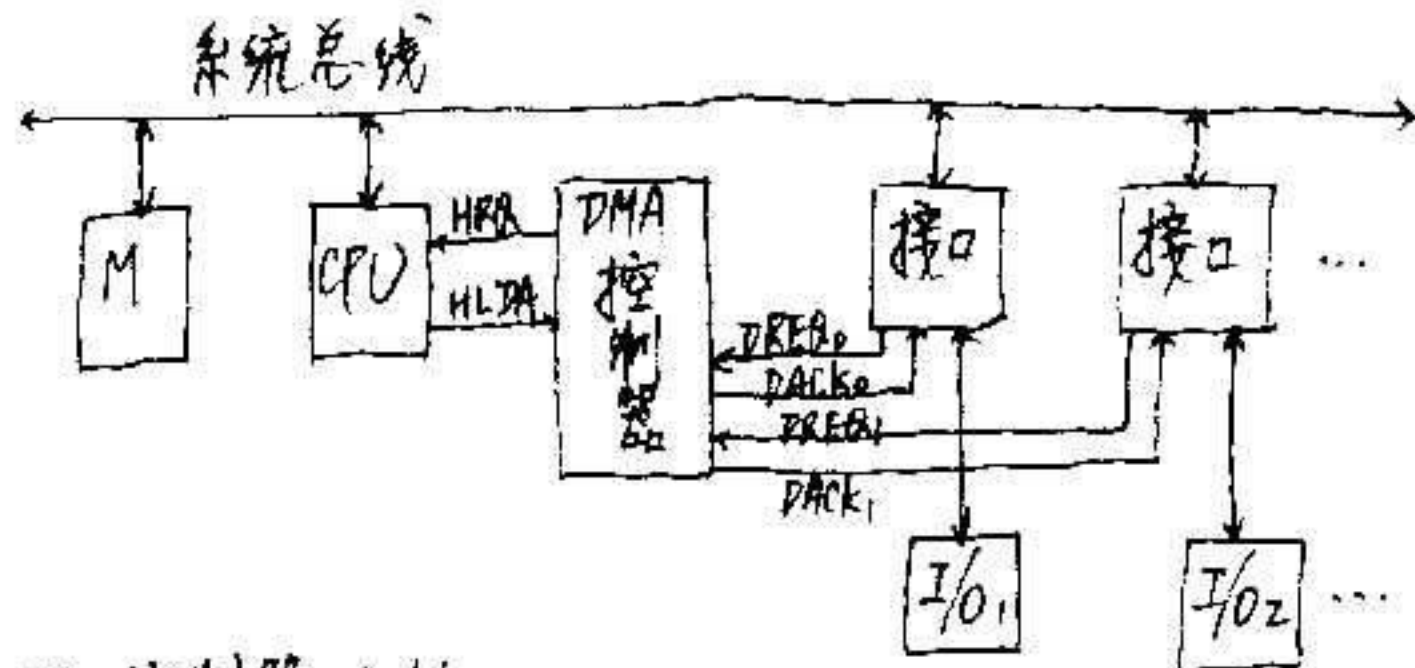
2) 直接依靠硬件传送, 而不是执行程序传送

3) 传送前的初始化和传送结束处理, 需CPU执行程序实现

(2) 应用

用于高速、简单、批量数据传送

## 2. DMA控制器与接口



(1) DMA控制器功能

1) 接收初始化信息 (传送方向、主存首址、交换量)

2) 接收接口的DMA请求 (DREQ), 向CPU申请总线 (HRA)

3) 接收CPU的总线响应 (HLDA), 向接口发回DMA应答 (DACK)

4) 接管总线权, 控制DMA传送。

(2) 接口功能

1) 接收初始化信息 (外设寻址信息)

