主要内容.

∴ CPU

① 逻辑组成: 寄存器、ALU 设置和数据通路结构

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*
  微程序控制级: 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 / 异步控制 } CPU 内部是同步控制 }

2. 常用运算方法的规则、进位链. (澄清一些模糊问题)

原码. 补码乘除法及浮点运算.

3. 存储器

① 基本概念.

② 半导体存储器芯片逻辑设计 *

(RAM 地址译码. 存储逻辑. 连接)

4. 总线

① 基本概念

② 总线总线、信号组成. 时序控制方式 { 同步方式及打限 / 异步方式 } 什么是同步. 异步. 用时序波形描绘

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. 针对某种应用描述

DMA: 基本概念. DMA 控制器与接口. DMA 过程.

多应用场合　　最早生产微命令的基本方法

用于高速计算机或小规模机器中.

2. 微程序控制方式 (MS)

1) 基本思想　a. 用存储逻辑实现CPU

b. 将程序技术引入CPU的物成级, 利用程序技术去编排指令的解释与执行

1) 将微命令以代码形式编成微指令, 控制一步操作.

2) 若干微指令组成一段微程序, 解释执行一条机器指令.

3) 微程序事先存放在控制存储器中, 执行机器指令时再取出.

机器指令	微指令	工作程序	微程序	主存(M)	控存(CM)
一般指令	一次操作	若干机器指令	若干微指令	存放工作程序	存放微程序
实现指令系统的一种基本操作	完成机器指令的一次操作	完成某一特定任务	解释执行一条机器指令	以外	以外

4) 优缺点:

结构规整　设计效率高　易于修改指令系统功能　速度较慢

执行效率较高

3) 应用场合

速度要求不高, 功能更复杂的机器中　特别适用于系列机.

组合型微指令和微程序控制方式比较：

① 产生微命令方法 < 组合型微命令提供（组）
存储型微命令提供（微）

② 时序方式 < 三级：工作周期、时钟周期、脉冲（组）
二级：微指令周期（时钟）、脉冲（微）

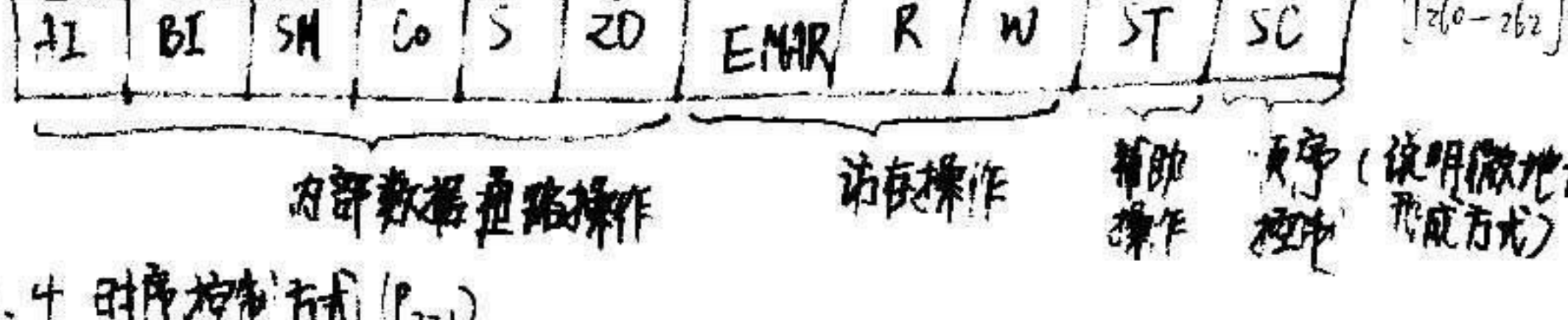
③ 规整性 < 不规整（组）
规整（微）

④ 可扩展性 < 不易扩展（组）
易于扩展（微）

⑤ 速度 < 快（组）
慢（微）

3. 微指令

格式：微命令字段区别，按操作类型分段，同类互斥微命令放同一字段。



1.2.4 时序控制方式 (P221)

1. 同步控制方式

① 定义：各项操作与统一时序信号同步，其主要特征是有严格的时钟周期规定。

② 特点：有明显时序信号标记。

时钟周期时间固定。

各操作由统一，各部件之间的数据传送受严格规定时控制。

b 应用.

CPU内部. 设备内部. 系统总线操作.

2. 异步控制方式.

① 定义. 各项操作按需要或预先安排不同时间, 不受统一时序约束.

② 特点: 无严格时钟周期节拍.

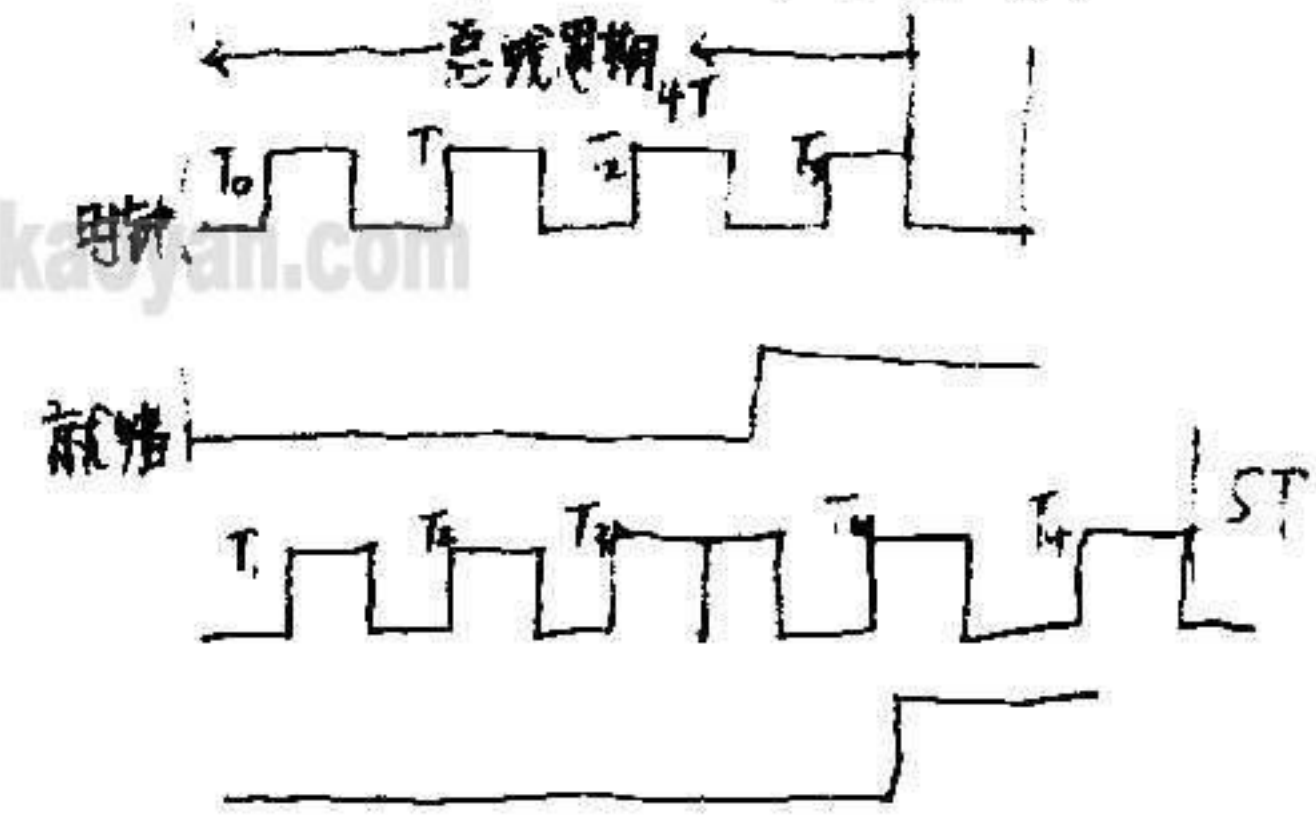
各操作间的衔接各部件间的数据传送采用异步应答方式.

③ 应用: 异步总线操作.

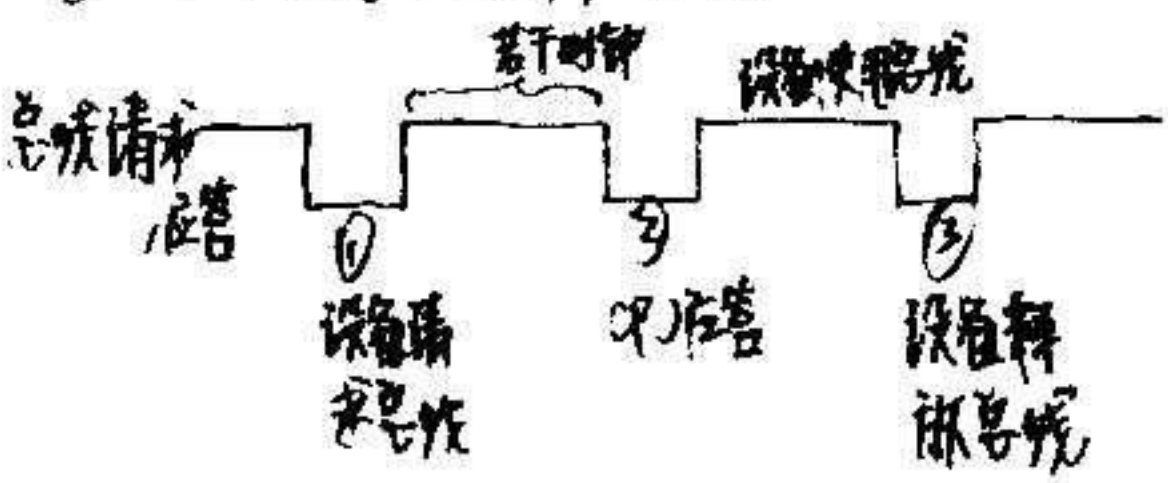
3. 同步方式在实际应用中的改进 —— 扩展同步.

