建立整机概念.

二个层次： CPU. 系统.

二个方面. 逻辑组成，工作机制.

1. CPU

① 逻辑组成：寄存器级的部件设置、数据通路结构

② 工作机制 < 指令流程（寄存器传送级）
微命令序列 < 产生方法：组合逻辑控制. 微程序控制
时序控制方法，同步控制 （CPU内部是同步控制）

2. 基本算法 规则 （澄清一些模糊问题）

重点：补码位乘. 原码不恢复余数除 、补码不恢复余数除. 浮点四则运算. 进位信号 逻辑

3. 半导体存储器 逻辑设计*

4. 系统总线

① 信号组成

② 控制方式与时序关系 < 同步方式及打版 什么是同步、异步，用时序波形描绘
异步方式

5. 接口*

6. I/O 传送的控制机制

① 中断，基本概念 中断控制器与接口 . 中断过程. 应用 针对某种应用描述

② DMA，基本概念、DMA控制器与接口. DMA过程 应用

7. 常用外设原理.

① 键盘， 矩阵、扫描码的产生. 转换为键码.

3. 澄清

* 统一的时序信号，不一定是由CPU发出。
* 同步控制并非所有指令执行时间相同

  应是：时钟周期固定，指令周期、工作周期长度可变

指令控制都是同步控制，异步控制用于与外部设备打交道

*. 异步控制一般不用于CPU及设备内部

*. CPU与I/O设备可以有各自的时序系统，但它们之间的信息传送可以用同步控制方式协调。

六. RISC技术.

1. 选取使用频率高的一些简单指令，以及一些很有用但不复杂的指令。
2. 采用简单的指令格式（如固定指令字长，指令格式种类少，寻址方式简单等）
3. 采用多级指令流水线结构，提高流水效率，多数指令的执行周期很短。

   平均 1.3T/条，甚至更多。

4. 减少访存次数

① 采用加载/存储结构，只有取数/存数指令才可以访存。

② 增多CPU中寄存器数。

③ 采用Cache

● 5. 用硬连逻辑控制，不用或少用微程序控制。

6. 优化编译技术以减少指令数、数据合理分配以减少访存次数。

   减少重复技术

七. 运算方法与运算器

1. 补码一位乘（Booth法）

① 何谓补码运算：用补码表示操作数

   直接符号位一起运算。

② 规则.

y_n	y_{n+1} (补)	操作
0	0	原部分积右移一位
0	1	原部分积 + X补位, 再右移一位
1	0	原部分积 − X补位, 再右移一位
1	1	原部分积右移一位

2. 原码加减交替除

① 何谓原码运算.

用原码表示操作数, 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.

② 规则.

A. 余数 r_i 为正(符号位为0). 商 1, 下一步, $2r_i - |y|$
余数 r_i 为负(符号位为1). 商 0, 下一步, $2r_i + |y|$

$r_{i+1} = 2r_i + (1 - 2q_i)y$

b. 尾数运算后, 先让余数为正, 然后再加符号.

3. 补码加减交替除

a. r_i、Y同号 商1, 下一步, $2r_i - y_补$
r_i、Y异号, 商0, 下一步, $2r_i + y_补$

b. 假商修正: 假商符加1(或) 末位商置1

假商加 $(1+2^{-n})$

4. 浮点四则运算 P33 P74-77

① 加减中对阶 小阶向大阶对齐 即: 以大的阶码为基准, 调整小的阶码.

② 乘 $R^E \cdot M$

③ 除

5. 硬件

全加器
求位链 > 并行加法器 > ALU
加法器输入选择

寄存器，堆存器

ALU 输入选择

移位器

运算器

① 串行进位并走逻辑式 P45

② 并行进位链逻辑式

③ 分组、月级进位链，逻辑式

第二讲 存储子系统

一. 半导体存储器逻辑设计 P105

1. 例1：主存容量 4KB，可选存储芯片 1K×4/片，地址总线 $A_{19}\sim A_0$（低），双向数据总线 $D_3\sim D_0$（低） R/W 控制读写

① 存储空间分配与芯片

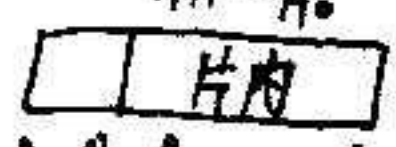
0000

1K×4	1K×4
1K×4	1K×4
1K×4	1K×4
1K×4	1K×4

先确定所需芯片数，并进行存储空间分配，作为片选逻辑的依据

② 地址分配与片选逻辑

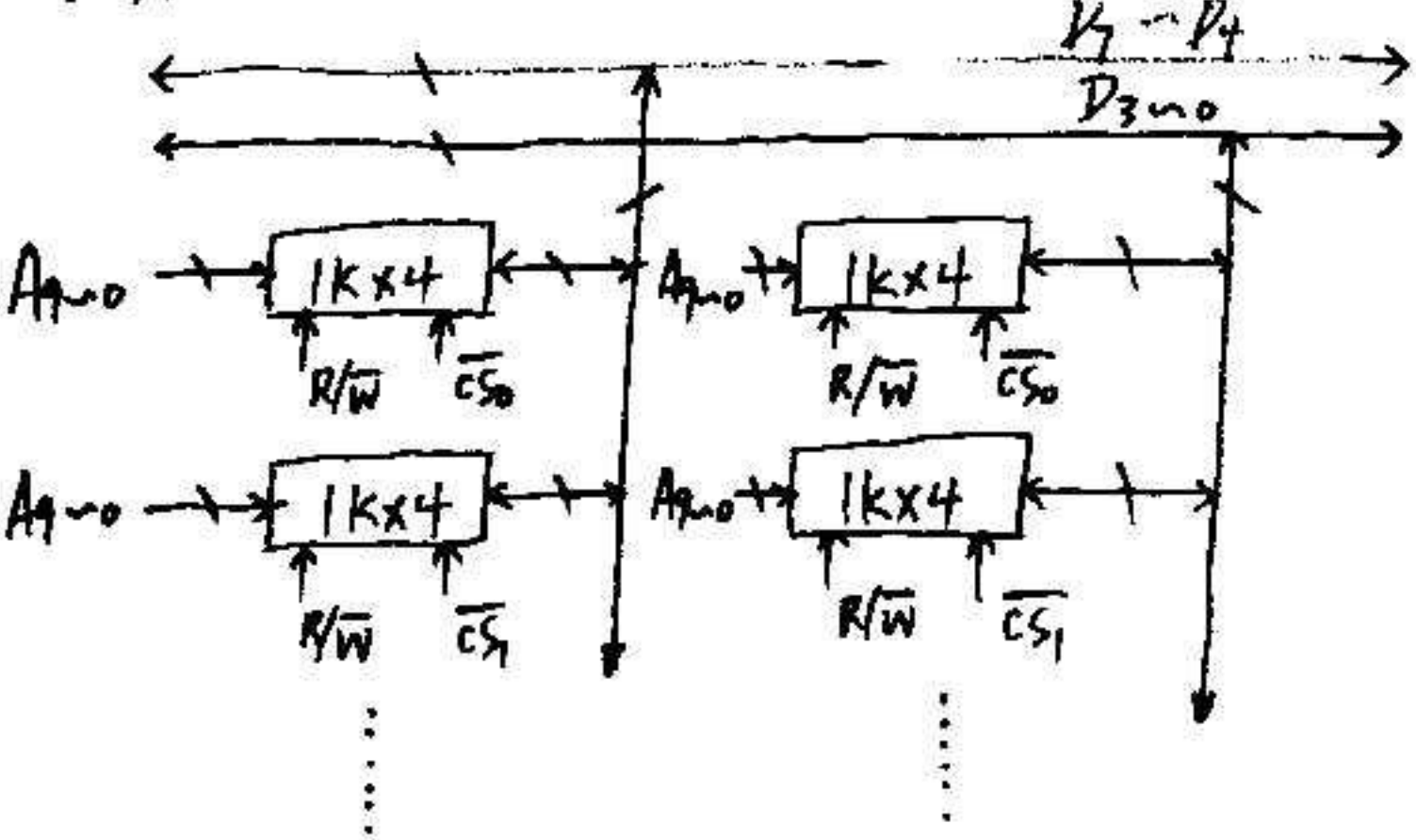
芯片容量	芯片地址	片选逻辑
1K	低10位 $A_9\sim A_0$	$\overline{CS_0}=\overline{A_{11}}\,\overline{A_{10}}$
1K	$A_9\sim A_0$	$\overline{CS_1}=\overline{A_{11}}\,A_{10}$
1K	$A_9\sim A_0$	$\overline{CS_2}=A_{11}\,\overline{A_{10}}$
1K	$A_9\sim A_0$	$\overline{CS_3}=A_{11}\,A_{10}$



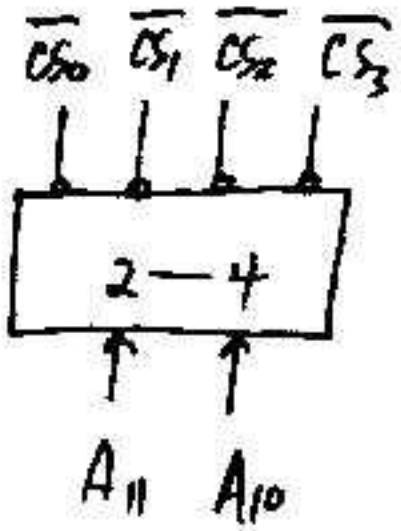
总容量是4K单元，共需12位地址 $A_{11}\sim A_0$，高8位 $A_{19}\sim A_{12}$ 恒为0，可以舍去不用。