2) 接收CPU发向设备的命令, 反映设备、接口的状态。

3) 传送数据。

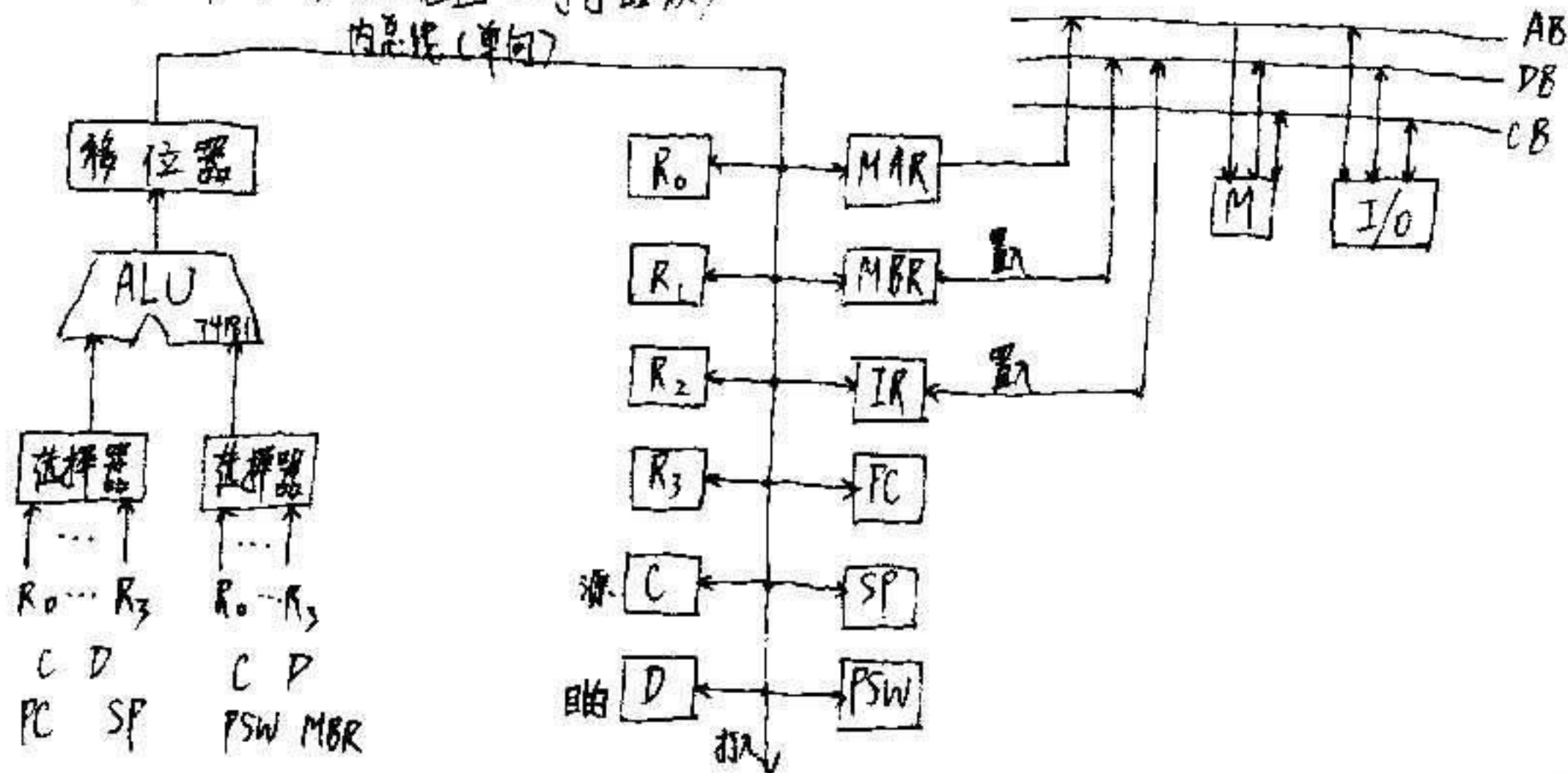


## 第一章

## F) 组织

## 1.1 逻辑组成 (模型机)

## 1. CPU 数据通路框图 (寄存器级)



## 2. 结构特点

## 1) 寄存器 独立结构

可编程: 通用寄存器  $R_0 \sim R_3$ ,  $PC$ ,  $SP$ ,  $PSW$

非编程:  $C$ ,  $D$ ,  $IR$ ,  $MAR$ ,  $MBR$

## 2) ALU

作为CPU内部数据传送通路的中心

输入选择器: 选择操作数来源

ALU: 运算处理

输出移位器: 选择输出方式

## 3) 内总线

### 3. DMA全过程

#### (1) 初始化

CPU通过程序向DMA控制器和接口送出初始化信息，启动设备

#### (2) DMA传送

DMA控制器获得总线权，控制直传，并自动修改地址、交换量。

#### (3) 结束处理

批量传送完毕，接口申请中断，CPU执行中断处理程序，作结束处理。

注意：

三个阶段各采用什么方式？各完成哪些操作？

( 程序传送方式， 硬件传送方式， 中断方式 )