① 不同指令安排不同时钟周期数. 时钟周期宽度不变.

② 总线周期中允许插入若干个时钟周期.



③ 同步方式引入异步应答.



第3章 运算方法与运算器

3.1 数的表示和运算方法

1. 原码乘法 P53

原码对绝对值运算，符号单独处理。一位乘两个数的积

用定点小数说明

用移位和加法代替乘法

A存乘码两位乘后中，设置乘积符号位Cf，初值为0。当需要+X时，先作-X操作，得结果右移两位，并置Cf为1。下次再作+X操作

2. 补码乘法 一位乘（比较法）

用双符号位。判断规则

y_n	y_{n+1}	操作
0	0	原部分积右移一位
0	1	原部分积+X补后，再右移一位
1	0	原部分积-X补后，再右移一位
1	1	原部分积右移一位

补码运算：用补码表示操作数，连同符号位一起直算

3. 原码除法

用原码表示对操作数，只取尾数（绝对值）相除

符号位单独处理：同号除，商为正；异号除，商为负

余数 R_i 为正，商 1，下步操作 $2R_i - [y]_{补}$

余数 R_i 为负，商 0，下步操作 $2R_i + [y]_{补}$

4. 补码除法

余数 R_i 与除数 y 同号，商 1，下步操作 $2R_i - [y]_{补}$

余数 R_i 与除数 y 异号，商 0，下步操作 $2R_i + [y]_{补}$

假商修正：假商符加1（变反）末位商置1

5. 浮点运算 P53 P74-77

加减步骤

阶码、结果规格化，小阶向大阶对齐

3.2 运算器

逻辑运算 P44-47

C_i 串、并行累加器同时进位表达式

串：$C_0 = A_0 B_0 + (A_0 \oplus B_0) C_{-1}$

并：$C_0 = A_0 B_0 + (A_0 \oplus B_0) C_{-1}$

用级进：

$$C_0 = G_0 + P_0 G_5 + P_0 P_5 C_I$$

$$C_I = G_{4I} + P_{4I} C_0$$

$$\left.\begin{aligned} G_{4I} &= G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 \\ P_4 &= P_4 P_3 P_2 P_1 \end{aligned}\right\}$$

$$G_i = A_i B_i \qquad P_i = A_i \oplus B_i$$

§1.4 其他基本概念、

溢出标志 方法 地址结构 隐地址 显地址

堆栈的设置及功能扩展

主机对外设的寻址方式 { 单独寻址 / 统一寻址

第二章 存储子系统

§2.1 半导体存储器 逻辑设计 P105

例1. 用SRAM芯片（1K×4位）组成 4KB存储器，地址线 A_{15}～A_0（低），双向数据线 D_3～D_0（低），读写线 $R/\overline{W}$

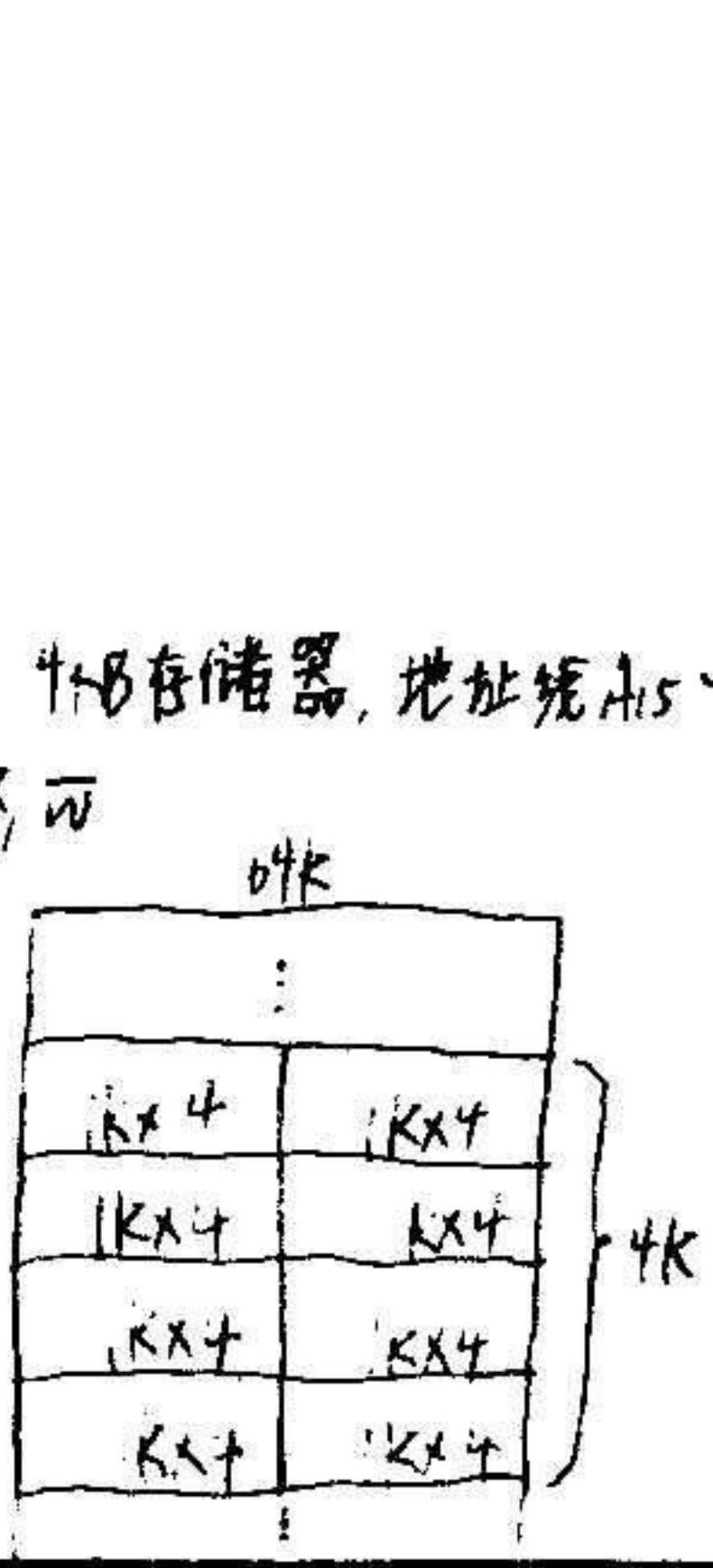
① 芯片数：8片

② 存储空间安排

任意连续区间

③ 地址分配和片选逻辑

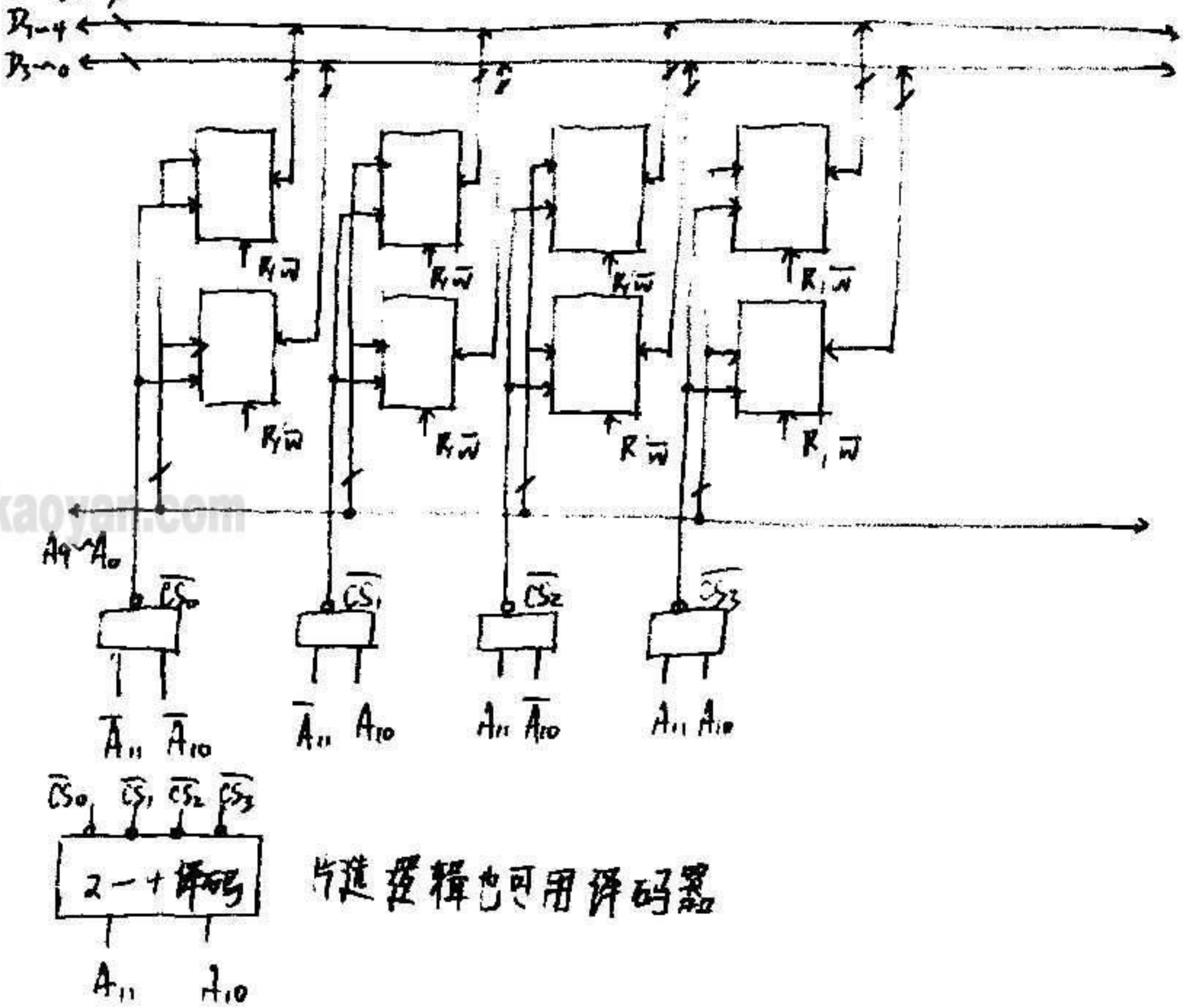
4KB：12位地址，A_{11}～A_0



芯片　地址范围　片选逻辑

1K	$A_9 \sim A_0$	$\overline{CS_0} = \overline{A_{11}}\,\overline{A_{10}}$
1K	$A_9 \sim A_0$	$\overline{CS_1} = \overline{A_{11}}\,A_{10}$
1K	$A_9 \sim A_0$	$\overline{CS_2} = A_{11}\,\overline{A_{10}}$
1K	$A_9 \sim A_0$	$\overline{CS_3} = A_{11}\,A_{10}$