对于每片芯片，应将低10位地址 $A_9\sim A_0$ 连接芯片，余下的高位 A_{11} A_{10} 作为片选依据

③ 逻辑图

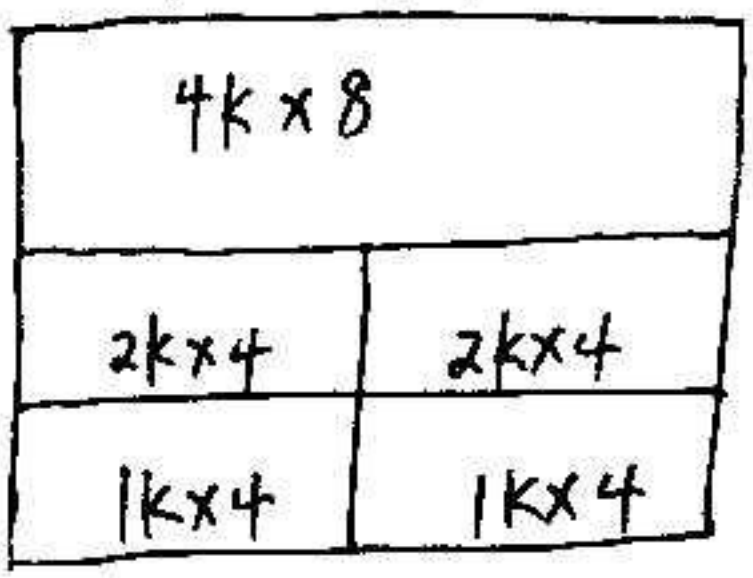


注意片选信号是低电平有效



2. 例2. 主存容量 7KB, ROM区 4KB, 可选芯片: EPROM 4K×8/片、RAM 2K×4/片
1K×4/片

① 存储空间分配与芯片



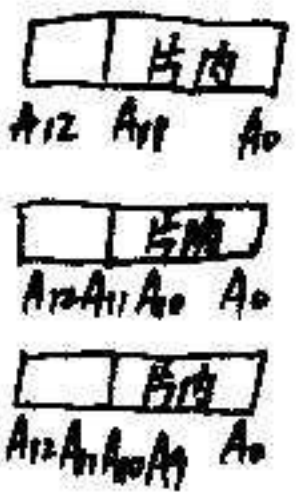
先确定所需芯片数，并进行存储空间分配，作为片选逻辑的依据

一般从大到小，这样片选逻辑较简单

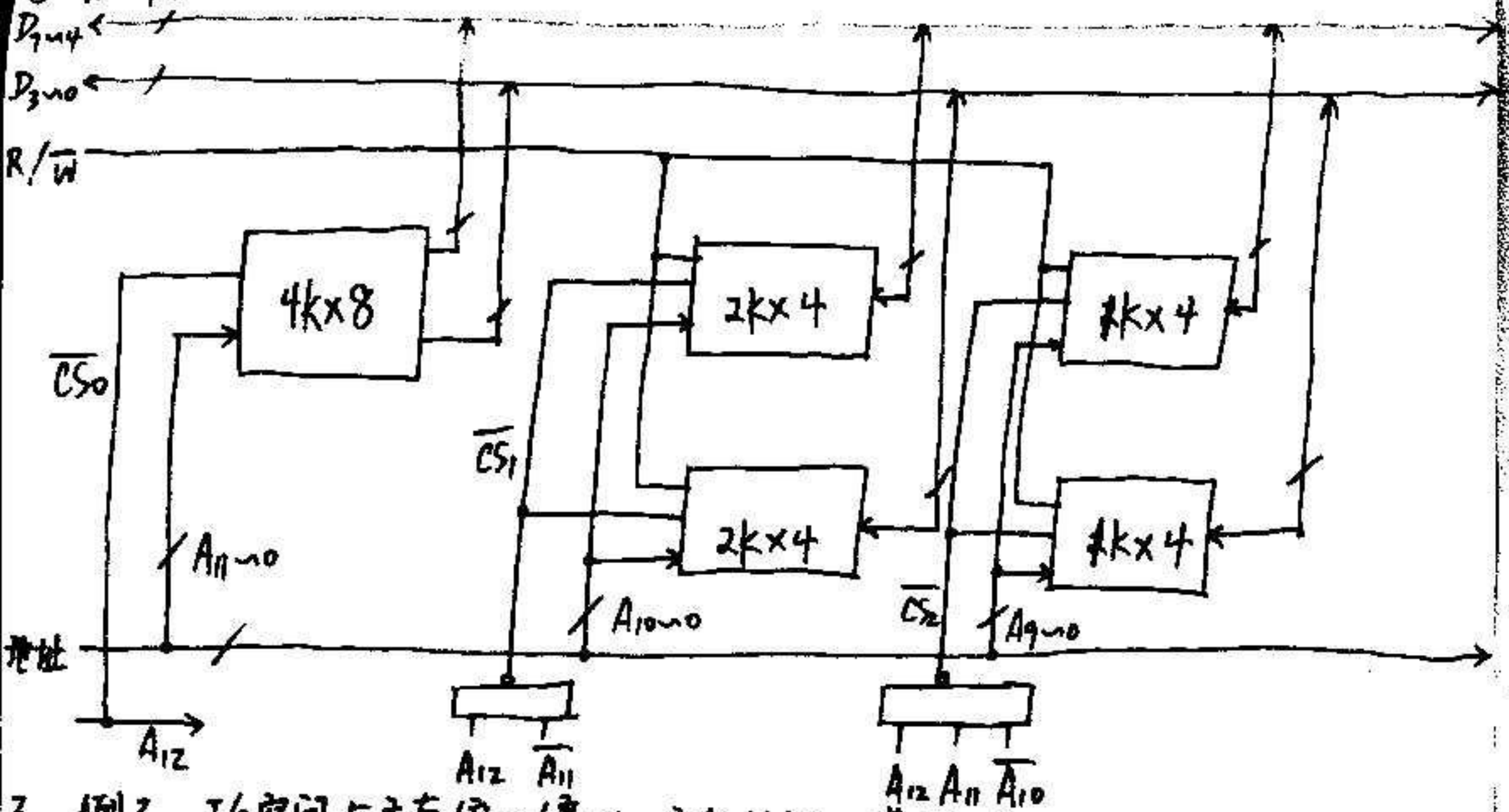
② 地址分配与片选逻辑

芯片容量	芯片地址	片选逻辑
4K	低12位 $A_{11}\sim A_0$	$CS_0 = A_{12}$ $\overline{A_{12}}$
2K	低11位 $A_{10}\sim A_0$	$CS_1 = A_{12}A_{11}$ $A_{12}\overline{A_{11}}$
1K	低10位 $A_9\sim A_0$	$CS_2 = A_{12}A_{11}A_{10}$ $A_{12}A_{11}\overline{A_{10}}$

片选信号是低电平有效

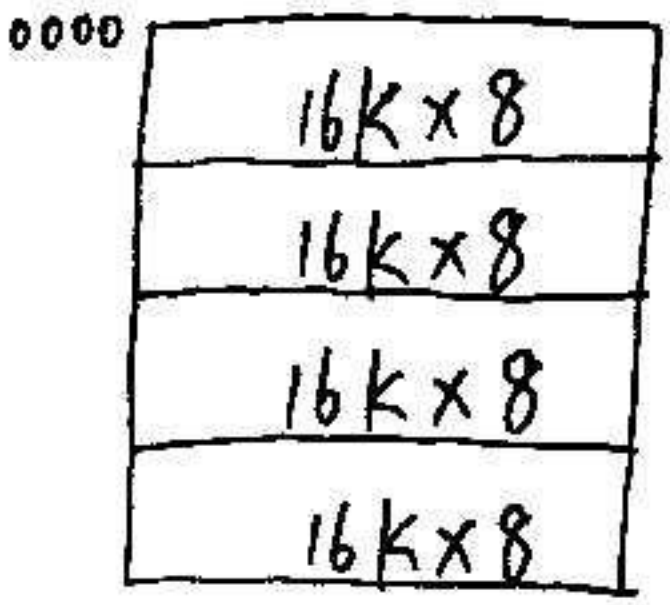


③ 连接图



3. 例3. I/O空间与主存统一编址，主存64KB. 其中高端1K，用于I/O编址，可选芯片 16K×8/片

① 存储空间分配与芯片

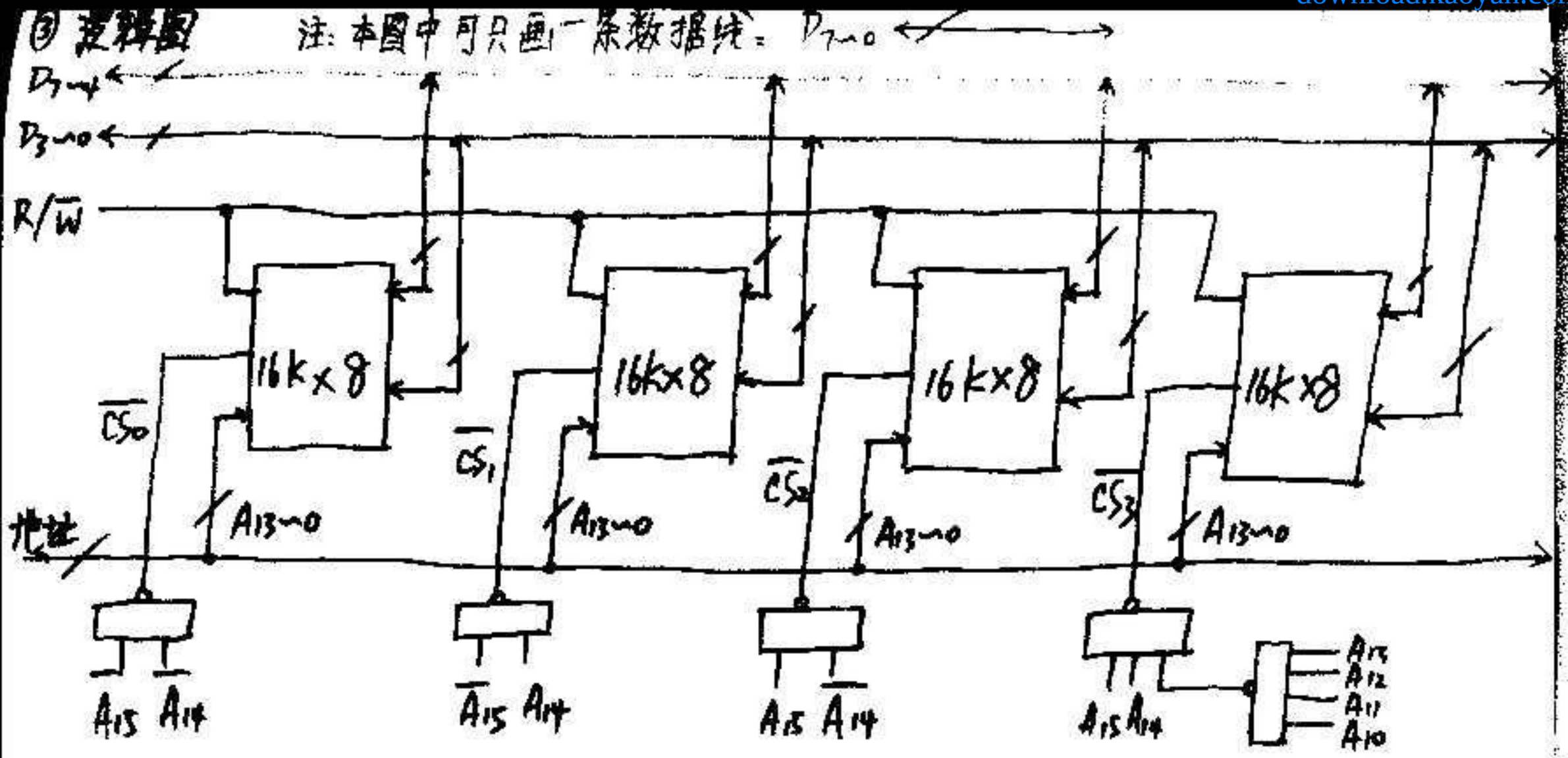


先确定所需芯片数，并进行存储空间分配，作为片选逻辑的依据.

② 地址分配与片选逻辑

芯片容量	芯片地址		片选逻辑
16K	最小地址	$A_{13}\sim A_0$	$CS_0=\overline{A_{15}}\,\overline{A_{14}}$
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		$A_{13}\sim A_0$	$CS_3=A_{15}\,A_{14}\,\overline{A_{13}\,A_{12}\,A_{11}\,A_{10}}$

③ 波特图　注：本图中可只画一条数据线：$D_{7\sim0}$



4. 其他变化

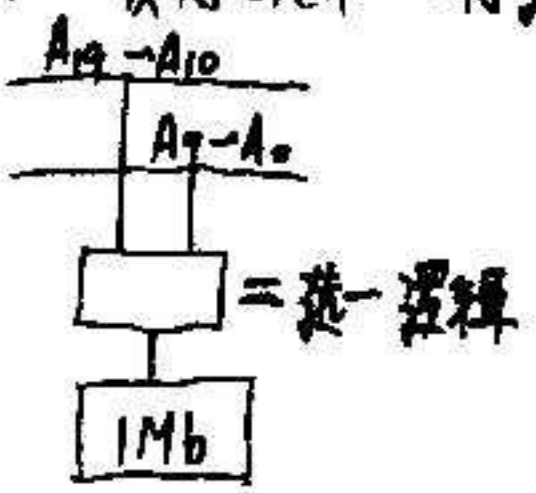
① 控制信号 $\overline{MREQ}$、VMA　　2-4 —$\overline{MREQ}$ 使能端

② 编址空间计算

给出十进制地址范围如 0～4095　　4k

给出十六进制地址范围如 0～FFFF　　64k

③ 地址复用技术（行选、列选）



④ 数据通路宽度匹配（P109）　　8086中：规则字，不规则字→从奇单元开始的字（从偶单元开始的字）

⑤ 有效空间不连续，片选逻辑相应变化

二. 基本概念.