## 第四章 主要I/O设备原理

### 4.1 磁盘

#### 4.1.1 信息分布与寻址信息

##### 1. 信息分布

盘片、记录面、磁道(圆柱面)、扇区

##### 2. 寻址信息

驱动器号、圆柱面号(磁道号)、磁头号、扇区号(起始扇区)、交换量(扇区数)

#### 4.1.2 磁记录方式

NRZI (早期低速磁带机)、 PE (常规磁带机)

FM、MFM (磁盘)、 GCR (流式磁带机)

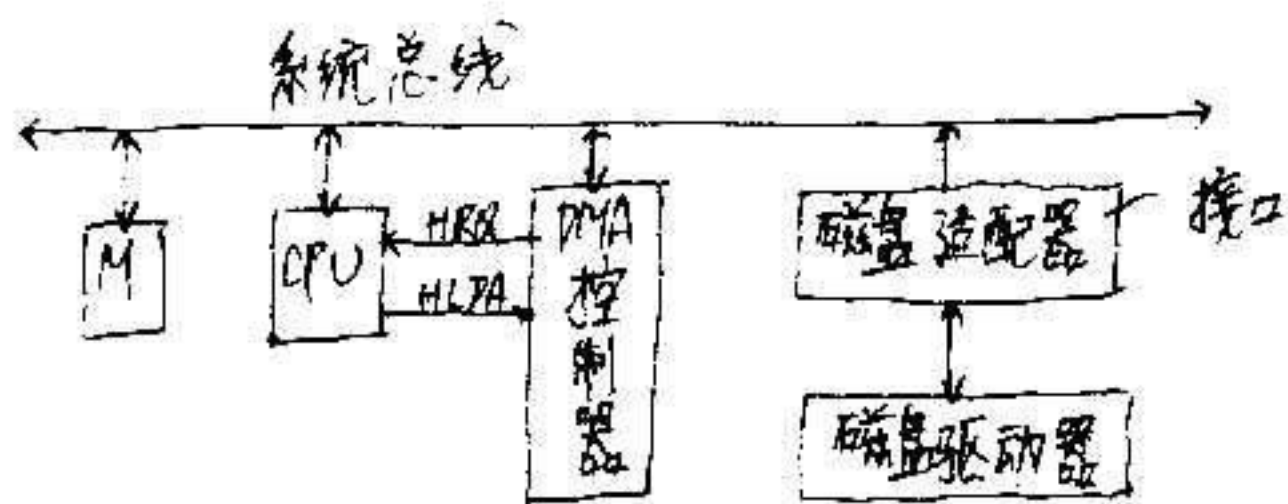
NRZI: 归零-1制 PE: 调相制 FM: 调频制 MFM (M<sup>2</sup>F): 改进型调频制

GCR: 群码制



#### 4.1.3 调用过程 (DMA方式)

- (1) CPU向适配器送出驱动器号、圆柱面号、磁头号、起始扇区号、扇区数等外设寻址信息；向DMA控制器送出传送方向、主存首址、交换量等信息。
- (2) 适配器启动寻道，并用中断方式判寻道是否正确。



- (3) 适配器准备好 (读盘: 扇区缓存满一扇区; 写盘: 扇区缓存空一扇区), 提出DMA请求。
- (4) CPU响应, 由DMA控制器控制总线, 实现传送。
- (5) 批量传送完毕, 适配器申请中断。
- (6) CPU响应, 调回状态字, 作善后处理。

#### 4.1.4 主要技术指标

##### 1. 容量

- 非格式化容量: 磁盘总容量 (由位密度计算)
- 格式化容量: 磁盘格式化后的有效容量 (由扇区数据长度计算)

##### 2. 速度

- 平均寻道 (平均定位) 时间: ms
- 平均旋转 (平均等待) 时间: ms
- 数据传输率: b/s

## 4.2 CRT 显示器

### 4.2.1 显示方式与分辨率

1. 字符、数字方式 (A/N 方式):  $m$  行  $\times$   $n$  列 (25 行  $\times$  80 列)
2. 图形方式 (APA 方式):  $m$  点  $\times$   $n$  线 (1024 点  $\times$  768 线)