④ 选择



片选逻辑也可用译码器

例2. 用4K×8的ROM芯片、2K×4和1K×4的RAM芯片组成7KB存储器

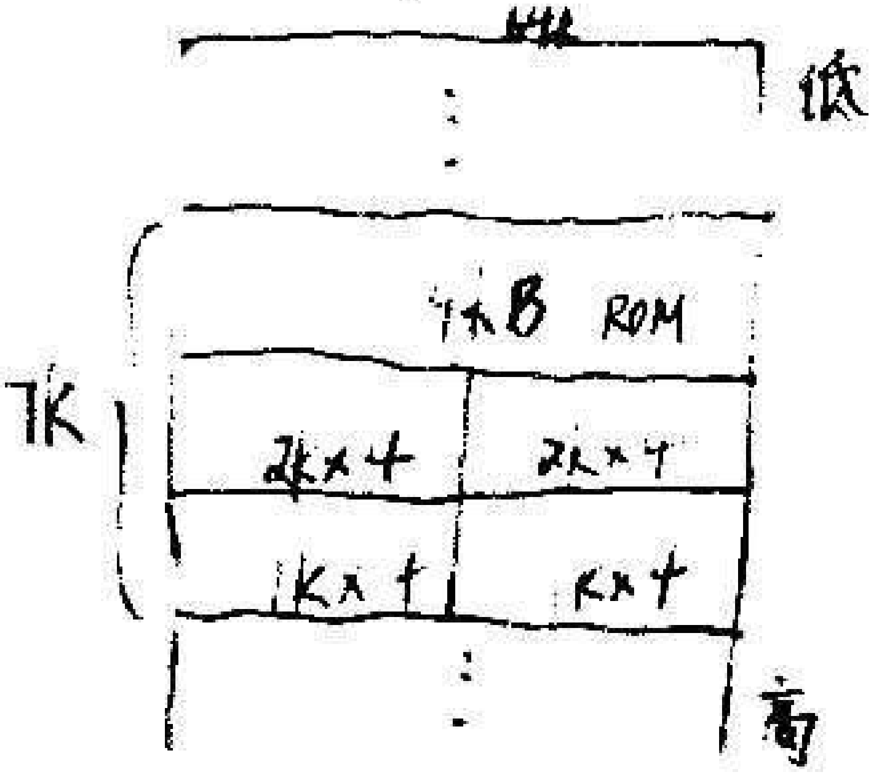
地址总线 $A_{15} \sim A_0$（低），双向数据线 $D_7 \sim D_0$（低），读写线 $R/\overline{W}$

① 芯片数： 5片

② 存储空间安排、任意连续区间

先安排大容量芯片，后安排小容量芯片

一般从大到小，这样片选逻辑较简单

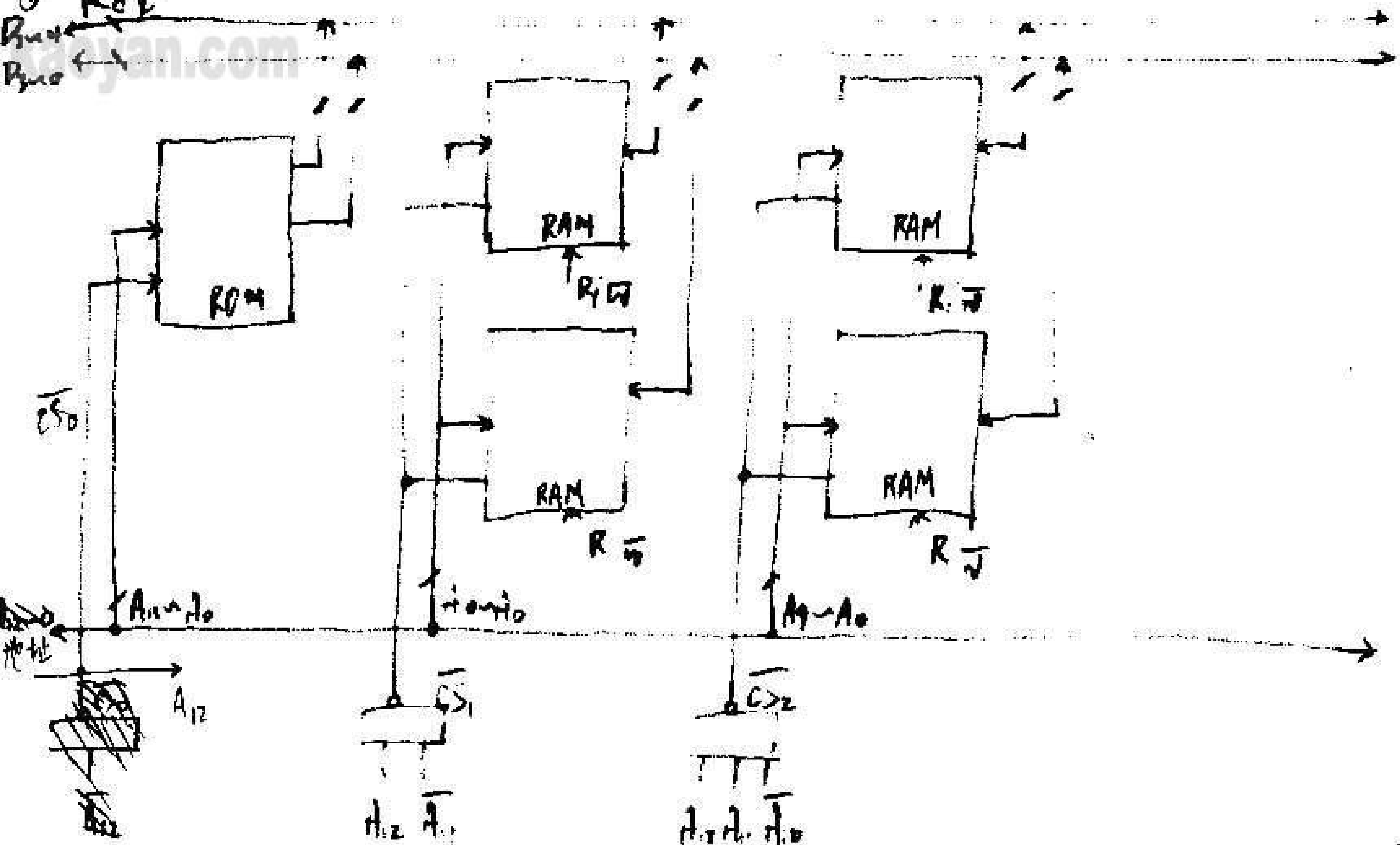


③ 地址分配和片选逻辑

7K: $A_{12} \sim A_0$

芯片	地址分配	片选逻辑
4K	$A_{11} \sim A_0$	$CS_0 = \overline{A_{12}}$
2K	$A_{10} \sim A_0$	$CS_1 = A_{12}\overline{A_{11}}$
1K	$A_9 \sim A_0$	$CS_2 = A_{12}A_{11}\overline{A_{10}}$

④ 连接



例5. 某机器地址空间 [illegible] ROM区: [illegible] 0111 1111 1111

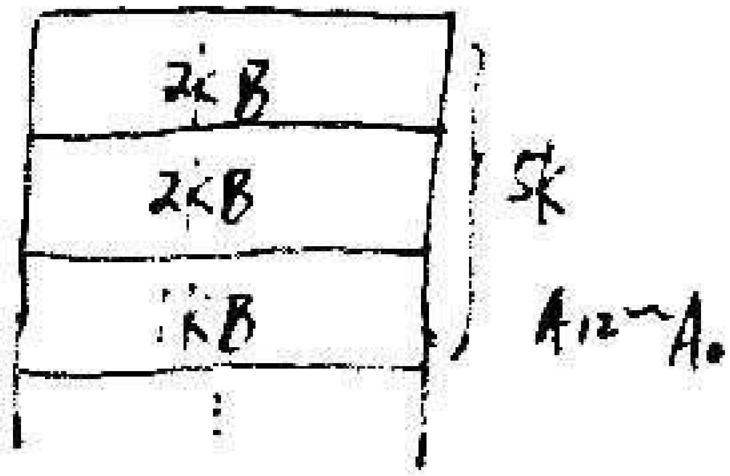
RAM区: 0800H ~ 13FFH

所用芯片: EPROM 2K片 RAM 2K片, 1K片

① 计算容量

ROM区: $0 \sim 2^{11}-1$ 即 2KB

RAM区: 13FF − 0800 = BFF 3KB 1011 1111 1111 +1



② 地址分配 片选逻辑 1100 0000 0000 [illegible] 3K

芯片	地址	片选逻辑
2K	$A_{10} \sim A_0$	$CS_0 = \overline{A_{12}}\,\overline{A_{11}}$
2K	$A_{10} \sim A_0$	$CS_1 = \overline{A_{12}}\,A_{11}$
1K	$A_9 \sim A_0$	$CS_2 = A_{12}\,\overline{A_{11}}\,\overline{A_{10}}$

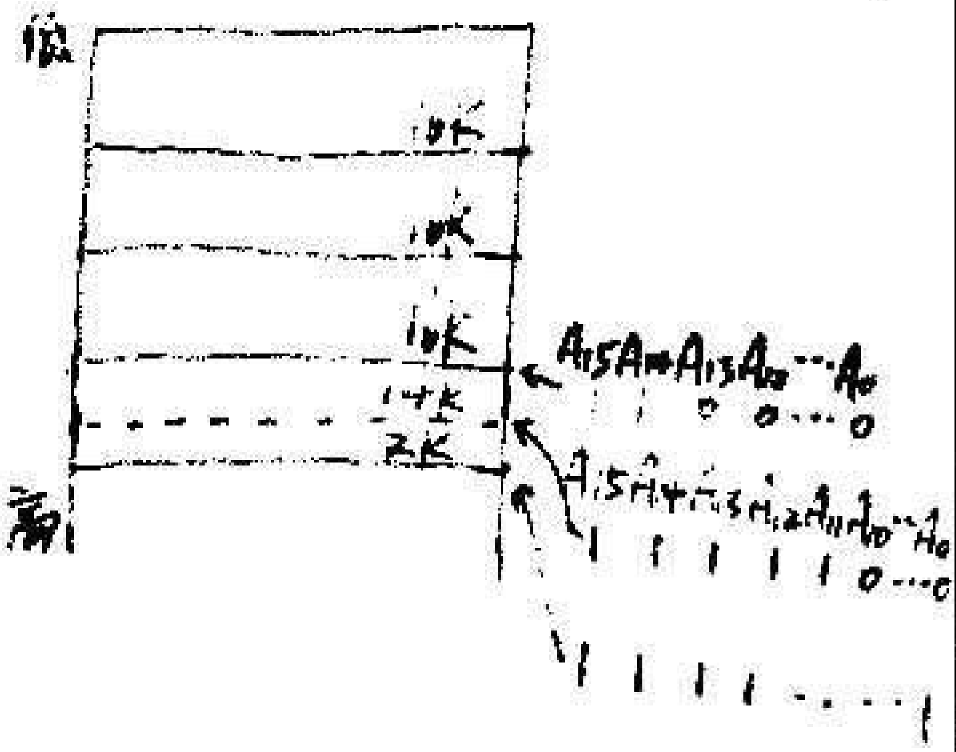
(高位地址线) $A_{15} \sim A_{13}$ 为全0

例1. [illegible]

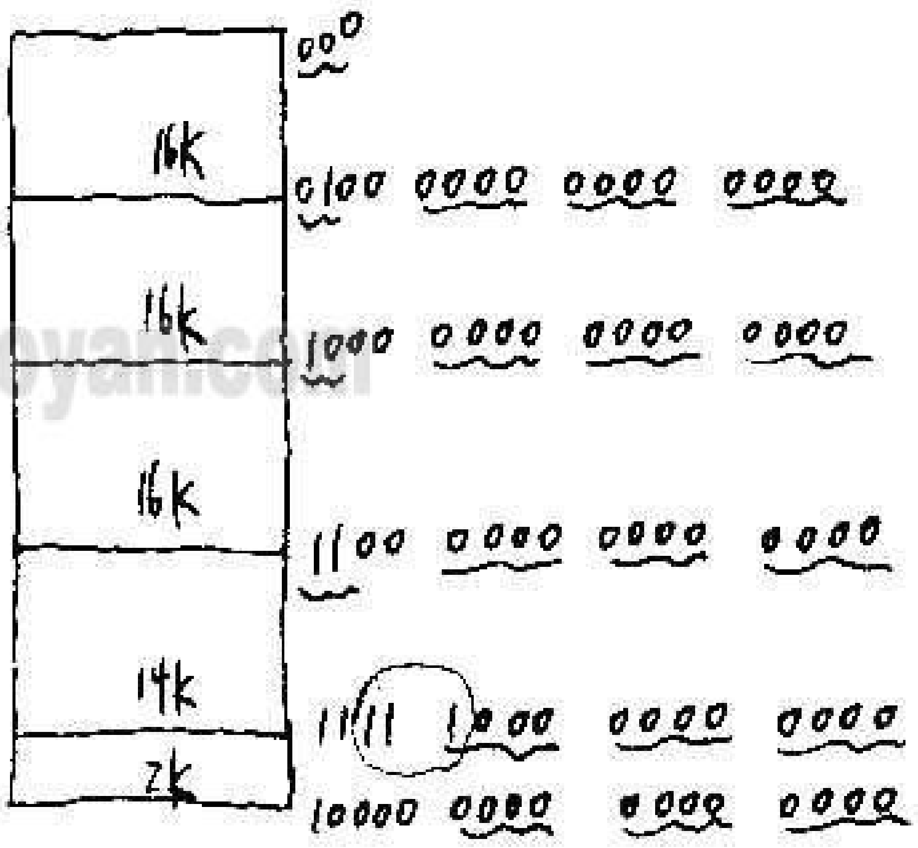
① 芯片数：4片

② 存储空间安排

③ 地址分配与片选逻辑



芯片	地址分配	片选逻辑
16K	$A_{13}\sim A_0$	$\overline{CS_0}=\overline{A_{15}}\,\overline{A_{14}}$
16K	$A_{13}\sim A_0$	$CS_1=\overline{A_{15}}A_{14}$
16K	$A_{13}\sim A_0$	$CS_2=A_{15}\overline{A_{14}}$
16K	$A_{13}\sim A_0$	$CS_3=A_{15}A_{14}\overline{A_{13}A_{12}A_{11}}$



注：16：10000

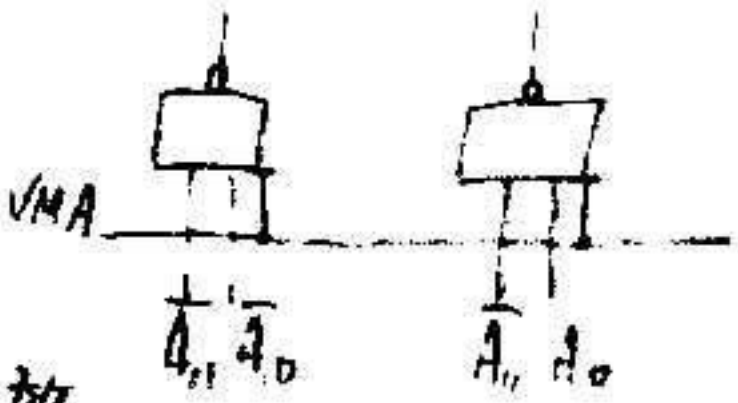
16K：10000 00 0000 0000

14：1110

14K：1110 00 0000 0000

5. 其他问题

① 有关控制信号



VMA $\overline{MREQ}$

加在片选逻辑输入端. 控制片选逻辑有效.

② 地址复用技术

16位地址两次送入芯片地址端

$\overline{RAS}$: 为低电平, 高8位地址(A_{15}~A_8)送芯片地址端 行选

$\overline{CAS}$: 为低电平, 低8位地址(A_7~A_0)送芯片地址端 列选

③ 芯片不宜排在连续空间, 片选逻辑也不连续.

§2-2 基本概念

1. 存储原理. P85

SRAM: 采用触发器内部交叉反馈存储信息. → 功耗大
(静态)

DRAM: 采用电容存储电荷原理存储信息. 需动态刷新.
(动态)

2. 动态刷新. (P107)

刷新周期定时方式: ①集中. ②分散. ③异步.

3. 存取方式 P87

① 随机存取方式. (ROM属于随机存取;)

a. 可按地址访问M中任一单元, b. 访问时间与地址无关.
与SAM比 与DAM比

⑤ 接口设计，接口组成，扩充命令，状态字格式。

6. 常用外设原理

① 键盘，键码转换方法。矩阵，扫描码的产生，转换为键码

② CRT 显示器：（VRAM 与屏幕显示的对应关系，存储容量，地址组织，信息转换，同步计算器设置）

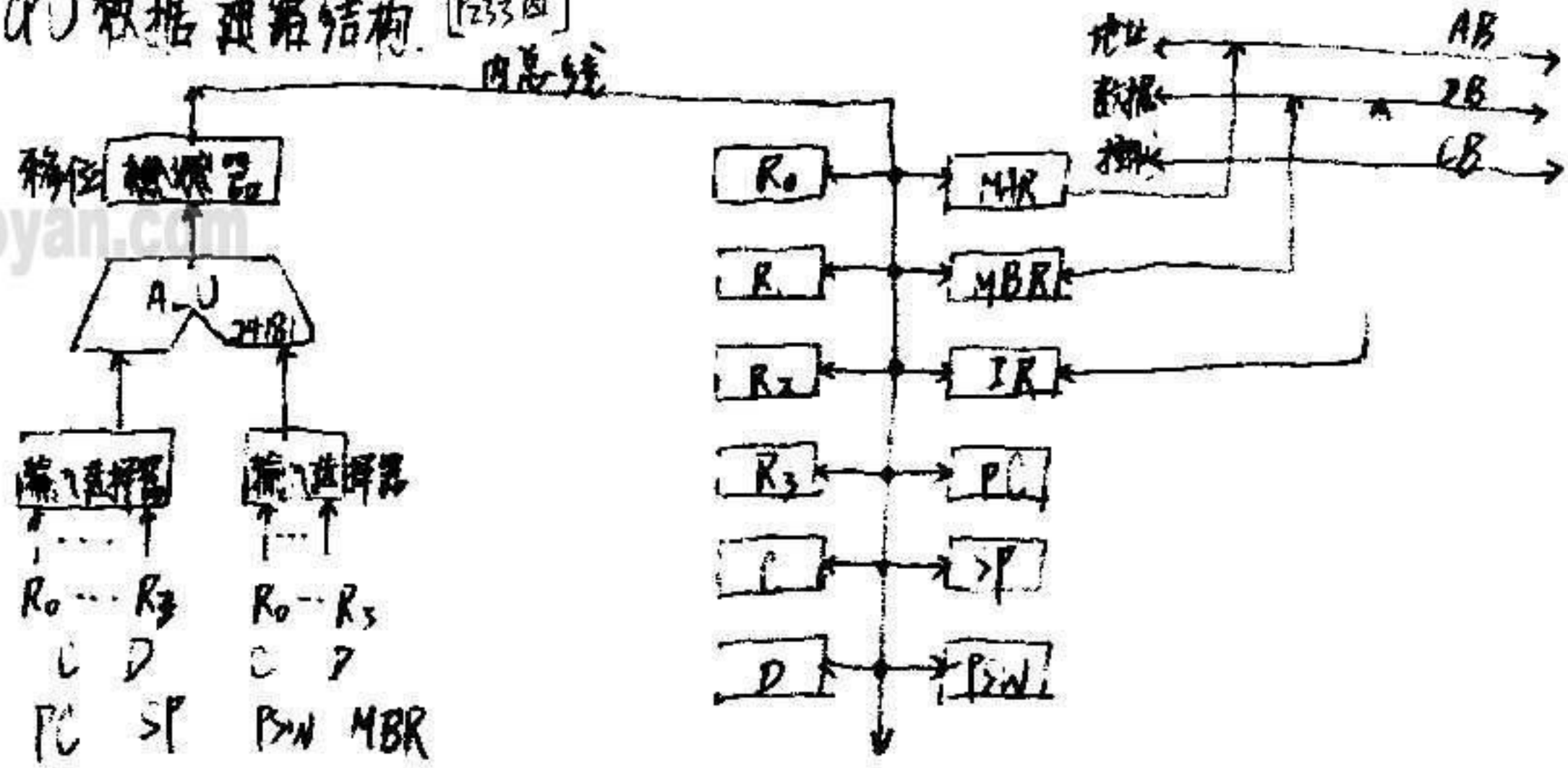
③ 打印机：信息转换，调用过程（中断方式）

④ 磁盘：信息格，磁记录方式，调用过程（DMA方式），磁盘指标，速度，容量

第一章 CPU 组织

1.1 逻辑组成

1. CPU 数据通路结构 [P33图]



2. 结构特点。CPU内部数据通路结构特点：采用自立寄存器，一组单向数据总线，以ALU为内部数据传送通路的中枢

① 寄存器：很比高

可编程：通用寄存器 $R_0 \sim R_3$，PC，SP，PSW

非编程：C，D，IR，MAR，MBR（透明的）

性能指标：包取时间

② 顺序存取方式 SAM

访问时按顺序查找数据，访问时间与数据位置有关。

性能指标：平均等待时间
数据传输率

③ 直接存取方式 DAM

访问时读写部件先指向某一个区域（磁道），再在这区域内顺序查找数据。

访问时间与数据所在位置有关。

性能指标：平均寻道时间（定位时间）
平均旋转等待时间
数据传输率

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用的信息传送线路。

2. 分类：

① 按功能划分：

1) 片内总线（芯片内）：CPU内部寄存器与算逻部件互连的直线一组数据线）。

2) 部件内总线（芯片间）：各计算机部件之间互连的总线。（地址、数据、控制线）

Δ 3) 系统总线（设备部件间）：系统各功能部件（组件机）之间互连的总线。

（地址、数据、控制线）

十 外总线：(通信总线)

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，数据控制线

⑤ 按时序控制方式分 (P27) [333-335]

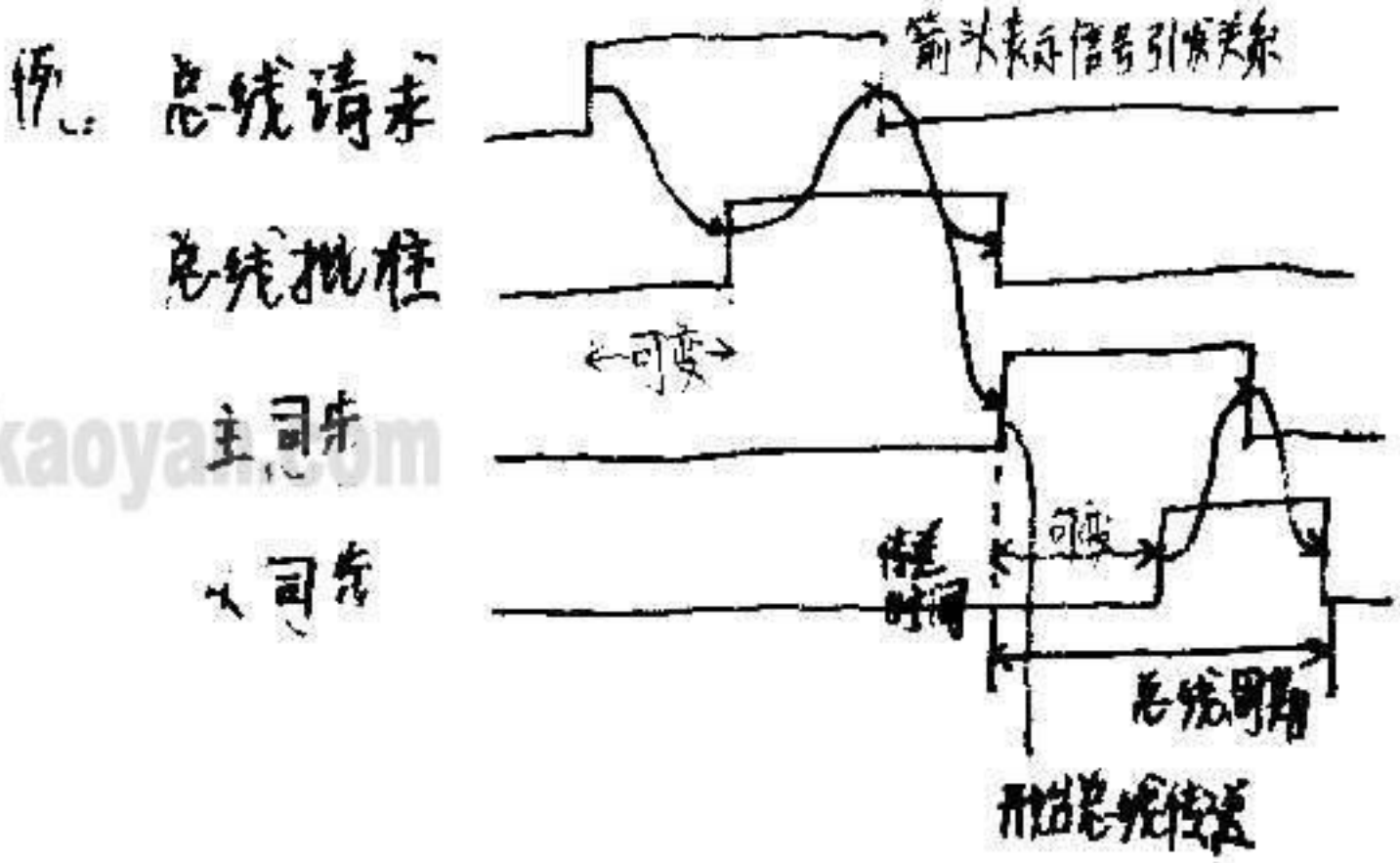
1) 同步总线：由控制模块提供统一时序的控制总线传送操作。

(1). 总线控制器　　时钟脉冲

2) 异步总线。

需长则长，需短则短

不采用统一时钟周期节拍，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。



主设备：申请并使用总线的设备（获得总线权）

从设备：响应主设备请求的设备（不掌握总线权）

* 不是指发送或接收来自

⑥ 按数据传送格式分 P26

1) 并行总线　用多根数据线同时传送一个字或一个字的所有代码位。

2) 串行总线　按位串行传送数据，即按数据代码位流的顺序逐位传送。

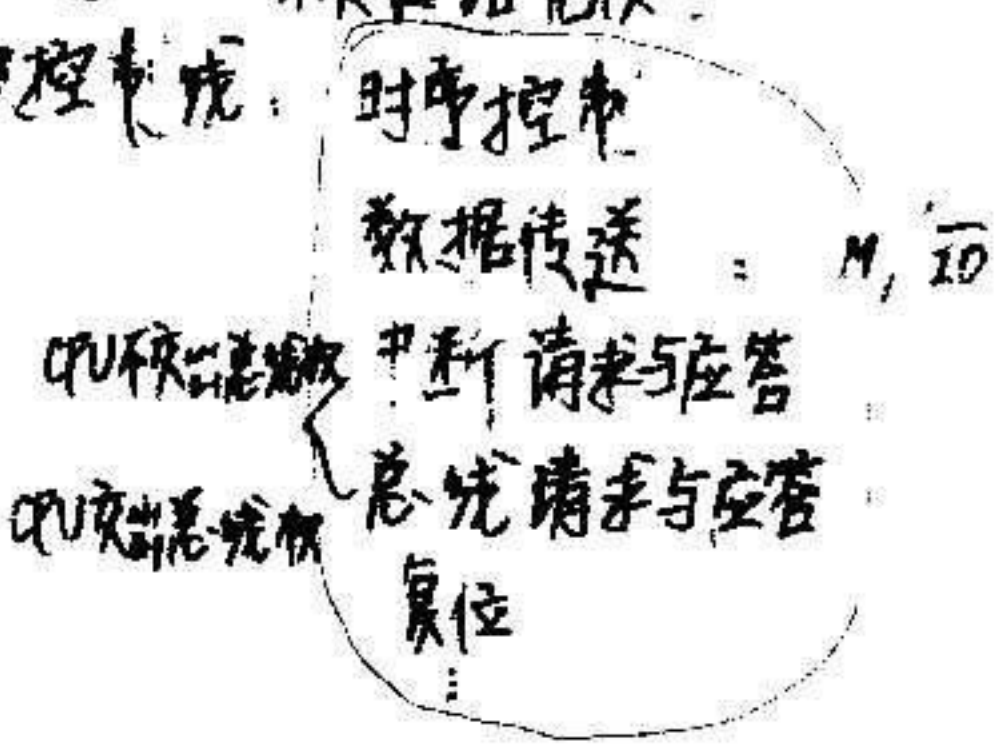
3. 系统总线的信号组：

1) 电源与地

2) 地址线：决定寻址空间大小.

3) 数据线：决定通路宽度.

4) 控制线：



3.2 接口 P78

定义：连接两个设备之间的支撑部件.

I/O接口位于系统总线和I/O之间.

2. 功能：P330

① 按数据传送格式分类

1) 并行接口：接口与系统总线之间、接口与外围设备之间，都以并行方式传送数据信息.

2) 串行接口：接口与设备之间采用串行方式传送数据，接口与系统总线之间仍采用并行方式传送数据.

② 按时序控制分类

1) 同步接口 与同步总线连接的接口，接口与系统总线间的信息传送由统一时序信号控制，接口与外围设备间则允许有相对独立的时序控制操作

2) 异步接口 与异步总线相连的接口，接口与系统总线间的信息传送采用异步应答的控制方式

③ 按I/O传送控制方式分类

1) 程序查询接口.

2) 中断接口 (中断接口往往可以覆盖程序查询方式的功能)

3) DMA接口.

§3.2 I/O 传送控制机制.

3.2.1 程序中断方式.

1. 实现原理

① 实质: 暂停执行程序, 转去执行为某[随机事态]服务的中断处理程序, 处理完毕后自动恢复原来程序的执行.

② 实质: 程序切换

切换时间: 一条指令结束时.

③ 特点. 随机性. { 随机发生的事态 (键盘)
自愿调用, 随机请求处理的事态 (打印机)
随机插入的事态 (软中断) 自愿调用, 随机插入

④ 应用.

处理中低速I/O的操作. 打印机 } 典型例
处理随机事态.

2. 中断服务程序入口地址的获得. P347

① 向量中断方式.

向量中断. 一组中断服务程序入口地址(一组矢量)

中断向量表. 存放中断向量的存储区, 一般位于主存首部

向量地址. 指向中断向量的首址.

4. 中断主过程（主机一方）

(1) 初始化

预置接口和中断控制器工作方式，设置中断类型码，送屏蔽字。

(2) 发命令字，启动设备（自愿调用）

(3) 设备申请中断

(4) 中断请求送中断控制器类优（排队）送出被请求IN

(5) CPU响应，发回批准信号INTA（关中断控制器），并保存断点，关中断

(6) 中断控制器送出中断类型号

(7) CPU将类型号转换为向量地址，查表，取入口 → 转服务程序

(8) 执行服务程序（进行中断处理，交换数据） ← 单级中断

(9) 开中断，返回原程序 ← 多重中断

多重中断：CPU在执行中断服务程序的过程中，允许响应处理更高级别的中断请求。

多重中断：进入服务程序后送新屏蔽字，并开中断。

5. 中断接口设计。

例：电梯接口（自拟命令要求）

命令字：允许中断　速度（00 慢速 01 快速 10 减速）　五转　暂停　启动

状态字：（8路输）

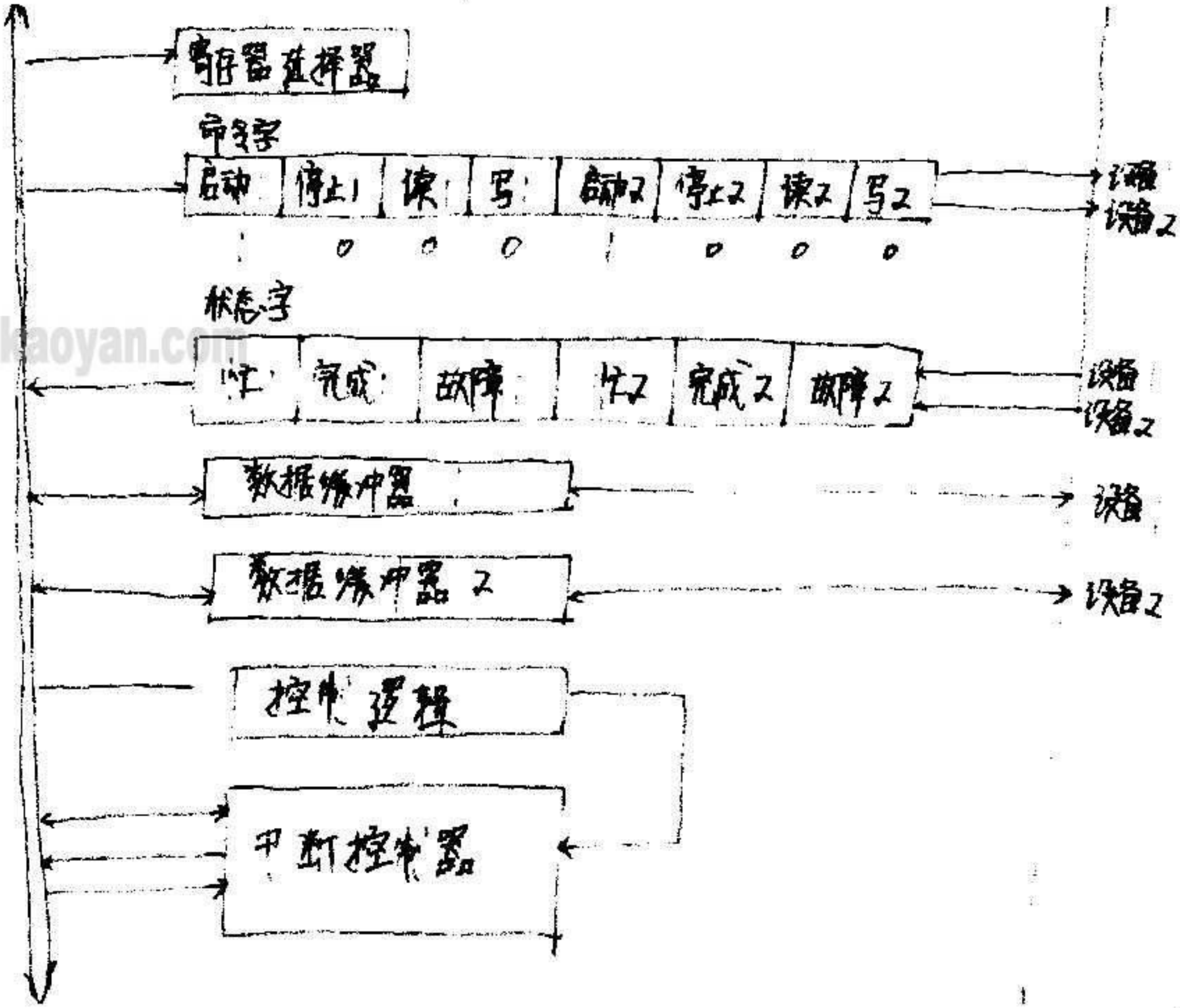
1	2	3	4	5	6	7	8

1 梯内请求
2 梯外请求
3 运行状态
4 故障

例2. 定时中断采集参数

练习. 某汇机系统有两个外中断源，实现两个设备的命令启动、停止、读、写，设备状态包括：完成、故障，设计有关接口。

(1) 接口组成：两设备共用一个接口。



6 中断源扩展（软件方法）

外设向CPU请求送IR以2. CPU响应 转IR以2服务程序（查询程序）
根据查询结果 转相应设备的服务程序.

3.2.2 DMA方式(P362)

1. 定义及应用

定义：直接依靠硬件在主存与I/O间传送数据，传送期间不需CPU程序干预.

应用：简单 批量数据传送. 一般应用于主存与高速I/O设备间的简单数据传送.

2. DMA主过程.

(1) 预处理阶段，CPU通过程序向DMA控制器和接口输送 预处理信息.
传送方向（读写） 主存缓冲区首址. 交换量 外设寻址信息.
启动设备.
→送DMA控制器 送接口.

(2) DMA传送阶段，由DMA控制器获得总线权，控制传送.

(3) 结束处理阶段. 批量传送完毕. 接口申请中断 CPU响应后执行中断服务程序，作善后处理.

第四章 主要I/O设备原理.

3.1 CRT显示器 [P305-307]

3.1.1 显示器 方式与分辨率.

字符. 分辨率：一帧显示的字符数 25×80 n行×m列

图形. 分辨率：一帧显示的像点数 1024×768 n点×m线

3.1.2 显示缓存VRAM与屏幕显示的对应关系

1. VRAM 容量计算

字符方式：内容，字符编码 容量 25×80= 次 一个字符编码占存储的一个字节

图形方式：内容，图形点代码 容量 640×200= $\frac{640\times200}{8}$ =16K

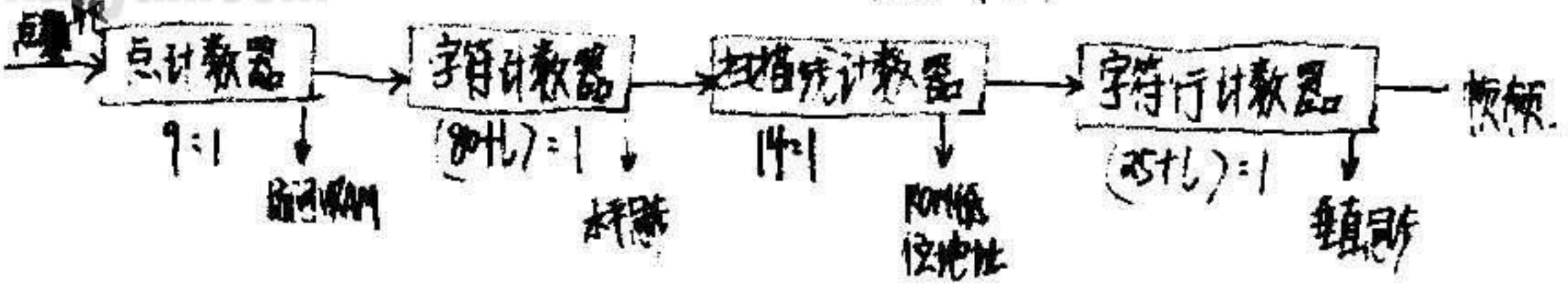
2. 信息转换

字符：VRAM 字符编码 → ROM 一行点阵代码 → 移位R 串行视频信号 → 显示器

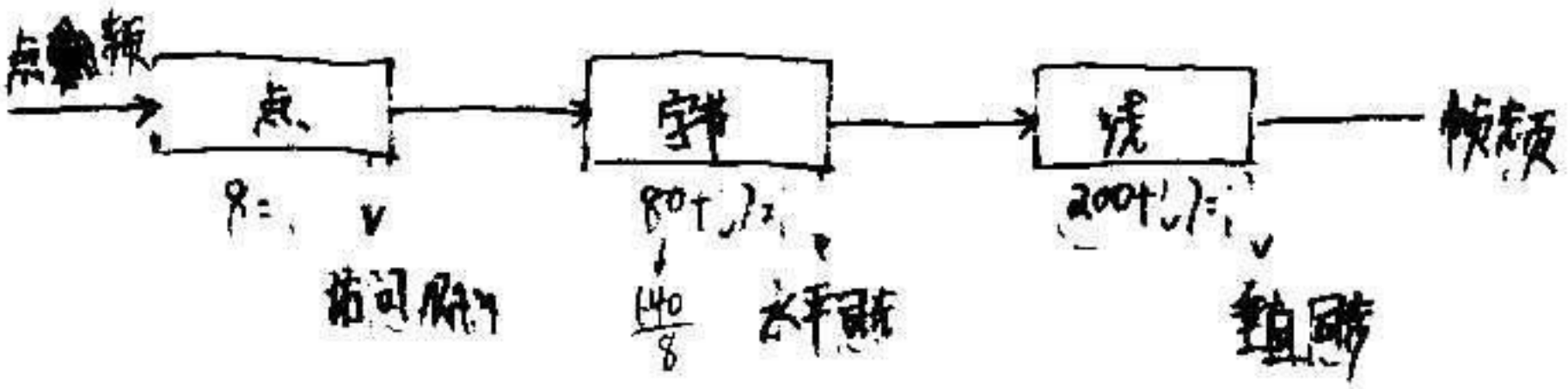
图形：VRAM 图形一行点代码 → 移位R 串行视频信号 → 显示器

3. 同步控制

① 字符方式 25×80 字符点阵 7×9 点阵5：9个



② 图形方式 640×200



§4.2 磁盘

1. 寻址信息：①驱动器号(台号) ②圆柱面号(道号) ③磁头号(面号) ④扇区号/数据块号 ⑤交换量(扇区数)

台号；圆柱面号；磁头号；扇区号；扇区数。

2. 磁记录方式

磁盘：调频(FM)和改进调频(MFM)。(P109)

3. 速度指标 (P131)
   ① 平均寻道(定位)时间
   ② 平均旋转延迟(等待)时间
   ③ 数据传输率(bit/s 或字节/s)

§4.3 磁盘

磁盘阵列 磁盘驱动器 P291-292

§4.4 打印机。

调制技术(中频)

C：暂存来自主存的源信息

D：暂存来自主存的目的信息

MAR、MBR：CPU与主存的接口

(2) 算逻部件 ALU

内部数据传送通路的中心，便于集中控制，方便各种运算。

(3) 内总线

一组单向数据总线，实现信息分配。

单向时序脉冲打入。

(4) 与系统总线的连接

MAR、MBR的输出通过三态门与系统总线连接

MBR { 输入 { 从内总线接收数据(打入)，内部传送
　　 { 从DB接收数据，(置入)，外部传送
　　输出 { 向ALU输出数据(内)
　　 { 向DB输出数据(外)

IR：从DB接收指令(置入)

扩展：寄存器采用多端口结构

内总线(双向，多组，多对)

第3章 CPU结构

3.2 指令流程(寄存器传送级)　注：M→MBR→IR(老)
　　M→IR　(新)

3.2.1 基本寻址方式 [P230]

R.　(R)　-(R)　(R)+　X(R)

-(SP)　(SP)+　(PC)+　X(PC)　@(PC)+ 直接寻址 [P230]

立即寻址　相对寻址

思路

① 明确概念-功能

MOV 源 → 目的

ADD 运算结果 → 目的

JMP 转移地址 → PC 转移指令

JSR 保存返回地址 转子指令
子程序入口 → PC

RST 返回地址 → PC 返回指令

② (PC)+ : 直接寻址

操作数地址紧跟着指令. 编程时将操作数地址存放在紧跟指令的单元中, 取指后PC+1, 将修改后的PC内容作为地址, 据此访问紧跟现行指令的存储单元(间址单元), 从中取得操作数地址(称为绝对地址), 据此再次访问, 读得操作数. 然后PC+1, 指向后继指令.

③ 确定寻址方式

④ 确定执行流程. 如何有序: ①通路有无冲突 如M→IR问题 ②时间上是否允许 如C→M问题 C→MBR→M(X)

1) 一次M读出操作. 并经内部数据通路传送的操作

2) 一次内部数据通路的传送操作.

3) 一次向M写入的操作

kaoyan.com

3. 问题

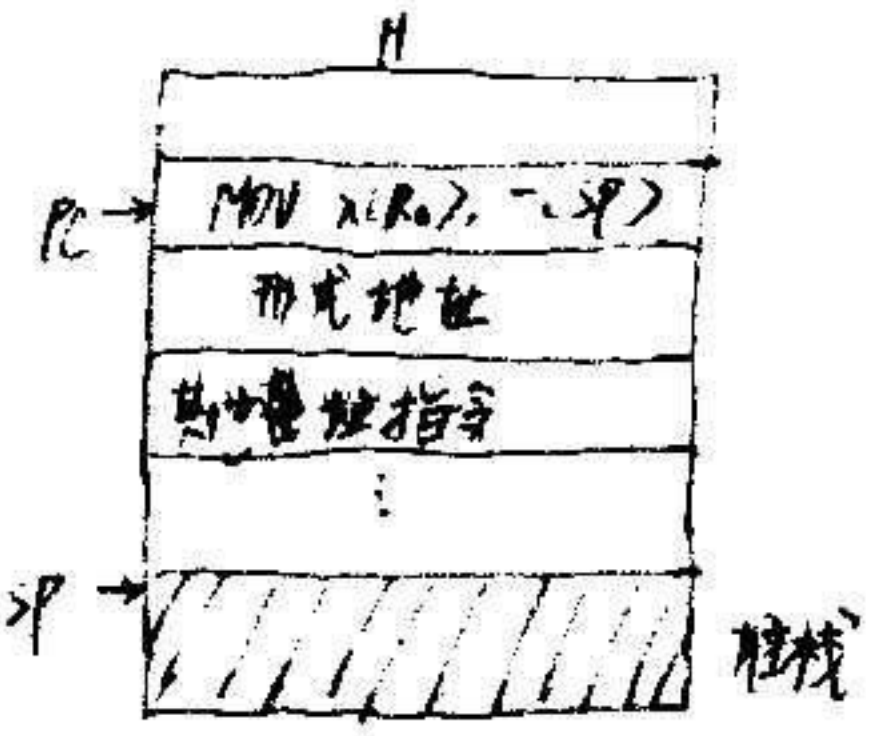
① MOV X(R0), -(SP)

FT: M→IR PC+1→PC

ST: PC→MAR
M→MBR→C — 形式地址.
C+R0→MAR
M→MBR→C — 源操作数.
PC+1→PC

DT: SP-1→SP, MAR

ET: C→MBR
MBR→M
PC→MAR



时序: 同上生成.

M→IR, PC+1→PC 可同时进行. 因为它们各自使用的数据通路无冲突. 读取指令经由数据总线, 而PC+1经由ALU与内部总线.

ADD (R0), (R3)+, 源变址的一种寻址标注符号

FT: M → IR, PC+1 → PC

ST: SP → MAR

M → MBR → C

SP+1 → SP

DT: PC → MAR

M → MBR → D — 形地

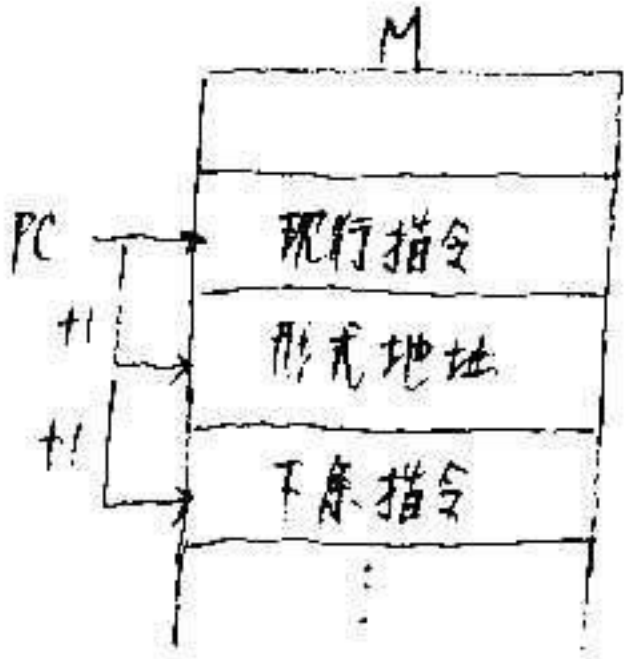
R1 + D → MAR — 目的

PC+1 → PC

ET: C → MBR

MBR → M

PC → MAR



③ ADD (R0), (R3)+,

FT: M → IR, PC+1 → PC

ST: R0-1 → R0, MAR

M → MBR → C

DT: PC → MAR

M → MBR → D — 形式地址

PC+1 → PC

R3 + D → MAR

M → MBR → D — 目的操作数.

ET: C+D → MBR

MBR → M

PC → MAR

④ ME_1 R_2, 寄存

FT: M → IR. PC+ → PC

ET: $\overline{R_2}$ + 1 → R_2

PC → MAR

⑤ JMP (R0) JMP: 转移地址 → PC

FT: M → IR, PC+ → PC

ET: R0 → MAR

M → MBR → PC, MAR

⑥ JSR (R0) 转指令: 保存返回地址; 子程序入口 → PC

FT: M → IR. PC+ → PC

ST: R0 → MAR

M → MBR → C ─ 子程序入口.

ET: SP−1 → SP, MAR

PC → MBR

MBR → M

C → PC. MAR

} 保存返回地址.压栈

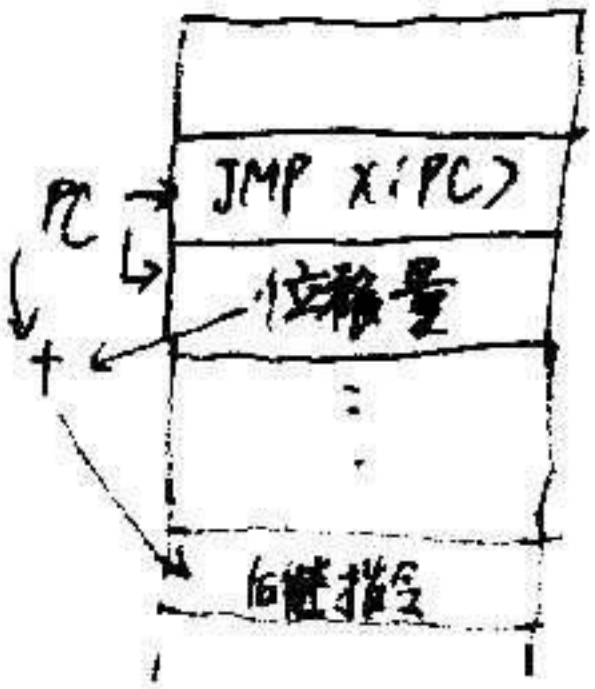
⑦ JMP X(PC), X(PC): 相对寻址(变址)

FT: M → IR, PC+ → PC

ET: PC → MAR

M → MBR → C ─ 位移量

C+PC → PC MAR



③ AN[,][,]

立即 相对

FT: M→IR, PC+ →PC — 取数地址

ST. PC→MAR

M→MBR→C — 立即数

PC+1→PC — 位移量地址.

DT. PC→MAR

M→MBR→D — 位移量

D+PC→MAR

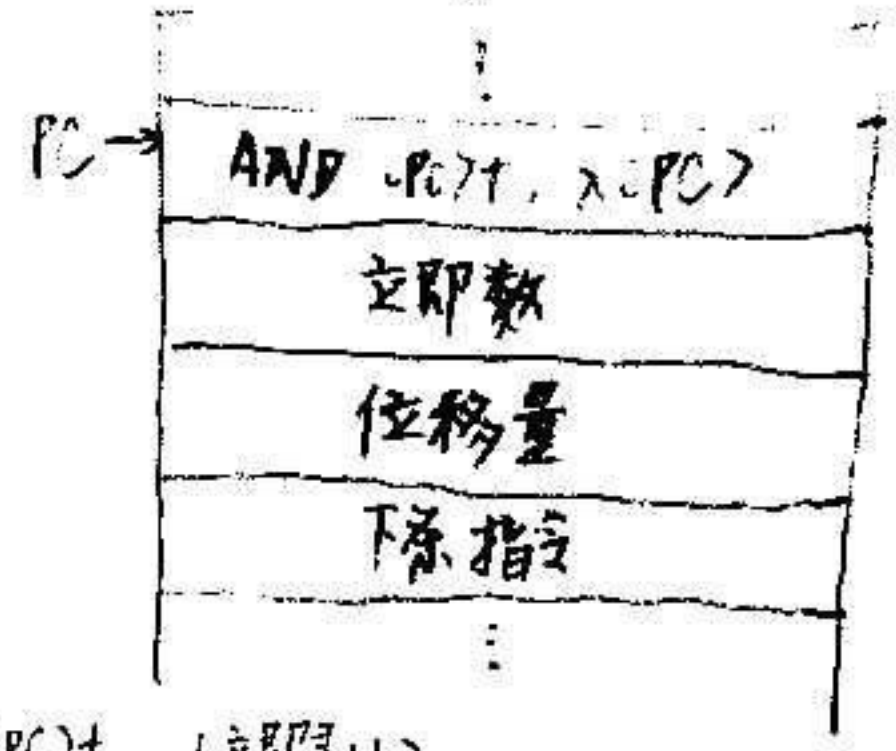
M→MBR→D — 目的数

PC+1→PC 下条指令地址

ET: C·D→MBR

MBR→M

PC→MAR



(PC)+: 立即寻址

立即数紧跟着指令，编程时将立即数存放在紧跟指令的单元中，取指后 PC+1，将修改后的PC内容作为地址，可读取立即数，取出立即数后PC内容再加1，指向立即数之后的一个单元，即后继指令

X(PC): 相对寻址

取指后 PC+1，以修改后的PC内容作为地址，从紧跟现行指令的存储单元中读取位移量，再与PC内容相加，获得操作数地址。因此操作数地址是以PC值为基准，加上位移量形成的。当前程序存在另一块存储区中，地址的相对关系不变

1.2.2 微命令序列 同流数据传送或时序部件的需要来拟定微命令

1. 微命令设置

① 数据通路指令.

ALU输入选择: $R_0 \to A$, $R_1 \to A$ / $R_0 \to B$, $R_1 \to B$

ALU功能选择: 加. 减. 传送.

移位功能选择: 左移. 右移. 直送(不移)

结果分配: 打入R_0 … 打入MAR (打入脉冲) → 脉冲型 ⊓

(CPR0 … CPMAR)

② 访存操作.

地址使能 EMAR ⊓

读R. 写W (R, W)

置1：

③ 时序控制（组合逻辑控制方式）时序切换 P238-239

周期状态的设置、清除。

→FT　→ST …　CPFT　CPST …

时钟计数　清除。TH，CPT

2. 例题。

MOV (>P)T, (R0),

FT: M→IR；EMAR. R. SIR.

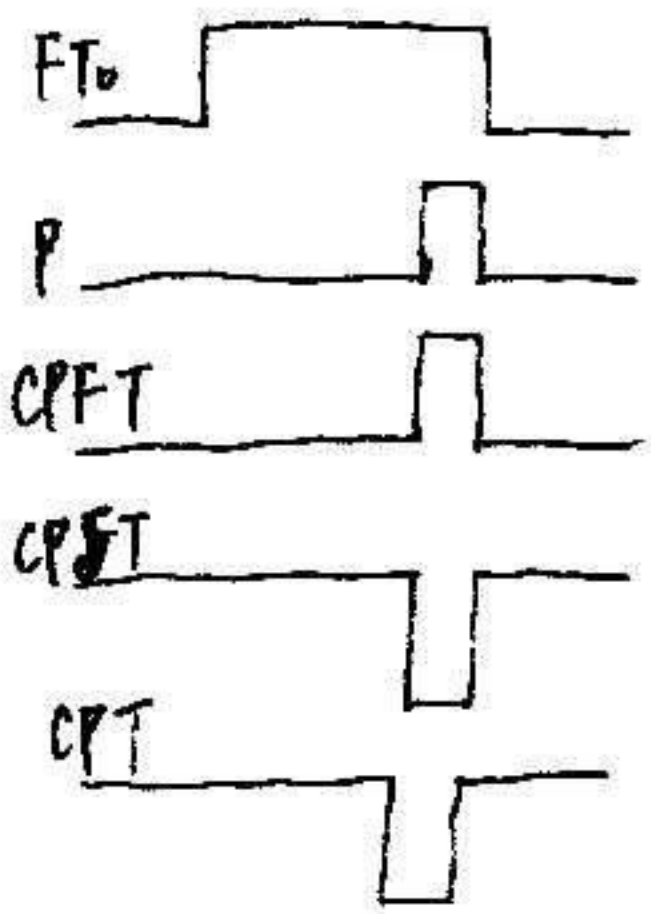
PC+1→PC. PC→A. At.. DM, CPPC

I→ST. CPFT(P̄). CPST(P̄)

CPT(P̄)

CPPT(P̄). CPET(P̄)

空操作。化简



ST: SP→MAR. SP→A 传送A

DM. CP.MAR

T+1 CPT(P̄)

M→MBR→C: EMAR. R. SMBR.

MBR→B. 传送B

DM. CPC, TH CPT.P̄)

$T_1 \rightarrow T$: $1 \rightarrow A$. $A1$. PM. $\overline{1}$

$1 \rightarrow DT$. $CPFT(\overline{P})$. $CPST(\overline{P})$.

$CPDT(\overline{P})$. $CP\overline{=}(\overline{P})$. $CPT(\overline{P})$ 有效.

DT: $R_0 \rightarrow MAR$: $R_0 \rightarrow A$. 传送A. PM. CP MAR

$\rightarrow ET$, $CPT(\overline{P})$ $CPFT(\overline{P})$. $CPST(\overline{P})$

$CPDT(\overline{P})$. $CPET(\overline{P})$.

ET: $C \rightarrow MBR$: $C \rightarrow A$. 传送A. PM. CP MBR Tt: $CPT(\overline{P})$

$MBR \rightarrow M$: EMAR. W. Tt: $CPT(\overline{P})$

$PC \rightarrow MAR$. $PC \rightarrow A$. 传送A - PM. CP MAR

$\rightarrow FT$. $CPF(\overline{P})$ $CPST(\overline{P})$.

$CPDT(\overline{P})$. $CPE(\overline{P})$. $CPT(\overline{P})$

1.2.3 微命令产生方式.

1. 组合逻辑控制方式.(硬连逻辑)

① 基本思想.

将合成需产生微命令的条件, 形成相应逻辑式, 用组合逻辑电路实现. 执行机器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.

速度快, 结构(第一次)零乱. 设计不规整. 设计效率低. 不易检查调试, 不易修改. 扩展指令系统功能.