1. 存储原理 P85

SRAM: 依靠双稳态电路内部交叉反馈机制保存信息 —— 不需动态刷新

PRAM. 依靠对电容充放电荷来存储信息. 保持一定电荷为1, 无电荷为0 —— 需动态刷新

2. 动态刷新方式 (刷新周期安排方式) P107

①集中刷新. ②分散刷新. ③异步刷新.

3. 易失性 (挥发性)

依靠低功耗CMOS芯片, 加后备电源

4. 随机存取方式 P87

① 可按地址直接访问任一单元 (与SAM比)

② 存取时间和地址无关 (与PAM比)

SAM: 顺序存取方式

PAM: 直接存取方式

5. 高速缓存 Cache

Cache 位于CPU与主存之间, 存放最近要用的程序和数据, 作为主存中当前活跃信息的副本, 容量不大, 速度很高.

6. 虚拟存储器 VS

虚拟存储器是指在OS支持下为用户提供的虚拟存储空间:

① 该虚拟空间远大于实际主存

② 用户可使用虚拟地址即逻辑地址编址, 不必过问主存物理细节.

③ 访存时, 计算机自动地将逻辑地址转换为主存物理地址

④ 实际上只有当前要运行的程序和数据放在主存中, 大部分程序和数据放在外存中, 依靠OS进行自动的替换.

第三讲　I/O 子系统

一. 总线与接口的基本概念

1. 总线：一组能为多个部件分时共享的公共信息传送线路，它分时接受与发送信息

2. 几种层次的总线（按任务分类）

① CPU 内总线（芯片内）

② 部件（插件）内总线（芯片间）

③ 系统（内）总线（设备 部件间）

④ 外总线（通信总线）.（系统与系统之间，系统与通信设备之间等）.

* CPU 内总线与系统总线的比较.

功能任务，信号组成，时序控制方式，可扩展性

3. 按时序控制方式分类 [32]

同步总线

异步总线

4. 接口：设备硬、软设备之间的连接部件（界面连接）[328]

外围接口（I/O 接口）：系统总线（主机）与 I/O 设备之间的接口

* 硬件接口，软件接口，硬软接口

5. 接口的分类 [330]

① 按数据传送格式分.

串行接口：接口与总线之间 并行，接口与设备之间 串行

并行接口，- - - - 并行，- - - - 并行

② 按与总线间的 数据 传送的时序 控制 方式分 同步接口 异步接口

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自独立的时序控制操作.

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 按与主机间的I/O控制方式分

直接程序控制接口.

中断接口. (中断接口往往可以覆盖程序查询方式的功能)

DMA接口.

二. 系统总线

1. 信号组成

① 电源线与地线

② 地址线

③ 数据线

④ 控制信号线

a. 时序控制信号: 同步定时信号(时钟, 状态)

异步应答信号

b. 数据传送控制: IO/M. R/W · 字节/字.

c. 中断请求与批准

d. 总线请求与批准 (含DMA)

e. 优先排队 入/出

f. 复位.

2. 同步总线

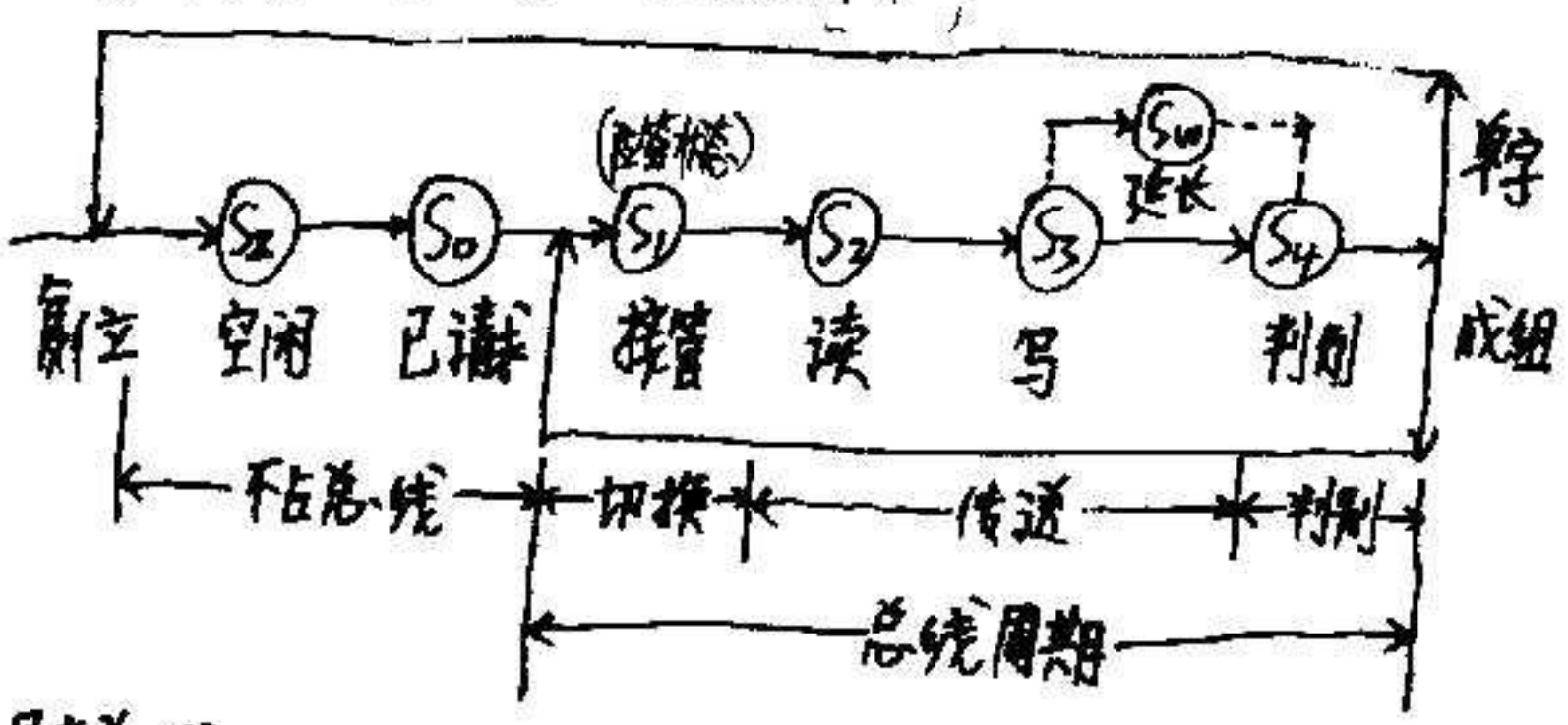
① 定义. 总线传送操作由统一的时序信号同步, 其主要特征是有严格的时钟周期划分.

② 时序举例: 如CPU访存, 总线周期一般固定为几个CPU时钟周期

3. 同步扩展方式.

① 定义. 以同步总线为基础, 但可"实现"异步事件同步化". 如应答时间为时钟周期的整数倍, 又如总线周期为时钟周期的整数倍, 时钟周期数可变.

③ 时序举例：PC机 PCI方式.[3P]

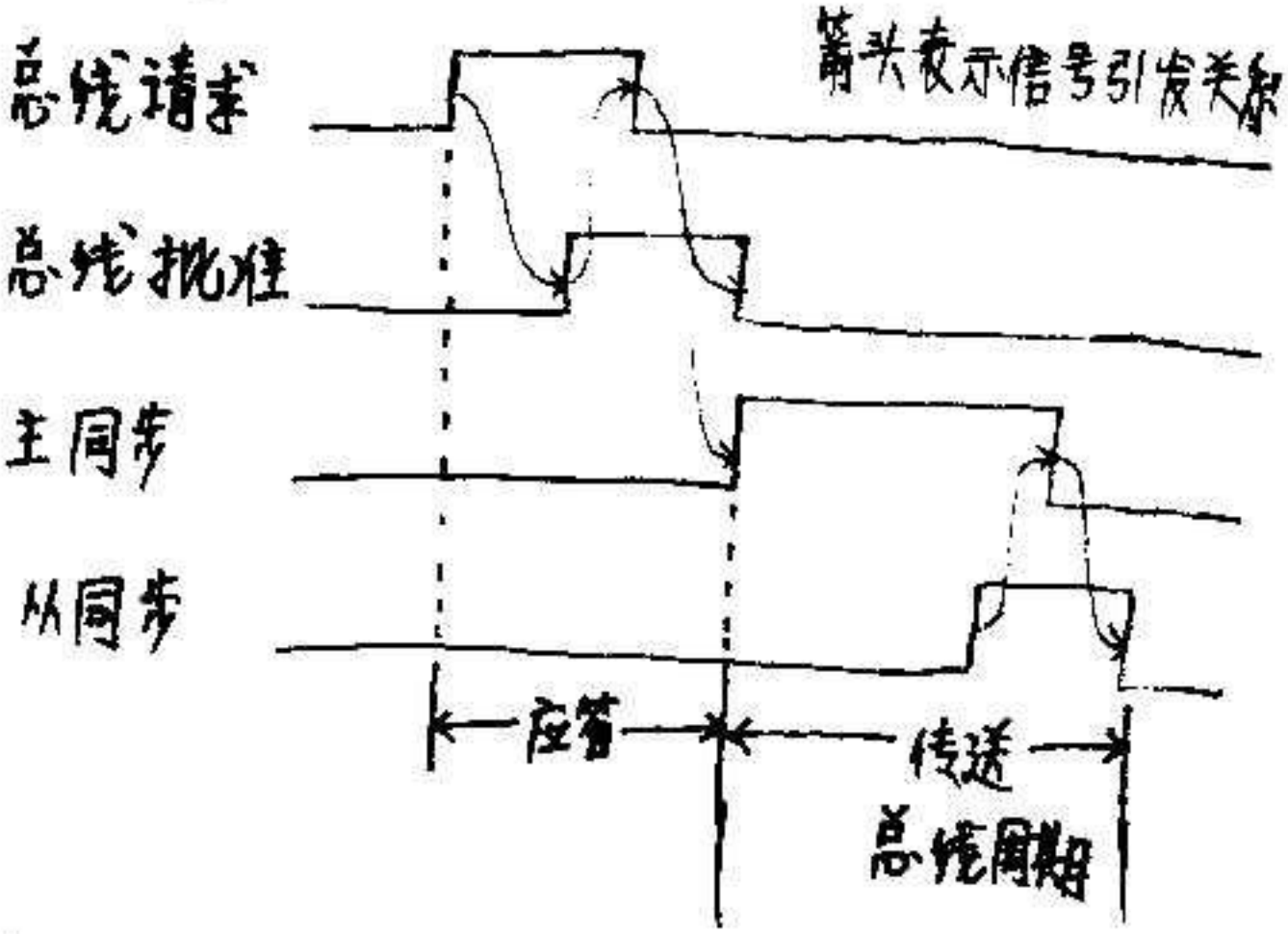


从时序控制方式看，如图所示过程是以时钟周期为单位，因而属于同步控制方式。时钟周期数可在一定程度上随需要而变，例如在传送过程中可引入或不引入 S_w，因而部分引入了异步控制的策略

4. 异步总线

①定义：总线传送条取应答方式实现，总线周期按传送的实际需要决定长短，需长则长，能短则短，其主要特征是不采用统一的时钟周期划分.

②时序举例.



全互锁

③概念.

A. 主设备：申请并控制总线的设备（主动的一方）

B. 从设备：响应主设备请求，进行传送的设备（被动的一方）

*. 不是按发送或接收来分.

② CRT 显示器： VRAM与屏幕显示之间的一组对应关系.
如：内容. 容量. 地址组织. 信息转换、同步计数器.
③ 磁盘、信息分布及寻址信息、
调用过程（DMA初始化、DMA传送、中断、速度指标）
④ 打印机，调用过程.
* 有关的重要概念.
述意，定义描述. 特点，应用、对比. 澄清一些似是而非的问题.

第一讲 CPU组织

一. 逻辑组成（模型机 [233]）[233]模型机数据通路框图!

1. 要求.
① 寄存器：
* 从功能上可分为三组：用于a.运算、b.控制、和c.主存（外部）接口
a. $R_0 \cdots R_n$. C. D
b. IR. PC. PSW
C. MAR. MBR. SP
* 从编程角度可分为二类：a.可编程的，b.透明的.
a. $R_0 \cdots R_n$. PC. PSW. SP
b. C. D. IR. MAR. MBR.
② ALU部件：→CPU内部数据通路结构简单：单向的寄存器，一组单向数据总线，以ALU为内部数据传送通路的中枢.
* 输入选择，选择器 或 锁存器 → 集成化寄存器组，一组双向数据总线.
* ALU
* 移位器
③ 总线与传送通路

*. CPU与其他设备通信，CPU总是主设备.

*. 主存与其他设备通信，主存总是从设备.

C. 选择同步方式还是异步方式.

各设备速度的差异程度：差异不大，便于统一规划，用同步；反之用异步（一般情况）

各设备操作时间的可确定性

传送距离远近：一般说，近距离用同步，远距离用异步.

时序信号的产生情况（单机系统或多机系统）

三. 程序中断方式

1. 定义及应用

当CPU接到某个随机性的中断请求，CPU暂停执行原程序，转去执行中断处理（服务）程序，为该随机事态服务，并在处理完毕后自动恢复原程序的执行.

① 实质：程序切换.

② 特点：随机性.

A.随机发生：如按键.

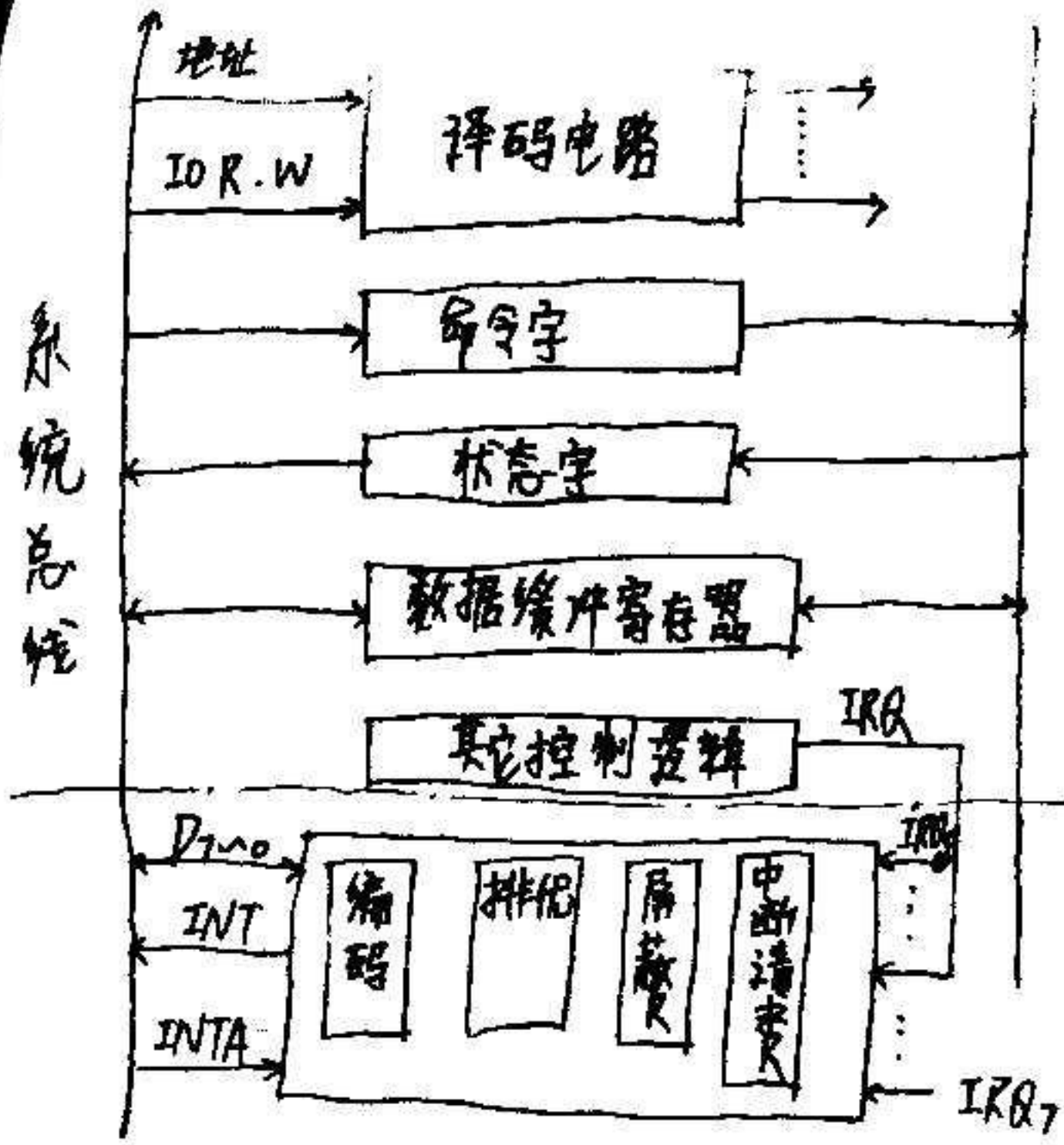
B.程序有意调用，但以随机请求方式进行处理，如打印机.

C.随机插入，软中断 INT n

③ 应用：控制中低速I/O操作 打印机

可处理突发的随机事态 } 能举例

2. 中断接口功能模型



见书 P354

虚线以上是一个外围接口

虚线以下是各设备公用的公共接口逻辑部分

① 地址译码

* 外设的编址方式 最灵活

与主存统一编址

I/O 端口编址 > 寄存器级

设端码、下属若干寄存器.

* 地址与 IOR.W 综合译码

② 命令字

③ 状态字

例1. 打印机接口 (P358)

a. 命令字: 允许中断请求 INTEN

选中输入信号 SLCTIN

初始化 $\overline{INIT}$

自动走纸 AUTOFDXT

数据选通 STROBE

b. 状态字：忙 $\overline{BUSY}$

确认 $\overline{ACK}$

纸完 PE

联机选择 SLCT

出错 $\overline{ERROR}$

例2：磁带机接口

A. 命令字

越过	数据块数	R/W	正/反	暂停	启动

b. 状态字

通路故障	校验错	数据迟到	忙	DMA请求	中断请求

例3：电梯接口

A. 命令字

允许中断请求	速度	上升/下降	运转/暂停	启动

00 匀速

01 加速

10 减速

b. 状态字

梯内请求

梯外请求

运转状态及故障

例4. 四个通信设备，共一个中断类型码

方案一：有四个接口，各占不同的I/O端口，但共用一个中断类型码，通过命令状态字判别

方案二：共一个接口。

④ 数据缓冲寄存器/存储器

⑤ 中断请求逻辑

⑥ 其他与设备特性有关的控制逻辑

⑦ 中断控制器

* 中断请求寄存器。

* 屏蔽技术

· 用在多重中断中（即允许中断嵌套）

· 动态调整优先级。

* 优先排队

快速设备优先于慢速设备；输入设备优先于输出设备。（一般情况）

* 编码

3. 中断服务程序入口地址的获得。

① 向量中断方式

将各个中断服务程序的入口地址组织成中断向量表，存于主存中，响应中断时，由硬件直接产生对应于中断源的向量地址，据此访问中断向量表，从中取出服务程序入口地址，由此转向服务程序。

* 中断向量：

一组中断服务程序入口地址（一组标量）

* 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（中断指针）

指向中断向量表的存储单元地址.

② 非向量中断方式

CPU响应各中断请求时，只产生唯一的地址，由此进入查询程序，通过软件查询确定中断源，转入相应的服务程序.

四. DMA方式

与程序中断方式相比，DMA方式仅需占用系统总线，不切换程序，因此不存在保存断点、保护现场、恢复现场、恢复断点等才操作.

1. 定义及应用

DMA方式一般应用于主存与高速 I/O 设备间的简单数据传送.

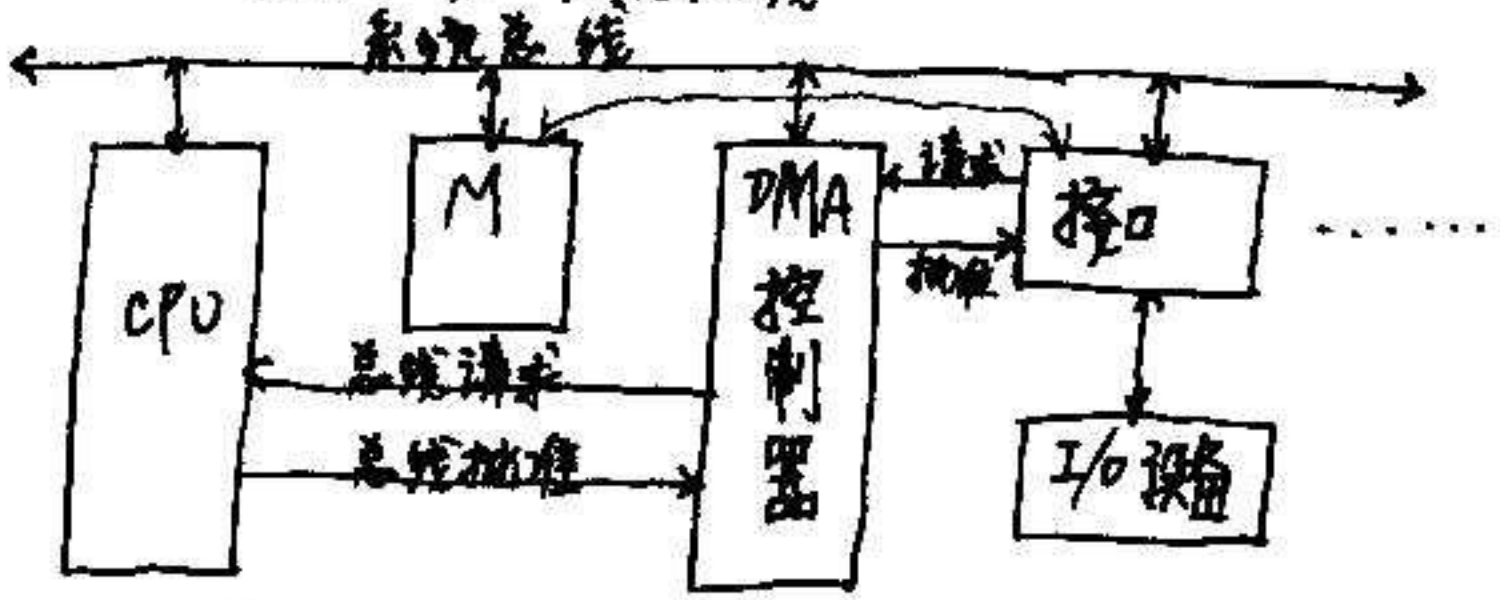
直接依靠硬件在主存与 I/O 设备间进行直接的数据传送，在传送期间不需CPU程序干预

*直传 主存 ⟷ I/O

*直接依靠硬件

*CPU不干预 只在传送期间

2. DMA控制器与接口 系统总线



① DMA控制器

A. 功能.

接受设备（接口）的DMA请求，向CPU申请系统总线

获得CPU批准后，向设备发批准信号

控制总线实现 DMA传送.

b. 组成

命令字（DMA控制器工作方式）

状态字、屏蔽字、

主存缓冲区首址，交换字数

② 接口

a. 功能

连接系统总线与I/O设备，实现数据缓冲与传送、　　反映设备特性

b. 组成.

命令字

状态字

数据缓冲存储器

DMA请求逻辑

中断逻辑

与设备有关部件

③ 传送路径

+ 请求、批准

+ 数据.

I/O设备 —— 接口 —— 总线 —— M

④ 单字传送 P363

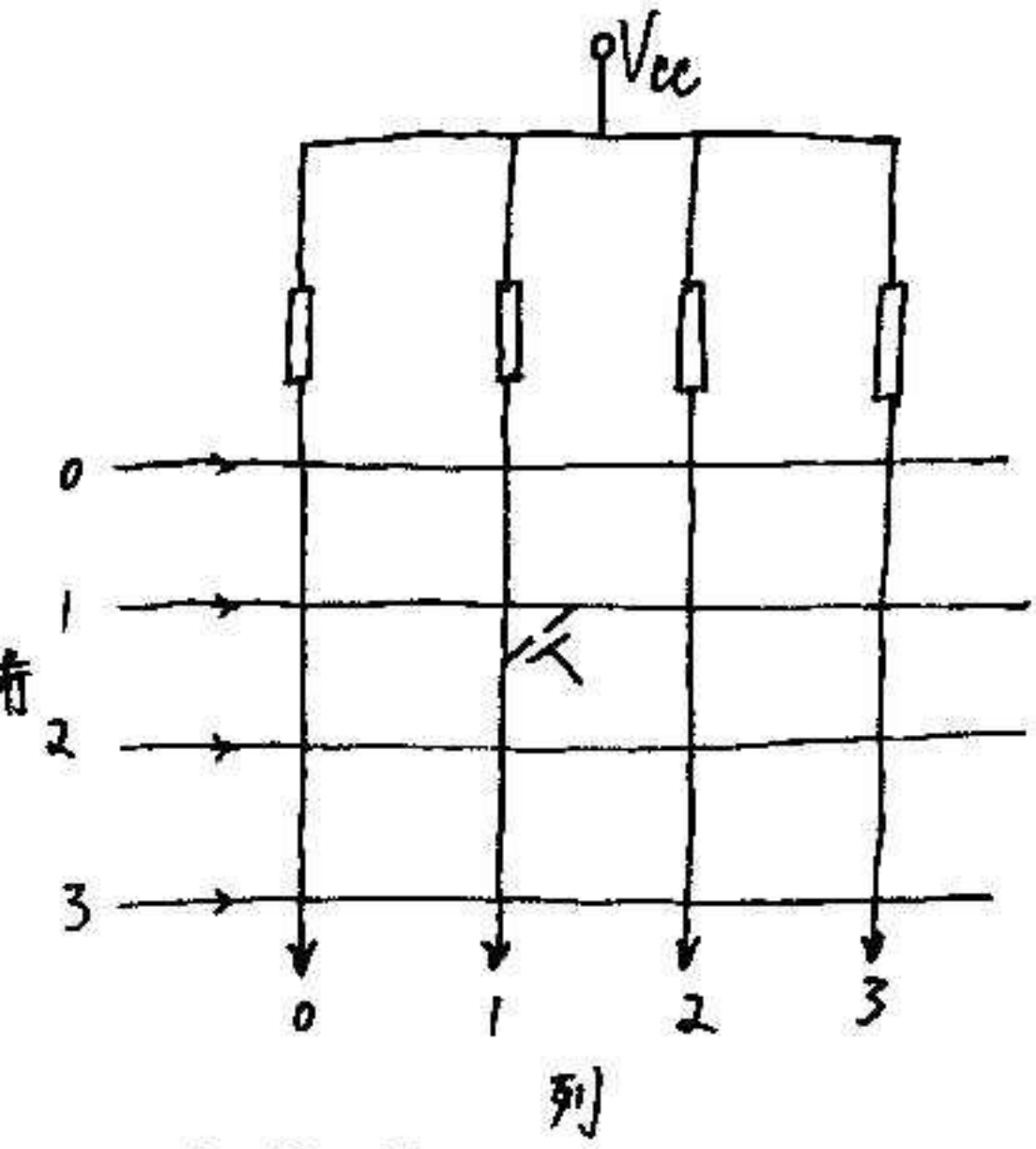
成组传送 P364

第四讲　主要I/O设备原理

一. 键盘 P291—292

* 将诸键 组成矩阵
* 用硬件扫描或软件扫描 判定按键位置, 即扫描码
* 查表, 将扫描码 → 键码 (ASCII)

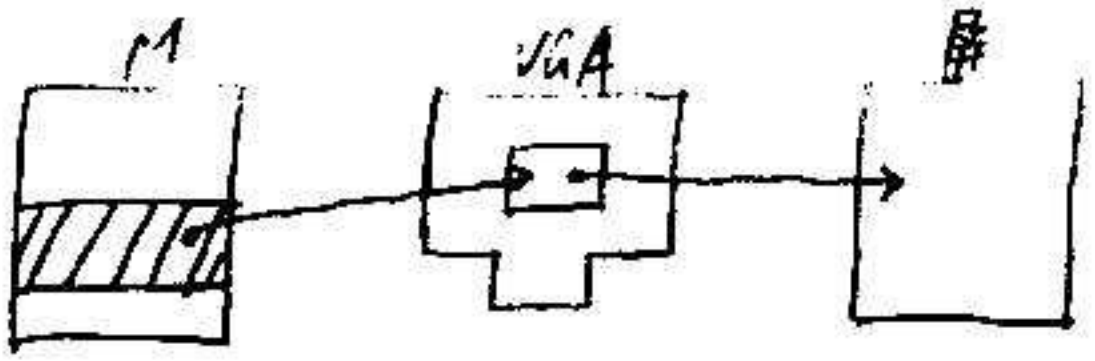
例: 软件 逐行 扫描键盘



① 按键产生键盘中断请求
② CPU (主CPU 或专用单片机) 执行扫描程序
③ CPU 输出 送往行线, 逐行为0 地扫描 (某行为0, 其他行为1), 列线输出送CPU, 判别按键行、列位置. 若第i行为0时, 第j列也为0, 则按键位置为 (i, j)
④ 查表, 将按为按键编码

二. CRT显示器

* 显示方式: ①字符/数字 方式 (A/N方式) (文本显示方式)
  ②图形方式 (APA方式)
* 发光原理: ①发光器件, 如CRT. LED (阴极射线管CRT, 发光二极管LED, 等离子显示器件PDP. 场致发光板ELD)
  ②光调制器件, 如液晶显示LCD. 电化学效应显示
* 分辨率: n行 x m列. 如 25×80 → 对于字符方式
  n点 x m线. 如 1024×768 → 对于图形方式
* VRAM与屏幕 显示之间 的对应关系



1. VRAM 内容

① 显示内容

A. 字符方式： 字符编码

b. 图形方式： 象点(元，素)代码

② 显示属性

③ 彩色

2. 信息转换

① 字符方式

字符代码(高位地址)
时序信号(低位地址) > ROM(字符发生器) → 象点代码 → 并/串 → Z调制端

② 图形方式：象点代码 → 并/串 → Z调制端

3. 不考虑属性，彩色的基本缓存容量

① 字符方式： 例 80×25 字符显示的缓存中存放字符的ASCII码

$80 \times 25\ B = 2000\ B \doteq 2KB$ 一个字符编码占缓存的一个字节

② 图形方式： 例 800×600

$$\frac{800 \times 600}{8} = 60000\ B \doteq 64KB$$

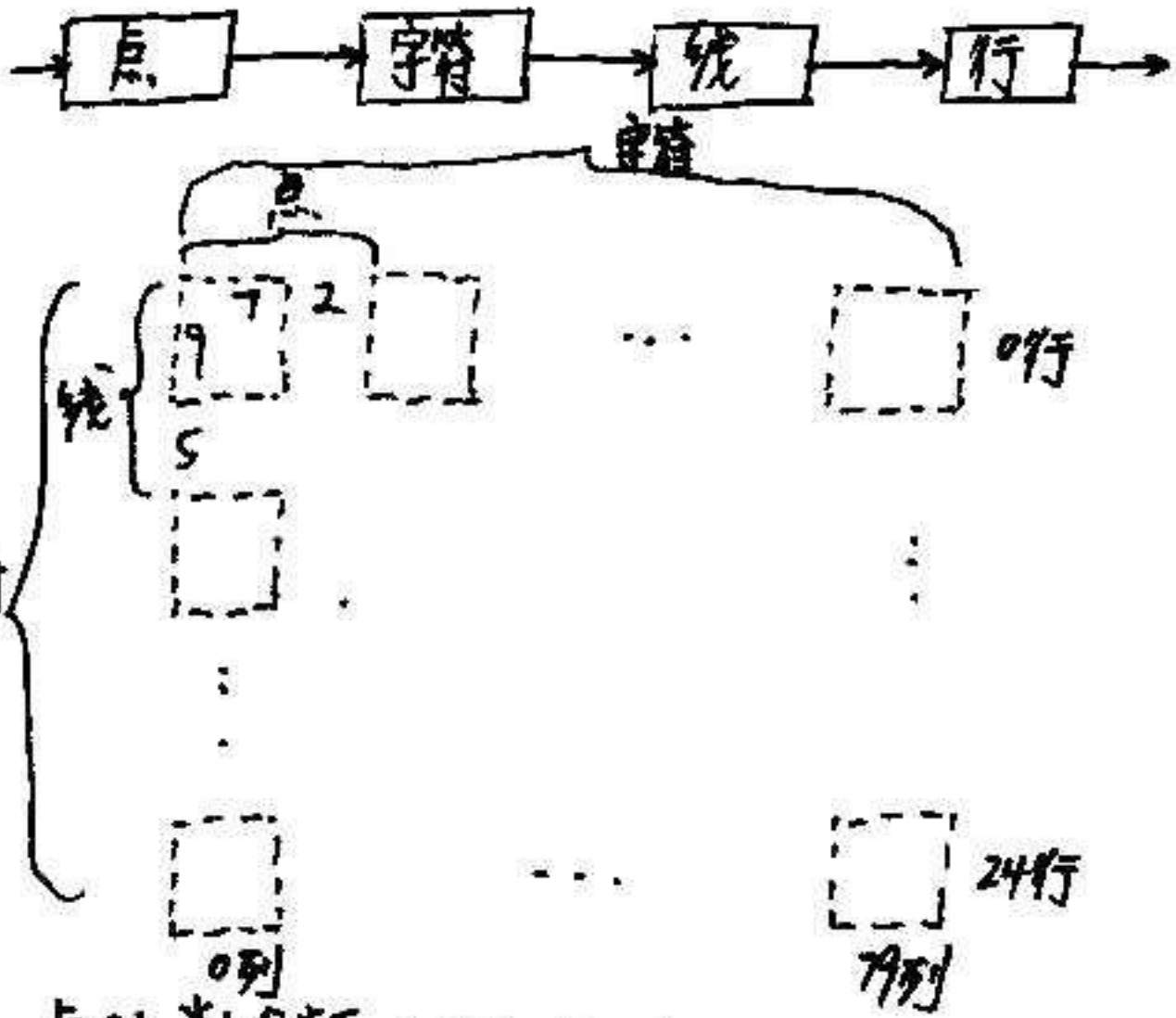
4. 地址组织

由显示位置(行列号)产生 VRAM 地址，扫描顺序自左向右，自上而下，地址相应递增(左上角显示字符编码存放在 VRAM 第一个单元)

5. 同步控制

①字符方式

例：分辨率 80列×25行，字符点阵 7横×9纵　字符区间 9横×14纵



点计数分频 (7+2)=1

字符计数分频 (80+L)=1

线计数分频 (9+5)=1

行计数分频 (25+m)=1

* 何时访问一次 VRAM？(L.M与显示技术有关)

一次点计数循环

* VRAM 地址？

行计数值与字符计数值 转换为 VRAM 地址.

字符计数器提供的当前显示位置列号，可以作为产生缓存低位地址的依据；

行计数值决定了字符行号，可以作为产生缓存高位地址的依据.

* 何时访问字符发生器？

每次访问 VRAM 之后，

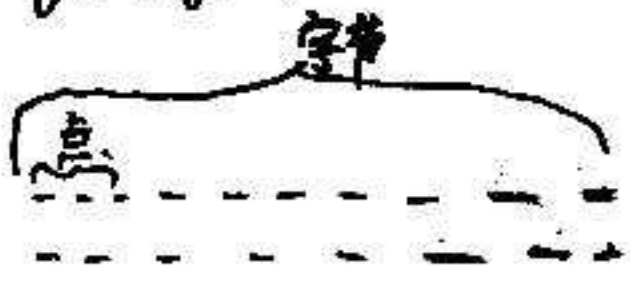
* 字符发生器 地址？

VRAM 输出 (字符，编码) 作为高位地址. 线计数值为低位地址.

*何时发水平同步？
-次字符计数循环
*何时发垂直同步？
-次行计数循环.
* 逐行扫描： -帧一场 场频=帧频 扫描一遍称为一场
隔行扫描， -帧两场 场频=2·帧频

② 图形方式

例：800×600.



线

- - - - - - - -

点计数： 8:1
字节计数： $(\frac{800}{8}+1)=1$
线 计数 $(600+m)=1$

三. 磁盘

磁盘读写采用DMA方式进行数据传送，而对寻道是否正确的判别处理、批量传送结束后的善后处理，则采用程序中断方式

1. 信息所布与寻址信息.

① 驱动器号(台号)，② 圆柱号(道号)，③ 磁头号(面号)，④ 扇区号/数据块号，⑤ 数据缓冲区

2. 调用过程.

① 送驱动器号给道配器，调回状态字，判可否调用.
② 送圆柱号，磁头号给道配器，启动寻道.
③ 寻道结束，提出中断请求(向道配器)；判寻道是否正确.

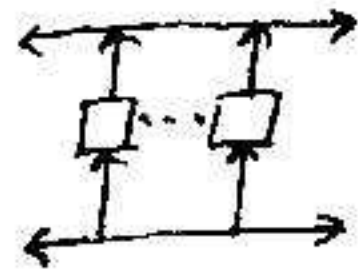
㊙与系统总线的连接.

2. 模型机结构特点. 由ALU汇集(选择),单向内总线采用数据分配,寄存器在逻辑上独立.

独寄存器. ALU汇集. 单向ALU总线.

3. 可能的变化:

集成化寄存器组(单口、双口). 双向ALU总线. 多组总线



▲注: M → MBR → IR (老)

M → IR (新) M → IR, PC+1 >

指令流程.

* 做何操作
* 源, 与目的 ADD (R0), X(R1)
源 目的
* 寻址方式的实现 关键所在! [P230]

例1: MOV (R0), -(SP); M → IR, PC+1 → PC 可同时进行. 因为它们各自使用的数据通路没有冲突, 读取指令经由数据总线, 而PC+1经由ALU与内部总线

将数据压入堆栈

M → IR, PC+1 → PC > 取指

R0 → MAR, M → MBR → C > 取源操作数

SP-1 → SP, MAR 取目的地址

C → MBR, MBR → M > 压入堆栈

PC → MAR 送回继指令地址

④ 送起始扇区号、交换扇区数给适配器
送主存缓冲区首址，交换字数给 DMA 控制器
⑤ 发读写命令给适配器与 DMA 控制器，启动 R/W
⑥ 找到起始扇区，连续读/写。（适配器——驱动器）
⑦ 以 DMA 方式 插占系统总线 进行 适配器与主存间的数据直传
a. 读盘，当适配器缓存装满一个扇区，提出 DMA 请求。
b. 写盘，当适配器缓存有一个扇区空，提出 DMA 请求。
C. 每进行一次 DMA 传送，DMA 控制器相应修改主存地址与交换量
⑧ 批量传送结束，向主CPU 提出中断请求，调回状态字，进行结束处理。

过程：
* DMA 初始化。
* DMA 请求的提出、传送、批准、传送。
* 中断。

3. 磁盘的速度指标 (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（b/s，波特率）

四. 打印机
1. 初始化，上电初始化。
主机命令初始化。
2. 打印机完成初始化（准备好），提出中断请求。
主机执行中断处理程序，将打印字符编码经适配器送入打印机字符缓冲区，每收到一个字符，发出一个确认信号 $\overline{ACK}$。

3. 方案之一：每送一个字符即输入打印，打印完后再提出中断请求.

4. 方案二：每次送一行字符，再转入打印

①若收到的是控制字符，则转入相应的功能码处理程序.

②若收到的是打印字符，则继续接收代码，直至一行送满或一行送完，自动转入打印

5. 打印完一个字符/行，再提出中断请求，继续传送，打印直至打印完毕.

② 微指令

AI	BI	SM	Co	S	ZO	EMAR	R	W	ST	SC

AI. BI: ALU输入选择

SM. ALU功能选择

Co. 进位选择

S. 移位功能选择

ZO. 内总线输出控制

EMAR, R, W: 访存控制

ST: 辅助操作控制

SC: 顺序控制

③ ADD (SP)+, X(R0):

M → IR
PC+1 → PC
} 取指

SP → MAR
M → MBR → C
SP+1 → SP
} 取源地址、源操作数

PC → MAR
M → MBR → D
PC+1 → PC
} 取形式地址

R0 + D → MAR
} 变址计算

M → MBR → D
} 取数

C + D → MBR
} 运算

MBR → M
} 存放结果

PC → MAR
} 送后继指令地址

取目的操作数

2. 用一台PC机控制牛台加热炉，为此设置一个共用的中断接口，占用一个中断源，CPU能分别向牛台加热炉发出命令：关闭、加热，实时采集加热炉温度值，以调节控制。加热炉能分别向CPU提供状态信息：工作、空闲、故障、温度正常、温度越限。

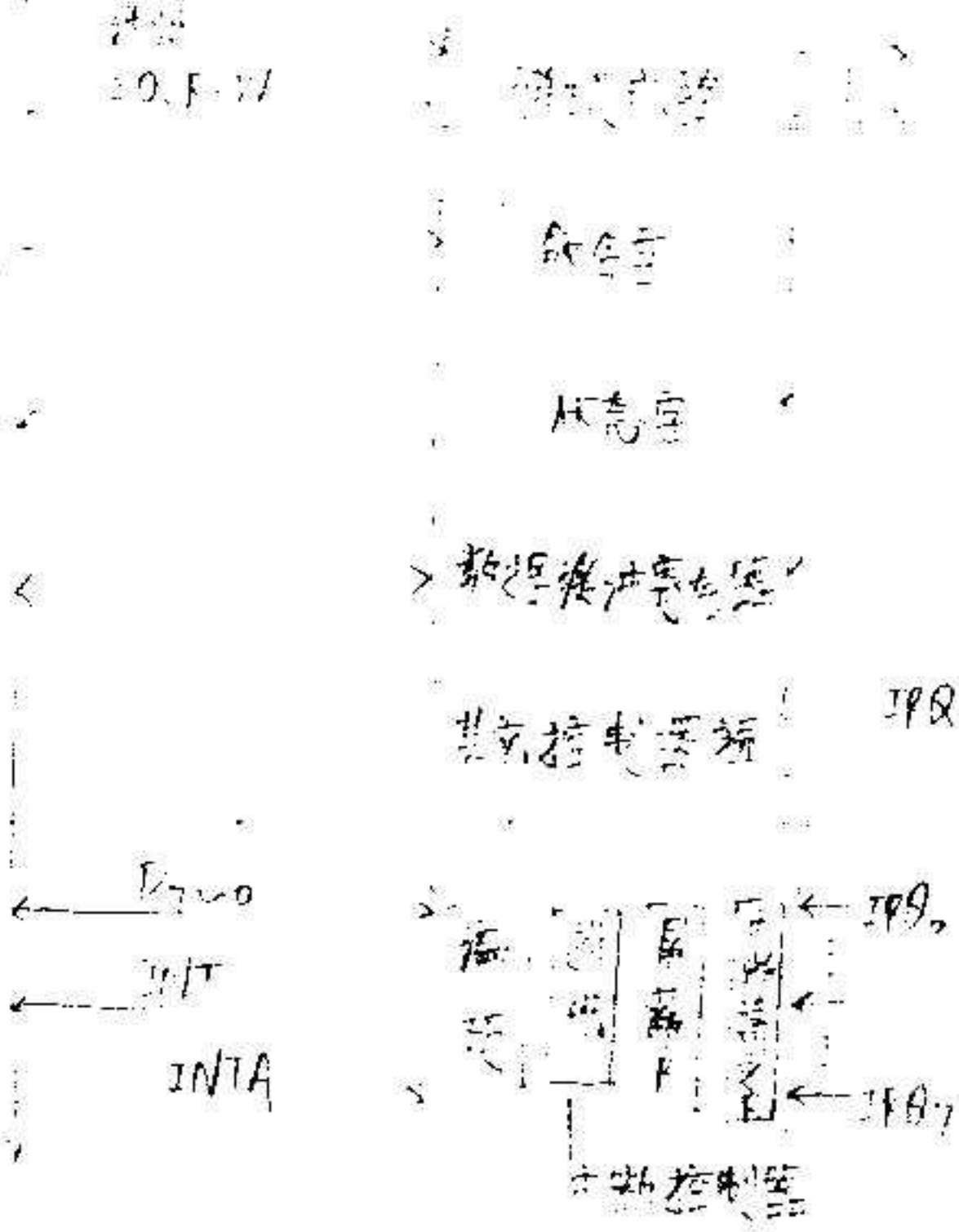
① 画出中断接口的功能模型（寄存器级粗框图）

② 简述上图中各部件的作用。

③ 拟定出命令字与状态字的格式

④ 所设计的接口如何实现中断排优与中断屏蔽。

①



④ 译码电路：选择接口有关寄存器

命令字：向设备发命令

状态字：反映外设及接口状态

数据缓冲寄存器：采集温度值，实现缓冲，过型速度匹配

其它控制逻辑，产生中断请求

中断控制器：汇集中断请求信号，经过屏蔽控制，优先排队，形成送往CPU的中断请求信号INT，在接到CPU批准信号INTA后，通过数据总线送出向量地址。

⑤ 命令字

加热/关	加热/关	加热/关	加热/关
3	2	1	0

1——加热
0——关闭

状态字

…	…	…	工作	空闲	故障	正常/越限
3	2	1				0

⑥ 在中断控制器中设置中断请求寄存器，寄存各请求信号
在中断控制器中设置中断排优逻辑，决定优先者
在中断控制器中设置屏蔽寄存器，决定是否屏蔽

3. 请按下列各题的命题，更正其错误或补充其不足。

① 原码乘法是指：用原码表示操作数，然后直接相乘。

更正：原码乘法是指：取绝对值相乘，符号位按"同号为正，异号为负"单独处理

② 随机存取方式是指可在任何时间随意地进行读出或写入

更正：随机存取方式是指：a. 可按地址直接指向任一单元 (P8)
b. 存取时间与地址无关。

③ 同步控制方式的基本特征是：所有指令的执行时间相同.

更正。同步控制方式的基本特征是：有严格的时钟周期划分.

④ 在异步控制方式中，主设备是指发送信息的一方.

更正，在异步控制方式中，主设备是申请、控制系统总线的一方.

⑤ 串行接口是指，接口与总线之间采取串行传送，接口与设备之间也采取串行传送.

更正。串行接口与系统总线之间一般采取并行传送，接口与设备之间采取串行传送.

4. 何谓微程序控制方式?

① 将微命令以微代码形式编成微指令，存入控制存储器(ROM)之中，

—— 由存储器提供微命令.

② 一条微指令控制一步操作，一段微程序(若干条微指令)解释执行一条机器指令

—— 以程序方式进行设计，以执行微指令方式实现机器指令功能

5. 何谓 DMA 方式? 应用于何种场合?

DMA方式 指直接依靠硬件在主存与I/O设备间进行直接的数据传送，在传送期间不需CPU的程序干预

应用于 主存与高速I/O设备间的简单数据传送

6. 何谓向量中断? 举例说明其获得服务程序入口地址的方法.

向量中断是指 采用一种中断响应方式：将各个中断服务程序的入口地址组织成中断向量表，存于主存中，响应中断时，由硬件直接产生对应于中断源的向量地址，据此访问中断向量表，从中取出服务程序入口地址，由此转向服务程序.

举例，在IBM PC微机系统中，中断向量表存放在主存的0～1023(十进制)单元中.每个中断源占用四个字节单元，存放其服务程序入口地址，其中两个字节存放其段地址，两个字节存放偏移量. 因此整个中断向量表可以驱动256个中断

源，与中断类型码 0~255 相对应。

当CPU响应中断请求时，向8259A中断控制器送去批准信号INTA；并由数据总线从8259A取回被批准请求源的中断类型码，乘以4，形成向量地址；访问主存，从中断向量表中读得服务程序入口地址；然后转向服务程序

7. 系统总线一般分为：地址、数据、控制三组，简要说明控制信号一般包含哪些？

① 时序控制信号：同步定时信号（时钟，状态）
异步应答信号

② 数据传送控制：I0/M，R/W，字节/字

③ 中断请求与批准

④ 总线请求与批准（含DMA）

⑤ 优先排队 入/出

⑥ 复位

8. 主机与外围设备间的信息传送控制方式一般有哪几种？分别适用于什么场合

① 直接程序传送方式。

适用场合：CPU速度不是很高，效率问题不是很重要，因而允许在I/O操作中CPU不干别的事

CPU的工作方式仅其在I/O过程中无别事可干，因而只能处于等待状态。在调试、诊断过程中，有意识让CPU工作方式尽量简单，以便集中考察其I/O操作的正确与否等。

② 程序中断方式

适用场合：处理外围设备的中、低速I/O操作与随机请求。这些随机请求可以是复杂的随机事态。

③直接存储器存取(DMA)方式

适用场合：主存与高速I/O设备间的简单数据传送.

9. 比较同步总线与异步总线的异同，常用PC总线属于哪种控制方式？

同步总线：

①定义：总线传送操作由统一的时序信号同步，其主要特征是有严格的时钟周期划分

②时序举例：如CPU访存，总线周期一般固定为几个CPU时钟周期

③优点，控制比较简单，较易实现

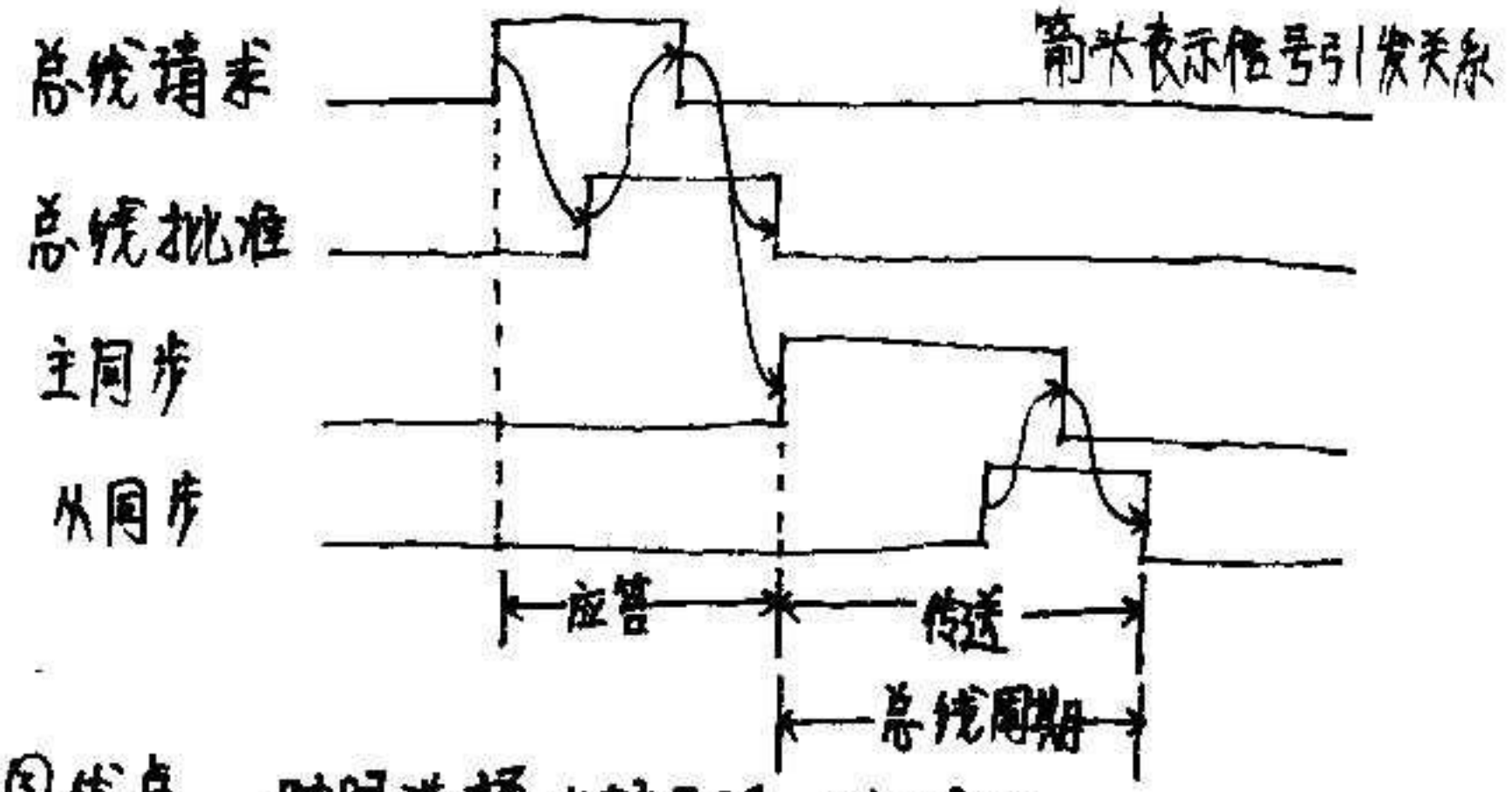
④缺点，时间利用上不够灵活，一些本可快速实现的操作也需占用固定的时钟周期

⑤应用场合：传送距离较短、系统内各设备差异不很大的系统中

异步总线.

①定义：总线传送采取应答方式实现，总线周期按传送的实际需要决定长短，需长则长，能短则短，其主要特征是不采用统一的时钟周期划分

②时序举例.



③优点：时间选择比较灵活，利用率高

④缺点：控制比较复杂

⑤应用场合：传送距离较长、系统内各设备差异较大的系统内

PC总线采用同步扩展方式，即属于同步控制方式，但引入异步控制思想。
以同步总线为基础，实现"异步事件同步化"。如应答时间为时钟周期的整数倍。
又如总线周期为时钟周期的整数倍，时钟周期数可变。

10. 举例说明三种中断屏蔽技术的应用

① 用在多重中断方式中（即允许中断嵌套）。

② 利用屏蔽技术动态调整优先级

11. 描述打印机接口的中断过程，需说明：

① 何时提出中断请求？

② 请求信号如何转换为服务程序的入口？

③ 在单级中断方式下，打印机中断服务程序主要完成哪些工作？

答：① 打印机准备好或打印完一行时，申请中断

② 请求信号送8259，经屏蔽、判优，产生公共请求INTA送CPU

CPU响应，发出批准信号INTA，并关中断，保存断点，从8259取回打印机中断类型码，转换为向量地址，查向量表，转打印机中断程序入口。

③ 打印机中断程序保存有关寄存器内容，向接口缓冲器传送数据，恢复现场，开中断，返回。

12. 针对下列各小题的题意，改正其结论中的错误，或补充其不足。

(1) 在补码除法中，够减商零，不够减商1。

更正：在补码除法中，异数和除数同号商1，异号商0。

② 微指令周期是指从主存中读取并执行一条机器指令所用的时间。

更正：微指令周期是指从控存中读取并执行一条微指令所用的时间。

③ 压栈操作是指：将有关信息写入堆栈指针SP。

更正：压栈操作是指：将有关信息写入SP所指示的栈顶单元。

讨论：

① 取指部分不能简化

② PC→MAR 也可放在第一步

PC→MAR
M→IR
PC+1→PC
> 取指

③ -(R) 与 -(SP) 自减型寄存器间址 "先减后操作" SP→MAR (X)
SP-1→SP

④ 如何分步？ a. 通路有无冲突 如 M→IR 问题
b. 时间上是否允许 如 C→M 问题 C→MBR→M (X)

C 打入→ MBR M 置入→ MBR 打入→ IR

例2. ADD X(R0), (R1)+

M→IR
PC+1→PC
> 取指

PC→MAR
M→MBR→C
PC+1→PC
> 取形式地址（源操作数在指令后）

C+R0→MAR 变址计算
M→MBR→C 取数
} 取源操作数

R1→MAR
M→MBR→D
> 取目的数

R1+1→R1 修改指针
} 取目的数

C+D→MBR 运算

MBR→M 存放结果

PC→MAR 适后继指令地址

讨论：

(4) 主存和外部设备交换数据时，主存既可作为主设备，也可以作为从设备。

更正：主存和外部设备交换数据时，通常作为从设备。

(5) 多重中断方式是指：CPU同时处理多个中断请求。

更正：多重中断方式是指：CPU在执行中断服务程序的过程中，允许响应更高级别的中断请求。

(6) 若采用串行进位方式，则 $C_4 = g_4 + P_4 g_3$

更正：若采用串行进位方式，则 $C_4 = g_4 + P_4 C_3$

$C_3 = g_3 + P_3 C_2$

$C_2 = g_2 + P_2 C_1$

$C_1 = g_1 + P_1 C_0$

(7) 每访问一次CRT字符显示器的缓冲存储器，取得一个字符的全部点阵代码。

更正：每访问一次CRT字符显示器的缓冲存储器，取得一个字符的编码。

(8) 向量地址是中断服务程序的入口地址。

更正：向量地址是访问中断向量表的地址。

5. 简要回答下列问题。

(1) 试比较对阶和右规的异同点。

相同点：阶码增大 尾数右移

不同点：(1) 对阶使小阶向大阶对齐，右规使尾数[入]

(2) 对阶后，原小阶的尾数不再是规格化形式 右规后尾数是规格化形式。

(3) 对阶不会使阶码上溢，右规可能引起阶码上溢。

(2) 简述微程序控制的基本思想。

(1) 将微命令以代码形式编写在微指令中，一条微指令控制一步操作。

(2) 若干条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。

(3) 中断屏蔽技术常用于：

(1) 动态修改优先级别，例如屏蔽优先级较高的请求，使优先级较低的请求获得响应

(2) 实现中断嵌套，例如多重中断中，屏蔽与现行优先级相同或较低级的中断请求，开放更高级的中断请求。

(4) 在系统总线中，什么情况下适宜采用同步控制方式？什么情况下适宜采用异步控制方式？

当总线上连接的各设备速度差异不大，传送时间确定，传送距离较近时，宜采用同步控制方式。

当总线上连接的各设备速度差异大，传送时间不确定，传送距离较远时，宜采用异步控制方式。

(5) 动态存储器为什么要进行刷新？采用哪种刷新方式可以既不影响CPU访存，又不影响存取周期？如何安排？

(1) 动态存储器依靠电容存储电荷，时间一长，电荷会泄露，需定期向电容补充电荷。

(2) 采用异步刷新方式。

(3) 将各行刷新分散安排在2ms间隔内，定期提出刷新请求，在CPU不访存时刷新。

(6) 某CRT显示器按图形方式工作，分辨率为1024×768线。若要显示四种颜色，应如何设置各级计数器的分频关系？

总计数器： ÷1 分频

宽计数器： ÷1024 分频

线计数器： ÷768 分频

(7) DMA方式的三个阶段各采用什么手段，完成哪些操作？

(1) 初始化阶段：用程序传送初始化信息，包括传送方向、主存首址、*交换量、外设寻址信息。

(2) 传送阶段：由DMA控制器接管总线权，控制传送。

(3) 结束阶段：用中断程序作善后处理。

6 [illegible] 显示器和打印机[illegible]等处理器工作上的异同点。

相同点：将字符编码转换为字符点阵代码。

不同点：显示器中，字符发生器按行组织点阵代码，取出一行代码后，转换为串行代码再送显示头。

打印机中，字符发生器按列组织点阵代码，取出一列代码后直接送打印头。

9 磁盘有哪两种容量指标？它们在计算上有何区别？

<1> 非格式化容量，由最大位密度计算。

<2> 格式化容量，由扇区内的有效字节数计算。

① 变址寻址方式的实现

X 仅是一个助记符　　X是变址的一种习惯标注符号

常见错误：

* X+R₀ → MAR

* R₀ → MAR
　M → C
　C + Rᵢ → MAR
　或：C + X → MAR

② (Rᵢ)+　先操作后修改
　若是 (SP)+ ，堆栈弹出。

例3. NEG (R₀)

M → IR
PC+1 → PC　> 取指

R₀ → MAR
M → MBR → D　> 取目的数

D+1 → MBR　求补运算

MBR → M　存放结果

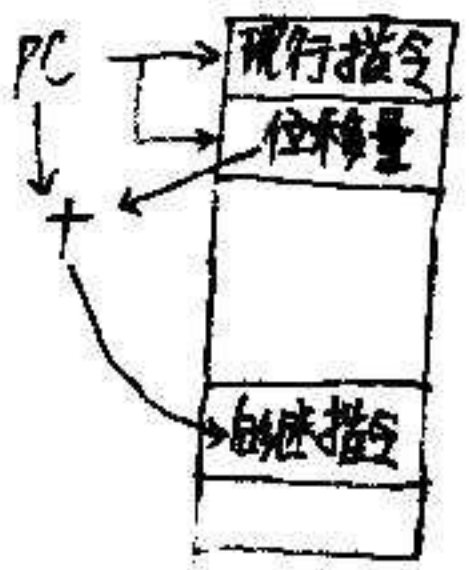
PC → MAR　获后继指令地址

例4. JMP X(PC) 相对寻址 [P130]

M → IR
PC+1 → PC　> 取指

PC → MAR
M → MBR → C　> 取形式地址（位移量）

PC+C → MAR、PC　变址计算，产生转移地址



例5. JSR (R0) 转子 转子指令，保存返回地址，子程序入口→PC

M→IR
PC+1→PC
> 取指

R0→MAR
M→MBR→C
> 取转移地址（子程序入口）

SP-1→SP, MAR 指向新栈顶
PC→MBR
MBR→M
> 压栈
} 保存返回地址

C→PC. MAR 转子子程序

三. 微命令序列

1. 两种微命令：电位型（维持一个时钟周期宽度）. 脉冲型（短暂的、起作用）

2. 围绕数据传送或时序切换的需要来拟定微命令

① CPU内数据传送，ALU输入选择.
ALU功能选择.
移位功能选择
打入脉冲
SMBR (SIR)

② 访存：EMAR. R/W 先做：VMA (MREQ)

③ 时序切换.

如何结束本工作周期与时钟周期！
如何进入新工作周期与时钟周期！

一个工作周期可包含若干节拍，根据不同指令的需要，节拍数可变。为此设置一个时钟周期的计数器T，其计数循环可更需要而改。若本工作周期应当结束，则发命令T=0，计数器T复位，从T=0开始一个新的计数循环，进入新的工作周期。若本工作周期并不需要结束，则发命令T+1，计数器T将继续计数，出现新的时钟周期。

在每个时钟周期末尾发一个工作脉冲P，其前沿作为打入寄存器的定时，它标志着一次数据通路操作的完成。P的后沿作为周期切换的定时，在此刻对时钟周期计数器T计数、打入新的工作周期状态。

例1. MOV (R0), -(SP)

FT0 EMAR, R, SIR
PC→A A加1, 直传 CPPC
(1→FT=0) CPFT($\overline{P}$)
(T+1=0) CPT($\overline{P}$)

P
执行

打入新表示可省略

	1→ST	CPST(P̄)			〈新版本〉
FT。	EMAR. R, SMBR				
	MBR→B,	输出B,	直传	CPIR	
	T+1			CPT(P̄)	
FT,	PC→A	A加1	直传	CPPC	
	1→ST	CPFT(P̄)	CPT(P̄)	CPST(P̄)	〈老版本〉
ST。	R。→A,	输出A,	直传	CPMAR	
	T+1			CPT(P̄)	
ST,	EMAR. R, SMBR				
	MBR→B.	输出B.	直传	CPC	
	1→DT	CPST(P̄)	CPT(P̄)	CPDT(P̄)	
DT。	SP→A,	A-1,	直传	CPSP, CPMAR	
	1→ET	CPDT(P̄)	CPT(P̄)	CPET(P̄)	
ET。	C→A.	输出A.	直传	CPMBR	
	T+1			CPT(P̄)	
ET,	EMAR. W写	PC→A	输出A.	直传 CPMAR	
	1→FT	CPET(P̄)	CPT(P̄)	CPFT(P̄)	

讨论：

① 取指的两种方案，对应于两种版本。
② 访存
③ CPU 内部传送
④ 时序切换

对照寄存器级的指令流程。

MOV (R₀), -(SP);

M→IR
PC+1→PC　> 取指
R₀→MAR
M→MBR→C　> 取源操作数
SP-1→SP, MAR　取目的地址
C→MBR
MBR→M　> 压入堆栈
PC→MAR　送后继指令地址

四. 微命令的产生方法

1. 组合逻辑控制器（硬连逻辑）

① 基本思想：

根据产生的微命令的逻辑条件（指令代码，状态信息）与时间条件（工作周期，节拍、脉冲），由组合逻辑电路产生微命令。

② 优点：速度快

缺点：繁琐、混乱，设计效率低，不易修改，扩充，不易检查调试

③ 应用：是产生微命令的基本方法。

应用于RISC、巨型机等高速计算机。

2. 微程序控制器[ISI]

① 基本思想：

a. 将存储逻辑引入CPU

b. 将程序技术引入CPU的构成层，利用程序技术来编排指令的解释与执行

a. 将微命令以微代码形式编成微指令，存入控制存储器（ROM）之中，

—— 由存储器提供微命令。

b. 一条微指令控制一步操作，一段微程序（若干条微指令）解释执行一条机器指令 —— 以程序方式进行设计，以执行微指令方式实现机器指令功能

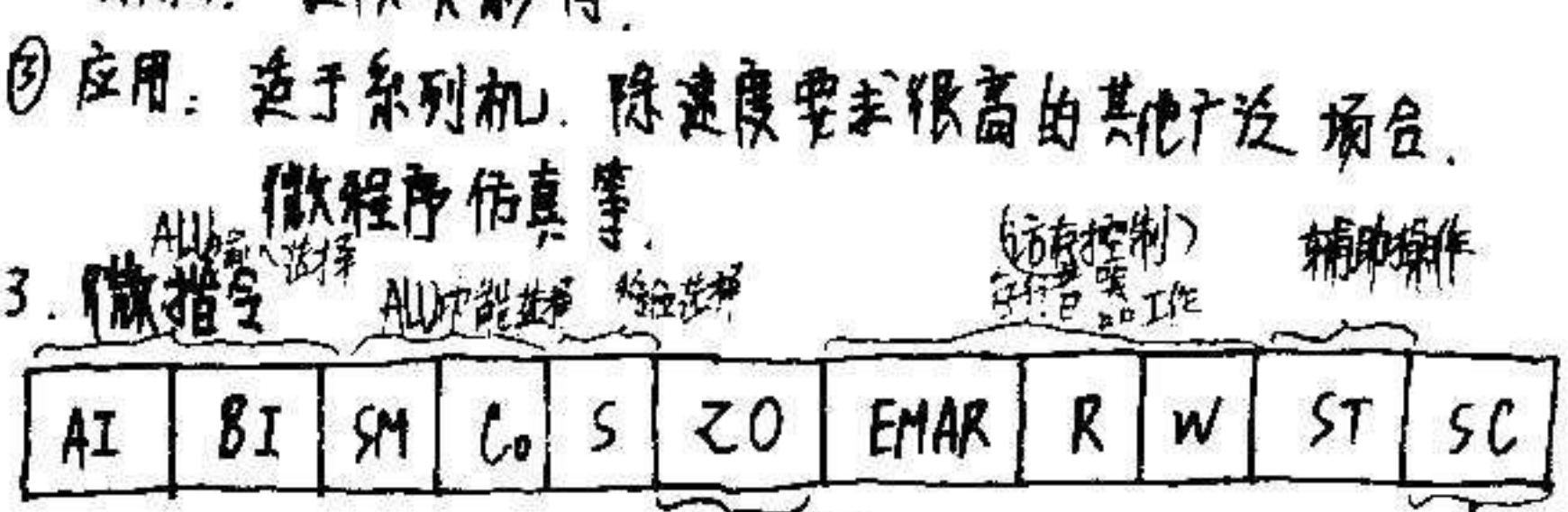
② 优点：机型，设计效率高，易修改，扩充，易于检查

缺点：速度受影响。

③ 应用：适于系列机，除速度要求很高的其他广泛场合。

微程序仿真等。

3. 微指令



① 微操作控制字段

A. 划分方法：按操作类型分段（同类、互斥）

例：CPU内部传送控制。

基本的分段原则：将同类操作中互斥的微命令归为一组

微指令分段的另一原则：微指令的每个小字段表示同一类型的操作

访存
辅助操作

b. 编码方法

直接控制法（不译法）

自投编译法（直接、间接）

自投直接编译法（显式编码、单重定义）
自投间接编译法（隐式编码、多重定义）

其它技巧：<1>微指令译码与机器指令译码的复合控制 <2>微地址与解释微指令代码

c. 分类： 水平型、垂直型、混合型

② 微程序顺序控制字段SC.

a. 分类：增量方式（顺序执行—转移方式）

断定方式（非固定加例试转移分量）

b. 具体类型.

顺序执行（加1）

转移 （全地址）

根据指令代码转移 { 操作码
源寻址方式
目的寻址方式
其他

转微子程序与返回

五. 同步控制方式 (P221)

1. 定义：各项操作与统一的时序信号同步，其主要特征是有严格的时钟周期划分.

*. 时间段的划分与安排.（由电位命令同步）. 操作流程.

*. 定时时刻.（由脉冲命令定时）.

*. 传送的协调方式.

2. 应用： CPU内部， 设备内部.
系统总线