### 4.2.2 显示缓存 VRAM 与屏幕显示的对应关系

#### 1. 显存内容与容量计算

##### (1) 内容

字符方式: 字符编码

图形方式: 图形象点代码

##### (2) 容量

1) 字符方式: 一个字节单元存放一个字符编码

例: 分辨率为 25 行  $\times$  80 列 基本容量 =  $25 \times 80 = 2kB$

2) 图形方式: 一位存放一个点

例: 分辨率为 640 点  $\times$  200 线 基本容量 =  $\frac{640 \times 200}{8} = 16kB$

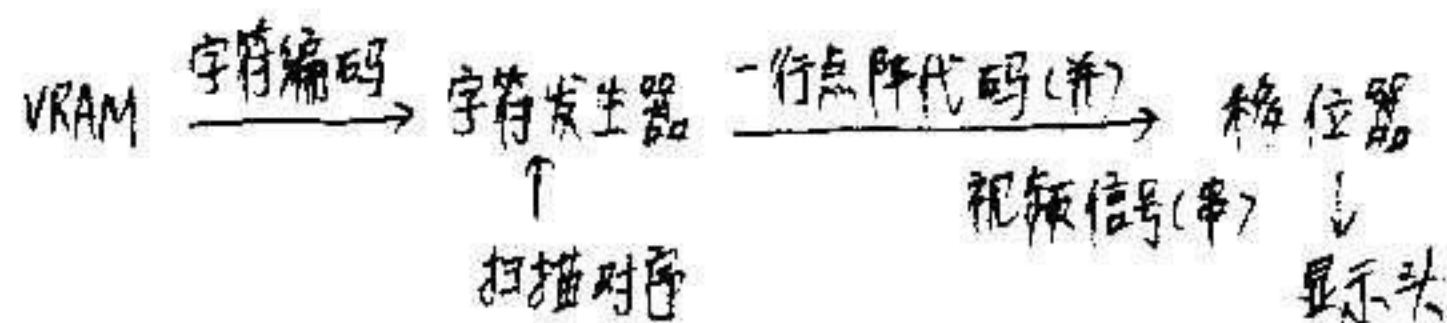
#### 2. 显存地址组织

显存单元的地址由屏幕显示的行、列号决定。

屏幕显示从左向右、自上而下, 显存地址从低到高安排。

#### 3. 信息转换

##### 1) 字符方式





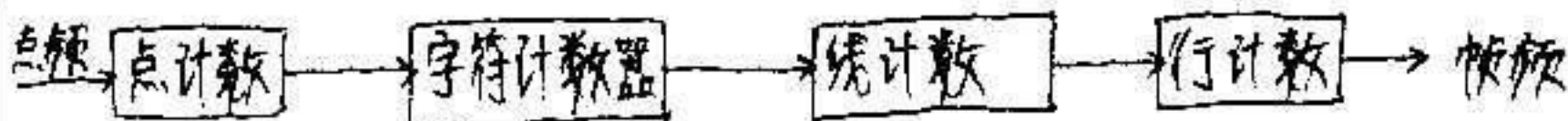
ROM: { 内容: 字符点阵代码  
容量: 字符种类数  $\times$  字符外包点数

## 2) 图形方式

VRAM  $\xrightarrow{\text{一字节点代码(并)}}$  移位器  $\xrightarrow{\text{视频信号(串)}}$  显示头

## 4. 同步控制

### (1) 字符方式



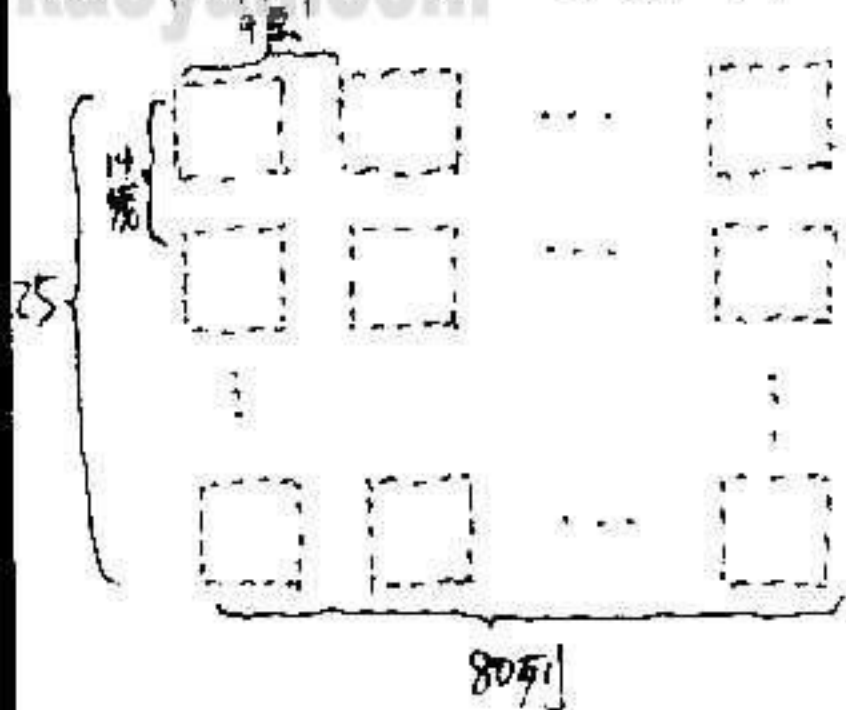
决定访问  
显存时间

发水平同  
步信号

提供  
ROM行选

发垂直  
同步信号

例: 分辨率  $25 \times 80$ , 字符点阵  $7 \times 9$ , 字符区  $9 \times 14$



80列

1) 点计数器: 对一个字符的一行点计数

$9 = 1$  帧频, 一次点计数值环访问一次 VRAM, ROM

2) 字符计数器: 对一帧的字符列计数

$(80 \text{ 列}) = 1$  帧频, 一次字符计数值环发一次水平同步信号

3) 线计数器: 对一行字符的扫描步计数.

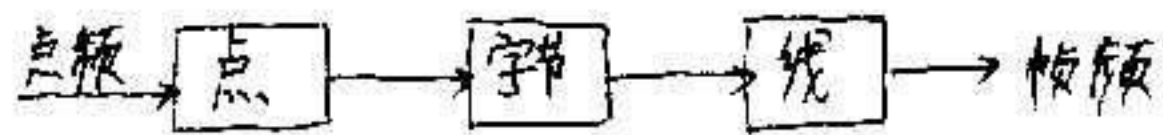
计: 帧频 线计数值提供 ROM 行地址.

4) 行计数器: 对一帧的字符行计数

$(25+n) = 1$  帧频, 一次行计数循环发一次垂直同步信号.

## (2) 图形方式

例: 分辨率  $640 \times 200$



1) 点计数器:  $8:1$  帧频, 访问 VRAM

2) 字节计数器:  $(80+m) = 1$  帧频, 发水平信号

3) 线计数器:  $(200+n) = 1$  帧频, 发垂直信号

## 4.3 键盘

软件扫描法: 逐行扫描

## 4.4 打印机

调用过程 (中断方式):

初始化, 接收代码, 打印处理

显示器 ROM: 按行组织点阵代码

打印机 ROM: 按列组织点阵代码



单向数据总线：ALU总线，实现数据分配

从内存接收数据用同步

打入方式，从系统总线接收数据采用置入方式

#### 4) 与系统总线的连接

由MAR、MBR实现连接

### 1.2 工作机制

#### 1.2.1 指令流程（寄存器传送级）

用寄存器传送语言描述指令从读取到执行的整个流程

拟实流程的关键：  
清楚了解数据通路结构  
熟练掌握基本寻址方式

#### 1. 基本寻址方式（模型机）

寄存器寻址： $R$

寄存器间址： $(R)$

自减型寄存器间址： $-(R)$ ， $-(SP)$ （用于入栈操作）

自增型寄存器间址： $(R)+$ ， $(SP)+$ （用于出栈操作）

立即寻址： $(PC)+$  变址： $X(R)$

相对寻址： $X(PC)$

#### 2. 思路

1) 了解指令功能，具体完成什么操作

MOV: 源数  $\rightarrow$  目的地

ADD: 结果  $\rightarrow$  目的地

JMP: 转移地址  $\rightarrow PC$

RST: 返回地址  $\rightarrow PC$

JSR: 子程序入口  $\rightarrow PC$ ，并保存返回地址

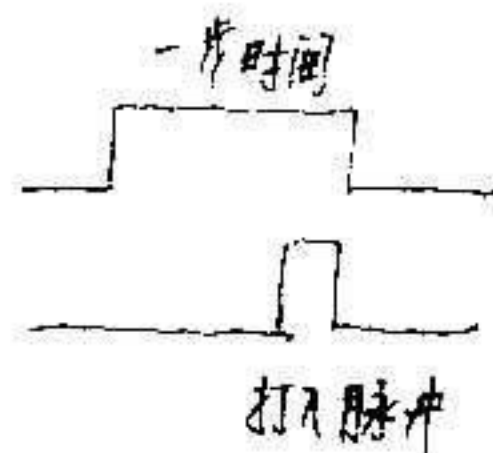
2) 分清源和目的，确定所采用的寻址方式

源在前，目的在后。

3) 按周期拟定分步流程

模型机允许：每一步完成

{ 一次从M读出, 并经数据通路传送的操作;  
 或一次经数据通路传送的操作;  
 或一次向M写入的操作



例题:

1) MOV (SP)+, X(R<sub>1</sub>)

2) MOV X(R<sub>2</sub>), -CSP

3) ADD -(R<sub>0</sub>), X(R<sub>3</sub>)

4) NEG R<sub>2</sub>

5) JMP (R<sub>0</sub>)

6) JSR (R<sub>1</sub>)+

7) JMP X(PC)

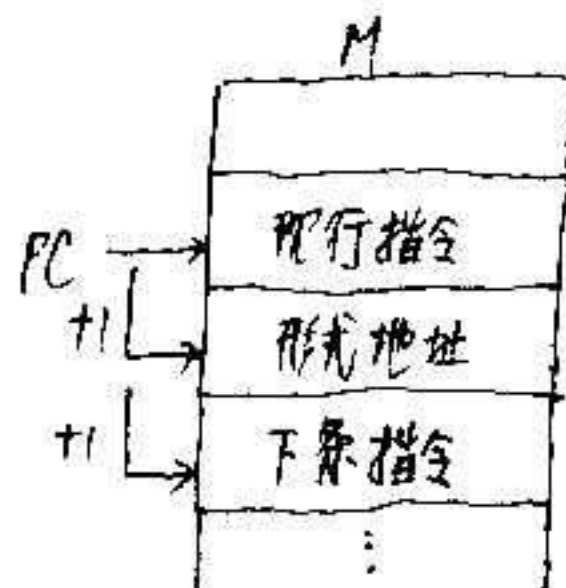
8) AND (PC)+, X(PC)

① MOV (SP)+, X(R<sub>1</sub>)

FT: M → IR,  
PC+1 → PC

ST: SP → MAR  
M → MBR → C  
SP+1 → SP

DT: PC → MAR  
M → MBR → D 形地





$$R_1 + D \rightarrow MAR \quad \text{目的}$$

$$PC + 1 \rightarrow PC$$

ET:  $C \rightarrow MBR$

$$MBR \rightarrow M$$

$$PC \rightarrow MAR$$

② MOV  $X(R_2), -(SP)$

FT:  $M \rightarrow IR, PC + 1 \rightarrow PC$

ST:  $PC \rightarrow MAR$

$$M \rightarrow MBR \rightarrow C \quad \text{形地}$$

$$R_2 + C \rightarrow MAR$$

$$M \rightarrow MBR \rightarrow C \quad \text{源操作数}$$

$$PC + 1 \rightarrow PC$$

DT:  $SP - 1 \rightarrow SP, MAR$

ET:  $C \rightarrow MBR$

$$MBR \rightarrow M$$

$$PC \rightarrow MAR$$

③ ADD  $-(R_0), X(R_3)$

FT:  $M \rightarrow IR, PC + 1 \rightarrow PC$

ST:  $R_0 - 1 \rightarrow R_0, MAR$

$$M \rightarrow MBR \rightarrow C \quad \text{源数}$$

DT:  $PC \rightarrow MAR$

$$M \rightarrow MBR \rightarrow D \quad \text{形地}$$

$$R_3 + D \rightarrow MAR$$

$$M \rightarrow MBR \rightarrow D \quad \text{目的操作数}$$

$$PC + 1 \rightarrow PC$$

ET.  $C + D \rightarrow MBR$

$$MBR \rightarrow M$$

$$PC \rightarrow MAR$$

④ NEG  $R_z$

FT.  $M \rightarrow IR, PC + 1 \rightarrow PC$

ET.  $\overline{R_z} + 1 \rightarrow R_z$

$$PC \rightarrow MAR$$

⑤ JMP  $(R_0)$

FT.  $M \rightarrow IR, PC + 1 \rightarrow PC$

ET.  $R_0 \rightarrow MAR$

$$M \rightarrow MBR \rightarrow PC, MAR$$

⑥ JSR  $(R_1) +$

FT.  $M \rightarrow IR, PC + 1 \rightarrow PC$

ST.  $R_1 \rightarrow MAR$

$$M \rightarrow MBR \rightarrow C \quad \text{子程序入口}$$

$$R_1 + 1 \rightarrow R_1$$

ET.  $SP - 1 \rightarrow SP, MAR$

$$PC \rightarrow MBR$$

$$MBR \rightarrow M$$

$$C \rightarrow PC, MAR$$

} 保存返回地址



⑦ JMP X(PC)

FT:  $M \rightarrow IR, PC+1 \rightarrow PC$

ET:  $PC \rightarrow MAR$

$M \rightarrow MBR \rightarrow C$  — 位移量

$PC + C \rightarrow PC, MAR$



⑧ AND (PC)+, X(PC)

FT:  $M \rightarrow IR, PC+1 \rightarrow PC$

ST:  $PC \rightarrow MAR$

$M \rightarrow MBR \rightarrow C$  — 立即数

$PC+1 \rightarrow PC$  — 位移量地址

DT:  $PC \rightarrow MAR$

$M \rightarrow MBR \rightarrow D$  — 位移量

$D + PC \rightarrow MAR$

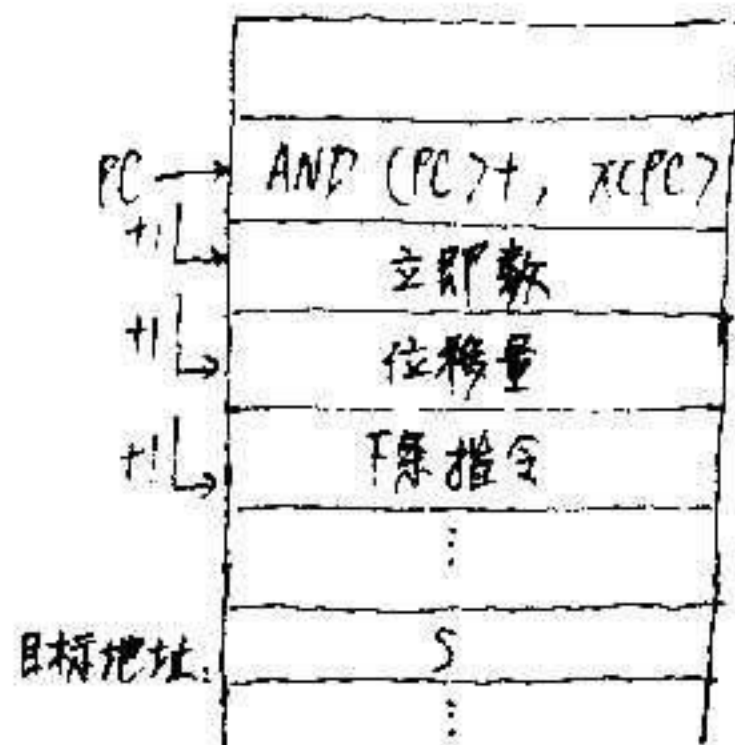
$M \rightarrow MBR \rightarrow D$  — 目的数

$PC+1 \rightarrow PC$  — 下一条指令地址

ET:  $C \cdot D \rightarrow MBR$

$MBR \rightarrow M$

$PC \rightarrow MAR$



## 1-2.2 微命令序列

指令流程在微操作级的具体实现

### 1. 微命令设置

## (1) 数据通路指令

ALU 输入选择:

$R_0 \rightarrow A, C \rightarrow B, \dots$

ALU 功能选择:

加, 减, 加1, 传送,  $\dots$

移位功能选择:

左移, 右移, 直送,  $\dots$

结果分配:

$CPR_0, CPR_1, CPC, \dots$

## (2) 访存操作

地址使能 EMAR, 读R, 写W (读/写 R/W)

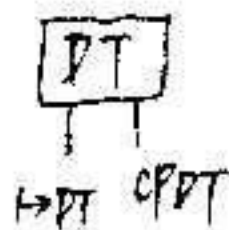
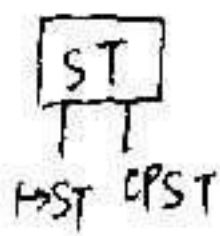
置入 SMAR, 置入 SIR

## (3) 时序转换 (组合逻辑控制方式)

周期状态设置, 清除:

节拍 (时钟周期) 计数, 清除:

周期状态触发器:



节拍计数器:

