

计算机组成与设计 复习手册

注：全在里面了,不会超，没有特难的题

第一大题：计算题：4 题

二：简答题：28 分

四道大题：

存储器扩展，指令执行流程<—这两题有
分析设计，

一， 概论：

1. 冯诺一帆

2. 计算机硬件主要基础图样

四 .

1. 存储器扩展，ram, rom 芯片，38 译码器，自己连线，画出一个框图

难点：片选信号，38 译码器使能端，信号端很好的结合起来

2.cache

作用，基本工作原理，三种。。。（映射方式？），设计主存地址

带 cache 系统的效率，命中率

3.两个存储层次带来什么好处，存在的理由

4.半导体存储器存储单元的工作原理（一般不考察）

5.书 P77-电路，怎么往里写入读出 0 和 1—似乎是无法考察

6.校验不讲

7.提高访存，交叉存储，低位高位

补：cache 写，替换算法，了解了解。。。额

辅助存储器不考

五 .

(重点四五六)

1. 三种方式—后两种更重要
2. 中断相应会中断处理更重要，中断的不同，相应的条件
3. DMA-每个阶段在做什么
4. 三种方式的比较，特点，应用在什么场合

六 . (相当一部分在计算题)

1. 二进制补码的加减运算和溢出判别—一定考
2. 除法不要求，乘法，补码原味，还有一种。。
3. 浮点加减运算
4. 真值—二进制，补码，原码，移码
5. 第五节，最基本的 ALU，输入输出有哪几个组成部分
6. 超前进位连得设计理念和好处，电路不要求

七 .

1. 基本的指令格式
2. 寻址方式—常见的都要求
3. 更重要—设计一定的指令系统的能力 (有这样的作业题)
4. RISC 了解就好。。。感觉就是不考

八 .

cpu 有哪些主要寄存器，常见寄存器的作用

cpu 的功能，操作控制，时间控制

指令周期，建议和后面的时钟周期和中断买中

节拍是啥

指令流水要求

中断系统重点：8.4.4 4.5 4.6

画出 CPU 运行轨迹，中断屏蔽

九 .

微程序工作原理，设计思想

不同机器周期中完成的微操作的描述（加入后面的东西以后，有作业题）

控制方式—了解。。

十

两种控制器的优缺点，应用场合

微程序控制器的设计思想，原理，构成，工作过程

微指令的编码的方式，直接，间接控制

驿马

操作码，后几位地址

如何区分指令和数据（突然补充的，应该要考 P19 1.11）

计算机组成原理复习纲要



徐卫霞 （第九、十章）

杜泽林 （第五、七、八章）

孙吉鹏 （第一、四、六章）

第一章 计算机系统概论

1. 冯诺伊曼计算机的特点 P8
 - 1.计算机由运算器，存储器，控制器，输入设备和输出设备五部分组成
 - 2.指令和数据以同等地位存放于存储器中，并可按地址寻访
 - 3.指令和数据均由二进制数表示
 - 4.指令由操作码和地址码构成，操作码用来表示操作的性质，地址码用来表示操作数在存储器的位置
 - 5.指令在存储器内按顺序存放，通常是按顺序执行，特定条件下可根据运算结果或特定条件改变执行顺序
2. 计算机的工作步骤 P13 （介绍的很全面，串联整本书，适合复习结束看）
3. 计算机的主要技术指标 P17
 - 机器字长：CPU 一次能处理数据的位数，通常与 CPU 寄存器位数有关。
 - 存储容量：主存容量和辅存容量
 - 运算速度：与许多因素有关，主频，操作类型，主存速度等

补充章节：数字逻辑

1. 基本逻辑关系

与 (\bullet) 逻辑乘法

或 (+) 逻辑加法

非

2. 最小项

定义：在一个具有 n 个变量的逻辑函数中，如果一个与项包含了所有 n 个的变量，而且每个变量都是以原变量或反变量的形式作为一个因子仅出现一次，那么这样的与项就称为该逻辑函数的一个最小项。对于 n 个变量的全部最小项共有 2^n 个。

最小项编号：为了表达方便，人们通常用 m_i 表示最小项，其下标 i 为最小项的编号。编号的方法是：最小项中的原变量取 1，反变量取 0，则最小项取值为一组二进制数，其对应的十进制数便为该最小项的编号。

3. 卡诺图化简（详见上课时 数字逻辑 1 课件后半部分 一言难尽提一提重点）

1. 逻辑相邻：相邻单元输入变量的取值只能有一位不同

2. 若两个最小项中只有一个变量以原、反状态相区别，则称它们为逻辑相邻。逻辑相邻的项可以合并，消去一个因子。

3. 相邻单元的个数是 2^N 个，并组成矩形时，可以合并。

4. 先找面积尽量大的组合进行化简，可以减少更多的因子。

5. 各最小项可以重复使用。

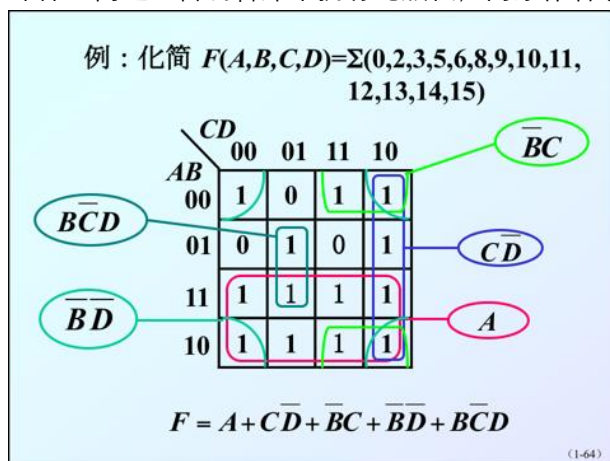
6. 所有的 1 都被圈过后，化简结束。

7. 化简后的逻辑式是各化简项的逻辑和。

8. 圈的数目越少越简；圈内的最小项越多越简。

9. 卡诺图中所有的 1 都必须圈到，不能合并的 1 必须单独画圈。

课件上例题：吉鹏智库不拥有此版权，向原作者致敬！



第四章 存储器

1. 存储器的分类 P68

RAM ? ROM ? 半导体存储器 ? 主存 ?

2. 存储器的层次结构 P70

缓存-主存 主存-辅存

为什么有这种结构？ P71

3. 主存 P73

字？字长？字节？字地址？字节地址？

4. 静态 RAM 和动态 RAM 的不同点 P77

静态为触发器原理，读出后仍保持原状态，不需要再生，电源断电时丢失信息，做实验用的 MM214 就是静态存储器。

动态为靠电容存储电荷原理，不断电也会丢失，需要刷新，比静态集成度高，能耗低，各类计算机广泛使用。但是存取速度慢于静态的。

5. 动态 RAM 的刷新 P86

1. 集中刷新 规定的刷新周期内集中一段时间对所有存储单元逐行刷新，死时间连续易发生冲突。
2. 分散刷新 每行的刷新分散到每个存取周期，但每个存储周期长了，拖慢了系统速度。
3. 异步刷新 用最大刷新间隔除以单元行数算出每个存储单元行的最大刷新时间间隔，以这个间隔为刷新周期。降低了冲突的概率。

6. 半导体存储器的扩展 P91

位扩展 增加存储字长，如让 2 个 1K*4 扩展为 1K*8，本质上就是让一个地址同时选中两片芯片，两片芯片分别存储这个数据的前后部分（高低位）。

字扩展 增加存储器字的数量，如让 2 个 1K*4 扩展为 2K*4，本质上就是把译码前的信号当作地址扩展，译码片选后的信号控制多个芯片的使能端，使一个“地址”只选中一个芯片，因为这时的“地址”位数扩大了（多了片选译码的输入位），实现了存储字的扩展。

字、位扩展

结合起来，一个扩展地址通过译码片选出一个输出位，让这一位同时接两个位扩展芯片的使能端，实现字位扩展结合。

好题：例 4.1 例 4.2

片选小技巧：

1. 如果三八译码器的三个输入地址没法满足片选地址的需求，如需要四位地址来区分各芯片，则考虑将第四位地址线和三八译码器的一条输出与起来，这样可以实现多位区分。
2. 害怕高位地址不做三八译码器的输入导致多选怎么办？把所有高位地址对应成三八译码器或访存的使能端就好。

7. Cache P109

出现原因？（作用）主存 CPU 速度不匹配。可能保障？程序访问的局限性。

命中率？Cache-主存系统平均访问时间？Cache 基本结构原理？P112 图 4.50

Cache 的读写？

写直达法

写时同时写 Cache 和主存，速度慢，但读时快。

写回法

写时只写 Cache，加标志位区分是否与主存对应块相同，只在替换时写回内存，写速度快但读失效时需访存，读时慢。

8. 主存地址，Cache 地址各字段的划分与映射 P117

直接映射

主存和 Cache 低 b 位都为块内地址，Cache 的高 c 位为 Cache 块地址，主存的高地址 m

位被分为 $t+c$, t 为标记位判断对应 Cache 存储块内容有效性。映射公式 $i = j \bmod C$, i 为 Cache 块号, j 为主存块号。缺点不够灵活, 缓存利用率不高。

全相联映射

标记位由 t 变为 $t+c$ 。主存块可以映射到任意 Cache 块, 但实现复杂。

组相联映射

两种方法的结合, 采取块间直接相联, 块内全相联的方法, 将 Cache 分为 $q+r+b$, q 为组位, r 为组内位, b 为块内位, 对应主存分为 $s+q+b$, q 来决定映射到哪几个可能的 Cache 块组, s 为剩余位, 用作字块标记, 决定映射到组内哪一块。一般又称为 2^r 路组相联。

第五章 输入输出系统

一、输入输出系统的概述

1、输入输出系统的发展概况 P156—5.1.1

2、输入输出系统的组成——I/O 软件+ I/O 硬件

(1) I/O 软件

1.三个功能：将用户编制的程序（或数据）输入主机内、将运算结果输送给用户、实现输入输出系统与主机工作的协调。

2.I/O 指令与通道指令详见 P158~P159

(2) I/O 硬件 P159

3、I/O 设备与主机的联系方式

(1) I/O 设备编址方式——统一编址/不统一编址（单独编址）

1.统一编址就是将 I/O 地址看作是存储器地址的一部分。

2.不统一编址就是指 I/O 地址和存储器地址是分开的, 所有对 I/O 设备的访问必须有专用的 I/O 指令。

(2) 设备寻址——实质上是译码

(3) 传送方式——并行传送/串行传送

(4) 联络方式——立即响应方式/异步工作采用应答信号联络/同步工作采用同步时标联络

1.异步工作方式下, I/O 设备与 CPU 各自完成自身的任务, 一旦出现联络信号, 彼此才准备交换信息。详见 P161 图 5.6 给出的模型

2.同步工作要求 I/O 设备与 CPU 工作速度完全同步

(5) I/O 设备与主机的连接方式——辐射式（初级阶段）/总线式

4、I/O 设备与主机信息传送的控制方式

五种方式：程序查询方式、程序中断方式、直接存储器存取方式 DMA、I/O 通道方式、I/O 处理机方式。主要介绍前三种方式。

(1) 程序查询方式——P163 图 5.9

CPU 通过程序不断查询 I/O 设备是否已做好准备, 若设备未准备就绪, 就继续查询, 直到查得设备准备就绪, 就将数据从 I/O 接口送至 CPU。这种方式下 CPU 和 I/O 设备处于串行工作状态, 在反复的查询过程中 CPU 不能执行原程序, 相当于原地踏步, 工作效率不高。

(2) 程序中断方式——P164 图 5.11

CPU 启动 I/O 设备后, 继续执行自身程序, 只是当 I/O 设备准备就绪并向 CPU 发出中断请求后才予以响应。这种方式解决了 CPU 原地踏步的问题。这种方式将在之后详细

介绍。

(3) DMA 方式

主存与 I/O 设备之间有一条数据通路，不需要调用中断服务程序就可以直接访问主存。当这种访问与 CPU 访问主存发生冲突时，CPU 总是将总线占有权让给 DMA，此时称为窃取周期。这种方式将在之后详细介绍。

二、I/O 设备——输入设备+输出设备

详见 P166—5.2

三、I/O 接口

1、总线方式的 I/O 接口：数据线+设备选择线+命令线+状态线

2、接口的功能和组成：选址+传送命令+传送数据+反映 I/O 设备工作状态

3、接口类型

(1) 数据传送方式：并行接口、串行接口

(2) 功能选择的灵活性：可编程接口、不可编程接口

(3) 通用性：通用接口、专用接口

(4) 数据传送控制方式：程序型接口、DMA 型接口

四、程序查询方式

程序查询方式的核心问题在于每时每刻不断查询 I/O 设备是否准备就绪。

程序查询流程及程序查询方式的接口电路请仔细阅读 P191 图 5.33、图 5.34 和 P192 图 5.35

五、程序中断方式

这里主要介绍 I/O 中断处理，有关中断的其他内容请参考第八章。

1、I/O 中断的产生

CPU 启动 I/O 设备后，继续执行自身程序，等到设备完成工作并向 CPU 发出准备就绪请求后，再转入 I/O 中断服务程序。P194 图 5.36 有助于理解程序中断方式。

2、程序中断方式的接口电路

(1) 中断请求触发器 INTR&中断屏蔽触发器 MASK

1.当多个中断源向 CPU 提出请求时，CPU 必须坚持任何瞬间只能接受一个请求的原则。

2.CPU 总是在同一时间，执行完每条指令后发出查询信号。

3. INTR 为“1”时表示提出中断请求，MASK 为“1”时表示封锁其中断源的请求，接口内完成触发器 D 的状态为 1 表示设备就绪。结合 P195 图 5.37 理解 INTR、MASK 与 D 的关系。

(2) 排队器 P196 图 5.38

1.此处介绍的排队器是设在各个接口电路中的排队器，又称链式排队器。设在 CPU 内部的排队器请参考第八章 P361 图 8.25

2.下面一排门电路是其核心。当它对应的 INTR 输入为 1 时便迫使其后的所有 INTP 变为低电平，封锁他们的中断请求。

(3) 中断向量的形成部件——设备编码器

向量地址≠中断服务程序的入口地址

(4) 程序中断方式接口电路的基本组成 P198 图 5.41

3、I/O 中断处理过程

中断请求→中断判优→中断响应→中断服务→中断返回

详见 P198——2.I/O 中断处理过程

4、中断服务程序的流程

(1) 保护现场

保护程序断点（中断隐指令）、保护 CPU 内部寄存器数据（中断服务程序）

(2) 中断服务

处理中断请求对应的问题

(3) 恢复现场

将 CPU 内部寄存器的数据恢复到中断前

(4) 中断返回

返回程序断点，继续执行原程序

*CPU 一旦响应了某中断源的请求后，便由硬件线路自动关中断。此后如果不用指令开中断，意味着当前中断服务程序执行过程中不在响应其它中断请求。如果在保护现场后用软件指令开中断，那么 CPU 可以在执行当前中断服务程序时响应优先级更高的中断，从而形成多重中断。请仔细阅读 P201 图 5.43

六、DMA 方式

1、DMA 方式的特点

主存与 I/O 设备之间有一条数据通路，不需要调用中断服务程序就可以直接访问主存。

由于 DMA 接口和 CPU 共享主存，有可能造成冲突，故采用以下三种方法完成 DMA 与主存交换数据

(1) 停止 CPU 访问主存

(2) 周期挪用

(3) DMA 与 CPU 交替访问

2、DMA 接口的功能和组成

(1) DMA 接口的功能：向 CPU 申请 DMA 传送、处理总线控制权的转交、在 DMA 期间管理系统总线、确定传送的起始地址和数据长度、数据传送完成后发出 DMA 完成信号

(2) DMA 接口的组成

详见 P205 图 5.47

3、DMA 的工作过程

(1) DMA 传送过程

预处理→数据传送→后处理；详见 P206~P208

(2) DMA 接口与系统的连接方式 P209 图 5.49

具有公共请求线的 DMA 请求&独立的 DMA 请求

*与程序中断方式相比 DMA 方式的特点：

1.从数据传送看，程序中断方式靠程序传送，DMA 方式靠硬件传送。

2.从 CPU 响应时间看，程序中断方式是在一条指令执行结束时响应，而 DMA 方式可在指令周期内的任一存取周期结束时响应。

3.程序中断方式有处理异常的能力，DMA 方式没有这种能力。DMA 方式主要用于大批数据的传送，如硬盘存取、图像处理、高速数据采集系统，可提高数据吞吐量。

4.程序中断方式需要中短现程序，故需保护现场；DMA 方式不中断现程序，无需保护现场。

5.DMA 的优先级比程序中断优先级高。

第六章 计算机的运算方法

1. 进位计数制之间的转换（整数，小数）P293

2. 定点数：原反补之间的转换 P220

3. 浮点数：给出给定格式的浮点数代码写出真值 P229

229 页中间的 N 的阶码是二进制表示法

浮点数定义，尾数，阶码，基数，规格化数（好处？），浮点数表示范围，上 / 下溢出，如何规格化浮点数

好题：例 6.6

移位运算：补码的好处，左移符号位不变，补 0；右移符号位不变，补符号位

4. 二进制补码的加减运算及溢出判断 P235

例 6.10, 6.11

($[-B]$ 补) = ($[B]$ 补) 符号位求反后再求补

一位符号位判溢出：观察两运算数与结果符号位是否相同

两位符号位判溢出：真符号看高位，结果两位符号位不同则溢出。

5. 定点原码一位乘法和定点补码一位乘法的计算过程 P243

乘，加，移

表 6.8 例 6.17

1) 符号位先心算，两数的绝对值进行定点一位乘

2) 几位运算数就运算几次加法和右移

3) 乘数末位值确定被乘数是否与原部分积相加，之后结果右移一位，形成新部分积，高位补零，之后乘数也右移一次，次低位做新的最低位，最高位放部分积的最低位。

4) 每次加法，被乘数仅与原部分积高位相加，低位被移至乘数所空出的高位位置。

5) 最后的结果是 符号位 . 部分积 乘数

补码一位乘 P250

例 6.19 例 6.20

符号位运算中自然生成，运算步骤

乘数 > 0

运算步骤与原码一位乘相同，乘加移，只不过符号位变为两位，运算时部分积符号位一直补 1，部分积为补码，尾数右移高位始终补 1。

乘数 < 0

按原码一位乘的运算规则运算，最后加上 ($[-x]$ 补) 负的被乘数的补码校正。

6. 浮点数的加减运算 P269

例 6.30

1) 对阶，小阶向大阶看齐，之后小阶尾数补码右移补 1，可能会丢失精度

2) 尾数求和

3) 规格化， $1/2 \leq |S| < 1$, 补码实现

两符号位相同，符号位与数值位第一位相同则左移， $|S| < 1/2$

两符号位不同，说明上下溢出，需要右移尾数使其减小，符号位右移给到数值位高位，直到符号位相同，符号位与第一位数值位不同规格化结束。

4) 舍入判断

0 舍 1 入（右移丢谁再加谁）；恒置 1 法（尾数末位横置 1）

5) 溢出判断

观察阶符，阶符两位 01/10 则发生上 / 下溢出

7. 原码一位除 P258

例 6.25

不恢复余数法，加减交替法

符号位自行异或判断

$[x/y]$ 原 保证 $x^* < y^*$

$r > 0$ $2r - y^*$, 减去除数

$r < 0$ $2r + y^*$, 加除数

步骤为 ; 判正负, 上商 10, 余数和商左移, 加减除数

第一次被除数为余数, $y^* > x^*$, 减去除数后结果为负, 第一次一定上商 0, 之后有几位就左移, 加减, 判余数正负, 上商几次。

结果为符号位. 商的后 n 位 (第一位选择正常一定为 0)。

左移余数末位补 0

第七章 指令系统

一、指令的一般格式：

1、操作码+地址码。

2、操作码的长度可以是固定的也可以会变化的, 通常采用拓展操作码技术, 使操作码的长度随地址数的减少而增加。P301 图 7.2& P302 例 7.1

3、可按地址数的不同对指令分类。P303

4、指令字长 P304

二、寻址方式

(一) 指令寻址

1、顺序寻址： $PC+1$ 。

2、跳跃寻址：转移类指令。

(二) 数据寻址

1、指令中的寻址特征字段指明寻址方式

2、区分形式地址 A 和有效地址 EA

3、十种寻址方式 (详见 P310~P319)

(1) 立即寻址—立即数 (图 7.8)

(2) (存储器) 直接寻址： $EA=A$; A 的位数限制寻址范围 (图 7.9)

(3) 隐含寻址— EA 由操作码确定 (图 7.10)

(4) 间接寻址： $EA=(A)$; 与直接寻址相比, 它扩大了寻址范围; 访存次数增加 (图 7.11)

(5) 寄存器 (直接) 寻址：不需要访问内存 (图 7.13)

(6) 寄存器间接寻址：操作数地址存放在寄存器中 (图 7.14)

(7) 基址寻址： $EA=A+(BR)$; 基址寄存器中的数据+形式地址→有效地址 (图 7.15)

(8) 变址寻址： $EA=A+(IX)$; 变址寄存器中的数据+形式地址→有效地址 (图 7.16)

*注意区分 (7) (8) 见 P315 最后一段

(9) 相对寻址： $EA=(PC)+A$; 位移量 A 通常是补码 (图 7.17)

*P318 例 7.2

(10) 堆栈寻址—可被视为一种隐含寻址 (图 7.18)

三、设计指令

P323~P325 例 7.4/7.5/7.6/7.7

P335 题 7.16

四、RISC 技术

1、RISC 技术的起因 P326

- 2、RISC 的特点 P330
- 3、RISC 和 CISC 的比较 P333

第八章 CPU 的结构和功能

一、CPU 的结构——运算器+控制器

(一) CPU 的功能

- 1、本章重点介绍控制器的功能，运算器的功能在第六章
- 2、控制器的基本功能：取指令+分析指令+执行指令（具体介绍见 P337）

(二) CPU 的结构

ALU+寄存器+中断系统+CU+系统/内部总线

P338 图 8.1、图 8.2

(三) CPU 的寄存器 P338~P341

- 1、用户可见（用户可以对其操作）：通用寄存器、数据寄存器、地址寄存器、条件码寄存器
- 2、控制和状态寄存器（通常用户不能对其操作）

MAR 存储器地址寄存器、MDR 存储器数据寄存器、PC 程序计数器、IR 指令寄存器

*以上四种寄存器在指令执行过程中起重要作用，以取指令为例：PC→MAR→M→MDR→IR
还有 PSW 程序状态字寄存器

二、指令周期——CPU 取出并执行完一条指令的全部时间

取指周期+（间址周期）+执行周期+（中断周期）

注：上面给出的指令周期中间址周期实质上也是在完成取指的工作，是否进行间址要根据指令的寻址特征判断。

三、指令流水

四、中断系统——中断请求+中断判优+中断响应+终端服务+中断返回

(一) 引起中断的因素&终端系统需解决的问题 P359CPU 响应中断的条件：中断请求&&没有屏蔽&&程序状态字允许中断

(二) 中断请求标记和中断判优逻辑

- 1、INTR 中断请求触发器，它可以分散在接口中，也可以集中设在 CPU 内组成中断请求标记寄存器。

2、中断判优逻辑：

(1) 硬件排队：

1.链式排队器：INTR 分散在各个接口电路中的情况 P196 图 5.38

2. 集中在 CPU 内的排队器 P361 图 8.25

*两种排队器都通过电路设计达到优先级高的信号封锁优先级低的信号的目的。

(2) 软件排队 P361 图 8.26

(三) 终端服务程序入口地址的寻找 P361~P362

- 1、硬件向量法——中断向量的形成部件实质上是一个编码器，它的输入来自排队器的输出。
- 2、软件查询法

(四) 中断响应 P362~P364

- 1、CPU 响应中断的条件：中断请求&&没有屏蔽&&程序状态字允许中断
- 2、响应中断的时间：指令执行结束后，CPU 发出终端查询信号，CPU 获知中断请求。
- 3、中断隐指令——进入中断周期，CPU 自动完成的一系列操作
 - (1) 保护程序断点
 - (2) 寻找中断服务程序入口地址——向量地址送至 PC，使 CPU 执行下一条无条件转

移指令，转至终端服务程序入口地址

(3) 关中断

4、保护现场和恢复现场

保护现场：保护程序断点（由中断隐指令完成）+保护 CPU 内部各寄存器的内容（由中断服务程序完成）

恢复现场：恢复 CPU 内部各寄存器的内容到中断前（由中断服务程序完成）

5、中断屏蔽技术——主要用于多重中断

(1) 多重中断（中断嵌套）：新出现的中断可能打断当前执行的中断服务程序，转去执行新的中断服务程序。

(2) 实现多重中断的条件

1. 提前设置开中断指令（在保护 CPU 内部寄存器之后，防止数据丢失）

2. 优先级别高的中断源有权中断优先级别低的中断源

(3) 屏蔽技术——MASK 屏蔽触发器+屏蔽字 P366 图 8.33

1. 每个 INTR 对应一个 MASK，将所有 MASK 组合在一起构成屏蔽寄存器，屏蔽寄存器中的内容被称为屏蔽字。

2. 屏蔽技术可以改变优先级——让中断服务程序加载新屏蔽字

*P368 图 8.35 中处理完 A 之后经由 B、C 逐级进入 D 的期间就是在加载 B 和 C 新的屏蔽字。

*置新屏蔽字后要在服务程序最后恢复原屏蔽字。

*具体过程见 P368 图 8.36，P369 例 8.2

第九章 控制单元的功能

理解指令周期、机器周期、时钟周期、节拍、控制信号的关系，重点：控制单元

一、指令周期：

1. 取指周期： $(PC) \rightarrow MAR$;

$M(R), M(MAR) \rightarrow MDR$; 此处注意“读信号”的持续时间

$MDR \rightarrow IR, (PC)+1 \rightarrow PC$;

2. 间址周期：完成取操作数有效地址

$IR(AD) \rightarrow MAR$;

$M(R), M(MAR) \rightarrow MDR$;

$MDR \rightarrow IR(AD)$; PS：此处指令寄存器的数据地址字段比 MDR 的长度要小，这样写会有错误，此处是为了突出间址。

3. 执行周期：P376-378 P378 图 9.1

4. 中断周期（详情见第八章、第五章中断部分）

相关题目：P393 9.1

二、控制单元：主要功能是发出各种微操作命令（控制信号）序列

了解控制单元：可根据 P379 图 9.2 了解 P379 内容

“ADD @X”为例理解指令流程各阶段相应的控制信号：(P376—P377)

1. 不采用 CPU 内部总线的方式

2. 采用 CPU 内部总线的方式

PS：两种方式的基本流程相同，只是有 CPU 内部总线的方式更简洁明朗，各种信号大都上总线进行传输。

相关题目：P382 例 9.1，9.2

三、多级时序：

机器周期：所有指令执行的一个基准时间，在各种操作中访问存储器的时间最长，故通常以访问一次存储器的时间定为机器周期（在同步控制条件下成立）

时钟周期（节拍）：完成一组微操作命令的时间，节拍的时间与时钟周期相同

指令周期、机器周期、时钟周期、节拍的关系：指令周期是 CPU 完成一条指令的时间，它包括若干个机器周期，一个机器周期包括若干个时钟周期（节拍）。P387 图 9.9

此部分：P386 图 9.8。

*（了解）控制单元的控制方式：同步控制方式、异步控制方式、联合控制方式、人工控制方式。

第十章 控制单元的设计

控制单元设计分为组合逻辑设计和微程序设计两种方法

一、组合逻辑设计：

取指周期是公共部分，间址周期和执行周期根据寻址方式的不同和指令的不同功能相应的微操作不同，相应的控制信号也不尽相同。

步骤：

1. 列出微操作命令的操作时间表：

具体步骤：各个指令的取指周期步骤相同，故对应的微操作命令也相同。写出各个指令执行流程的各个节拍里的微操作命令，得到操作实践表 P402 表 10.1

写各个指令执行流程的微操作命令时，注意节拍安排：P396 写流程相关例题：P398 例 10.2

2. 得到各个控制信号的最简逻辑表达式：输入信号（如 MAR_i, PC_i 等）需要配合 CP，输出信号则不需要。

如 $MAR_i : (T_0 + T_3(MOV + STA + ADD)) \cdot CP$ ，（解释： MAR_i 出现在了 T_0 节拍中， T_3 节拍中的所有命令并没有都出现 MAR_i ，故选择出现了的 MOV, STA, ADD 命令与 T_3 进行“与操作”， MAR_i 还需跟 CP 配合。）

3. 画逻辑电路 P403 图 10.3

二、微程序设计

微命令：同一节拍中执行的一个或一组微操作

微指令：由微命令和下地址字段组成

微程序：一组可以完成特定功能的微指令。例如取指周期有三个阶段，故分为三条微指令。

控制存储器（简称控存）：CM，位于 CU 中，用于存放所有微程序 详细结构见 P404 图 10.4

控存地址寄存器： μMAR

控存数据寄存器： μIR ，用于存放从控存读出的微指令，在执行时预取下一条微指令，从而形成流水线，并且“不断流”

执行流程：

取指阶段：（取指周期微程序是公共部分），例如取指阶段的三条微指令存于控存 00000, 00001, 00010 中，将取指周期微程序首地址（00000）送入 μMAR ，进入取值阶段，通过 μMAR 内容加“1”跟换地址信息，从而依次执行 00001, 00010。

执行周期：取指周期执行完 00010 后 IR 的内容（ ADD, STA, LDA 等指令的执行周期的微程序在控存中的首地址）送到 μMAR 中，开始执行周期，执行周期执行完后跳转到取指周期首地址，再开始新的微程序的执行。

下地址字段：包括“方式”和下一条微指令的地址，此处“方式”可举例：“1”表示顺序执

行，“0”表示跳转

微指令的编码方式：1.直接编码方式：微指令的的微命令字段中，每一位都代表一个微操作命令，P407 图 10.7

2.字段直接编码方式：互斥微命令放在一个字段中，减少了微命令字段的位数 P407 图 10.8

3.字段间接编码方式 4.混合编码等

后续微指令的地址的形成方式：

1.直接由微指令的下地址字段给出，又称为断定地址

2.根据机器指令的操作码形成

3.增量计数器法 4.分支转移 5.通过测试网络形成 6.由硬件产生微程序入口地址

微指令格式：1.水平型微指令 2.垂直型微指令

PS：微指令的存储方式：（补充说明，课本上没有）

1. 微命令部分放在微指令的后半部分：可以充分利用存储空间。

2. 微命令部分放在微指令的前半部分：不能可以充分利用存储空间，造成空间浪费

3. MAPROM：MAPROM 用于存放所有微程序的编码（如 ADD 为 000，STA 为 001 等）和它们在控存中的首地址，程序执行时先通过 MAPROM 找到微程序首地址，然后执行。类似于间址寻址，要访问两次存储器，速度比较慢。

1

计组部分重要概念整理

冯诺依曼计算机的特点：

- 1 计算机有运算器、存储器、控制器、输入设备和输出设备五大部件组成
- 2 指令和数据以同等地位存放于存储器内。并可按地址访问。
- 3 指令和数据均可用二进制表示
- 4 指令由操作码和地址码组成。操作码表示操作的性质、地址码表示操作数在存储器中的位置
- 5 指令在存储器中按顺序存放。通常，指令时顺序执行的。在特殊情况下，可根据运算结果或指定的条件来改变运算顺序。
- 6 机器以运算器为中心。输入输出设备和存储器之间的数据传送通过运算器完成。

典型的冯诺依曼计算机是以运算器为中心的，现代的计算机已转化为以存储器为中心的。

计算机硬件的主要技术指标：

机器字长：寄存器的位数

存储容量：存储单元个数*存储字长= $MAR * MDR$

运算速度：主频 MIPS（百万） 时钟周期 CPI

存储器的层次结构主要体现在什么地方？为什么要分这些层次？

1) 存储器的层次结构主要体现在 Cache—主存和主存—辅存这两个存储层次上。

2) Cache—主存层次主要解决 CPU 和主存速度不匹配的问题，在存储系统中主要对 CPU 访存起加速作用。从 CPU 的角度看，该层次的速度接近于 Cache，而容量和每位价格却接近于主存。这就答案要点决了存储器的高速度和低成本之间的矛盾；

主存—辅存层次主要解决存储系统的容量问题，在存储系统中主要起扩容作用。从程序员的角度看，其所使用的存储器的容量和每位价格接近于辅存，而速度接近于主存。该层次答案要点决了大容量和低成本之间的矛盾。

试比较静态 RAM 和动态 RAM。

答案要点：1) 静态 RAM 的特点：依靠双稳态触发器保存信息，不断电信息不丢失；功耗较大，集成度较低，速度快，每位价格高，适合于作 Cache 或存取速度要求较高的小容量主存。

2) 动态 RAM 的特点：依靠电容存储电荷来保存信息，需刷新电路进行动态刷新；

功耗较小，集成度高，每位价格较低，适合于作大容量主存。

什么叫刷新？为什么要刷新？说明刷新有几种方法。说明动态 RAM 各种刷新方式的特点。

答案要点：

1) 为防止信息丢失，将动态 RAM 的存储单元中的原存信息读出，并重新写入的数据再生过程，称为刷新。

2) 由于电容极板漏抗的存在，存储于电容中的电荷存在泄漏的情况，这会导致动态 RAM 存储单元中的原存信息丢失，因此，必须要进行刷新。

3) 动态 RAM 的刷新方式有集中式刷新、分散式刷新、异步式刷新和透明式刷新等四种方式。（只答前三种也可）

4) 各种刷新方式的特点如下：

集中式刷新的特点：在最大刷新闻隔时间内，集中安排一段时间进行刷新。其缺点是进行刷新时必须停止读、写操作。这对主机而言是个“死区”

分散式刷新的特点：刷新工作安排在系统的存取周期内进行，对主机而言不再有“死区”。但该方式加长了系统的存取周期，存在无谓刷新，降低了整机运行效率。因此，分散方式刷新不适用于高速存储器。

异步式刷新的特点：结合了上述两种方式的优点，充分利用了最大刷新闻隔。相对于分散式刷新而言，它减少了刷新次数；相对于集中方式来说，主机的“死区”又缩短很多。因此，这种方式使用得比较多。

透明式刷新的特点：该方式不占用 CPU 时间，对 CPU 而言是透明的操作；但控制线路复杂。

简述 Cache-主存地址映射有哪几种方式，以及各自的优缺点。

答：Cache-主存地址映射有直接映射方式、全相联映射方式和组相联映射方式三种。

直接映射方式的特点：主存的字块只可以和固定的 Cache 字块对应，优点是方式直接，硬件实现电路简单，成本低；缺点是利用率低，同时命中率和效率较低。

全相联映射方式主存中的字块可以和 Cache 的任何字块对应，优点是方式灵活，利用率高；缺点是所需逻辑电路复杂，使用成本太高。

组相联映射方式是对前两种映射方式的折衷，**组内全相联，组间直接映像**。其特点是集中了两个方式的优点，成本也不太高，是目前应用最为广泛的 Cache 映射方式。

I/O 设备与主机交换信息时，共有哪几种控制方式？简述它们的特点。

参考答案要点：

I/O 设备与主机交换信息时，共有 5 种控制方式：程序查询方式、程序中断方式、DMA 方式、I/O 通道方式和 I/O 处理机方式。其中前 3 种方式是基本的且广泛应用的控制方式。

程序查询方式的特点：控制简单，硬件开销小；CPU 与外设是串行工作的，系统效率低。适用于 CPU 不太忙且传送速度要求不太高的场合。

程序中断方式的特点：CPU 和外设可并行工作，提高了 CPU 的效率，不仅适于主机和外设之间的数据交换，还特别适于对外界随机事件的处理。适用于 CPU 较忙，传送速度不太高的系统中，尤其适合实时控制及紧急事件的处理。

DMA 方式的特点：完全由硬件（DMA 控制器）负责完成信息交换，信息传递从以 CPU 为中心，转为以内存为中心，CPU 和外设可并行工作，对高速大批量数据传送特别有用。但缺点是只能进行简单数据交换，电路结构复杂，硬件开销大。

接口的概念：

主机与 I/O 设备之间设置的一个硬件电路及其相应的软件控制。

什么叫中断？

计算机在执行程序的过程中，当出现异常情况或特殊请求时，计算机停止现行程序的运行，转向对这些异常情况和特殊请求的处理，处理结束后再返回到现行程序的间断处，继续执行源程序，这就是中断。

CPU 响应中断的条件是什么？

参考答案要点：

CPU 响应中断的条件可以归纳为三条：

- 1) 有中断请求；
- 2) CPU 允许中断，即中断允许状态 $IF=1$ （或 $EINT=1$ ）；

3) 一条指令执行结束。

试比较单重中断和多重中断服务程序的处理流程，说明它们不同的原因。

参考答案要点：

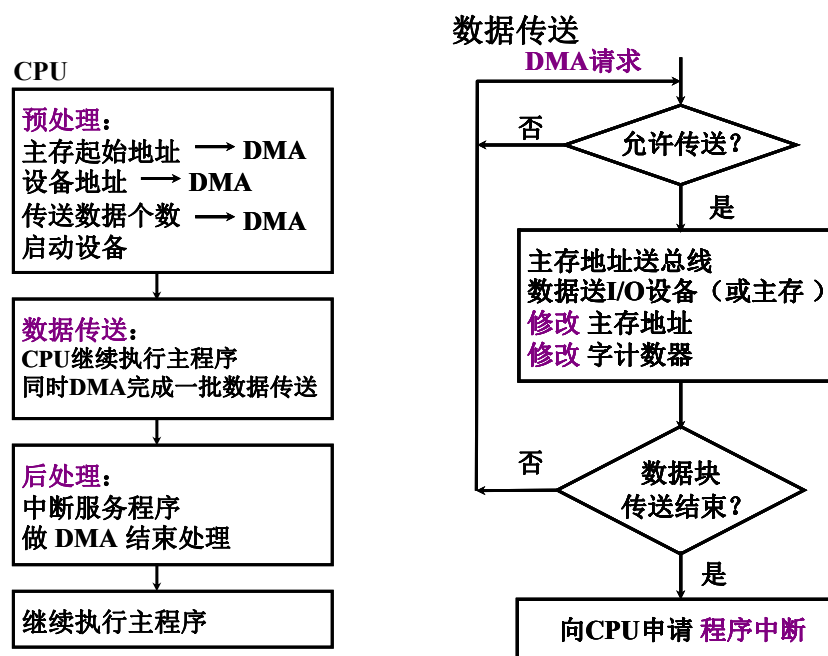
1) 二者的比较可用两种中断的服务程序流程图（见教材 P201）的对比来说明，此处略。

2) 单重中断和多重中断的区别在于“开中断”的设置时间不同。对于单重中断，开中断指令设置在最后“中断返回”之前，意味着在整个中断服务处理过程中，不能再响应其他中断源的请求。而对于多重中断，开中断指令提前至“保护现场”之后，意味着在保护现场之后，若有更高级别的中断源提出请求，CPU 也可以响应，从而实现中断嵌套，这是二者的主要区别。

结合 DMA 接口电路说明其工作过程。

参考答案要点：

DMA 的数据传送过程可分为预处理、数据传送和后处理 3 个阶段。工作过程如下图所示：



各阶段完成的工作如下：

- 1) 预处理阶段：CPU 执行主程序实现 DMA 传送的初始化设置；
- 2) 数据传送阶段：由 DMA 控制器实现内存和外设间的数据传送。

3) **后处理阶段**：中断处理程序判断传送的正误，对写入主存的数据进行校验，完成善后工作。

DMA 方式与程序中断方式的不同：

对照书或者 PPT 总结一下，列个表格比较容易记忆。

ALU 的功能和组成：

功能：既能完成算术运算有能完成逻辑运算。

组成：核心部件是加法器和寄存器。

寻址方式的含义：

是指确定本条指令的数据地址以及下一条将要执行的指令地址的方法。

什么是指令周期？指令周期是否有一个固定值？为什么？

- 1) 指令周期是指 CPU 每取出并执行一条指令所需的全部时间。
- 2) 由于计算机中各种指令执行所需的时间差异很大，因此为了提高 CPU 运行效率，即使在同步控制的机器中，不同指令的指令周期长度都是不一致的，也就是说指令周期对于不同的指令来说不是一个固定值。
- 3) 指令周期长度不一致的根本原因在于设计人员，为了提高 CPU 运行效率而这样安排的，指令功能不同，需完成的微操作复杂程度亦不同，因此，不同指令的指令周期也不同。

变址寻址和基址寻址的异同点：

基址寻址主要为程序或数据分配存储空间，故基址寻址的内容通常由操作系统或者管理程序确定，在程序的执行过程中其值是不可变的，而指令字中的 A 是可变的。在变址寻址中，变址寄存器的内容由用户设定，在程序执行过程中其值可变，而指令字中的 A 是不可变的。变址寻址主要用于处理数组问题。

中断周期前是什么阶段？中断周期后又是什么阶段？在中断周期 CPU 应完成什么操作？

中断周期前是指令的执行阶段（处于执行周期）。中断周期后是取指令阶段（处于取指周期）。在中断周期中，CPU 应完成关中断、保存断点和转中断服务程序入口三个操作。

什么是指令周期、机器周期和时钟周期？三者有何关系？

CPU 每取出并执行一条指令所需的全部时间叫指令周期；机器周期是在同步控制的机器中，所有指令执行过程中（执行一步相对完整的操作）的一个基准时间，通常以访问一次存储器所需的时间作为一个机器周期；时钟周期是指计算机主工作时钟的周期时间，它是计算机运行时最基本的时序单位，通常时钟周期=计算机主频的倒数。

三者之间的关系：指令周期常常用若干个机器周期数来表示，机器周期也叫 CPU 周期；而一个机器周期又包含若干个时钟周期（也称为节拍脉冲或 T 周期）。

CPU 的结构和功能

结构：书 338 页图 8.1 和图 8.2

功能：8.1.1 好好看看（我懒不想打 TT）

试比较组合逻辑设计和微程序设计的设计步骤和硬件组成，说明哪一种控制速度更快，为什么？

一）设计步骤

组合逻辑控制器的设计步骤：1）拟定机器的指令系统；2）确定 CPU 总体结构；3）确定时序系统，拟定指令流程；4）安排每条指令中微操作的节拍；5）列出微操作命令的操作时间表；6）写出每一个微操作命令的逻辑表达式并化简；7）画出相应的组合逻辑电路图。

微程序控制器的设计步骤：前三个步骤和组合逻辑控制器相同，后边的步骤如下：

- 1）写出对应机器指令的微操作及节拍安排；
- 2）确定微指令格式（确定微指令的编码方式和后继微地址的形成方式）；
- 3）编写微指令码点。

二）硬件组成：组合逻辑控制器由组合逻辑电路提供微命令，其核心器件是各种门电路构成的复杂树形网络；微程序控制器由存储逻辑(微指令)提供微命令，其

核心器件是控制存储器。

三) 组合逻辑控制器速度更快, 因为其微命令全部由硬件 (组合逻辑门电路) 产生。

说明微程序控制器的基本工作原理。

将控制器所需要的微操作命令, 以微代码的形式编成微指令, 存在专门的控制存储器中, CPU执行机器指令时, 从控制存储器中取出微指令, 对微指令中的操作控制字段进行解释, 即产生执行机器指令所需的微操作命令序列。

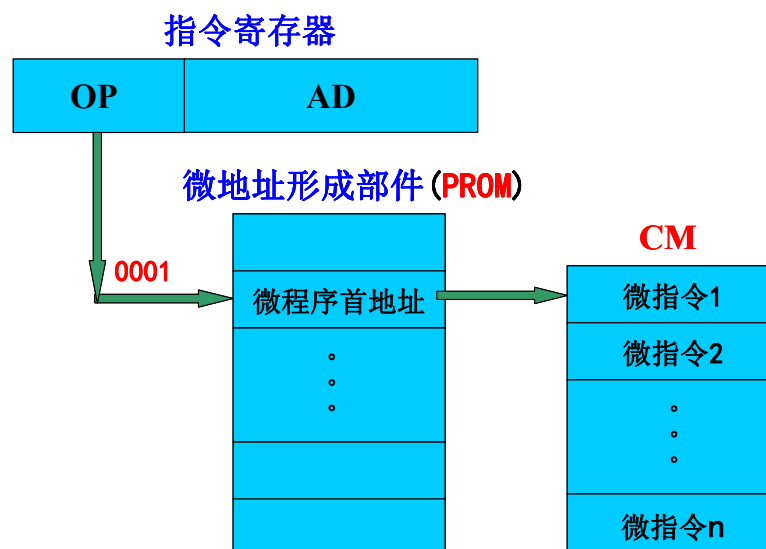
其具体的工作过程如下:

首先将用户程序的首地址送至PC, 然后进入取指阶段。

- 1) 取机器指令: 从控制存储器中读取“取指微指令”, 用产生的微命令控制CPU访存, 读取机器指令, 并送入指令寄存器IR。
- 2) 形成微程序入口地址: 根据机器指令的操作码, 通过微地址形成电路, 产生与该机器指令对应的微程序入口地址, 并送入CMAR。
- 3) 逐条取出机器指令对应的微程序并执行之。
- 4) 返回取指微指令, 开始又一条机器指令的执行。如此不断重复, 直到整个程序执行完为止。

画图并说明微程序控制器中是如何根据操作码形成相应微程序入口地址的。

参考答案要点:



如上图所示, 机器指令取至指令寄存器后, 指令的操作码作为微地址形成部件的输入, 来形成微指令的地址。因此, 可以把微地址形成部件理解为一个编码

器。微地址形成部件可采用 PROM 实现，即以指令的操作码作为 PROM 的地址，而相应的存储单元中就存放着对应该指令微程序的首地址。

操作控制字段常见的几种编码方法：

- 1 直接编码（直接控制）方式
- 2 字段直接编码方式
- 3 字段间接编码方式
- 4 混合编码
- 5 其他

注意：红色标注的题目都要看看

第1章 计算机系统概论

1. 什么是计算机系统、计算机硬件和计算机软件？硬件和软件哪个更重要？

解：P3

计算机系统：由计算机硬件系统和软件系统组成的综合体。

计算机硬件：指计算机中的电子线路和物理装置。

计算机软件：计算机运行所需的程序及相关资料。

硬件和软件在计算机系统中相互依存，缺一不可，因此同样重要。

2. 如何理解计算机的层次结构？

答：计算机硬件、系统软件和应用软件构成了计算机系统的三个层次结构。

（1）硬件系统是最内层的，它是整个计算机系统的基础和核心。

（2）系统软件在硬件之外，为用户提供一个基本操作界面。

（3）应用软件在最外层，为用户提供解决具体问题的应用系统界面。

通常将硬件系统之外的其余层称为虚拟机。各层次之间关系密切，上层是下层的扩展，下层是上层的基础，各层次的划分不是绝对的。

3. 说明高级语言、汇编语言和机器语言的差别及其联系。

答：机器语言是计算机硬件能够直接识别的语言，汇编语言是机器语言的符号表示，高级语言是面向算法的语言。高级语言编写的程序（源程序）处于最高层，必须翻译成汇编语言，再由汇编程序汇编成机器语言（目标程序）之后才能被执行。

4. 如何理解计算机组成和计算机体系结构？

答：计算机体系结构是指那些能够被程序员所见到的计算机系统的属性，如指令系统、数据类型、寻址技术组成及 I/O 机理等。计算机组成是指如何实现计算机体系结构所体现的属性，包含对程序员透明的硬件细节，如组成计算机系统的各个功能部件的结构和功能，及相互连接方法等。

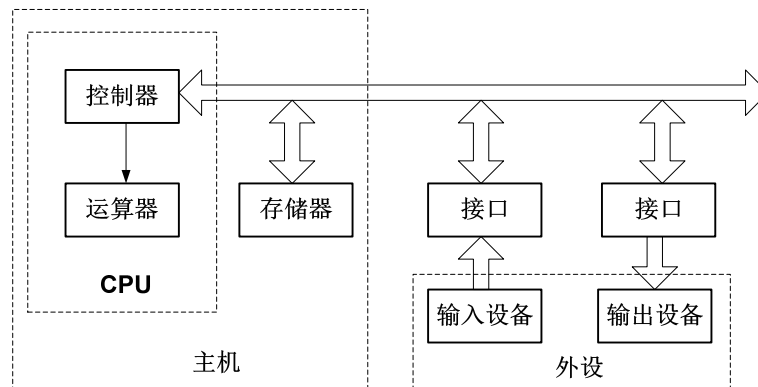
5. 冯·诺依曼计算机的特点是什么？

解：冯·诺依曼计算机的特点是：P8

- 计算机由运算器、控制器、存储器、输入设备、输出设备五大部件组成；
- 指令和数据以同等地位存放于存储器内，并可以按地址访问；
- 指令和数据均用二进制表示；
- 指令由操作码、地址码两大部分组成，操作码用来表示操作的性质，地址码用来表示操作数在存储器中的位置；
- 指令在存储器中顺序存放，通常自动顺序取出执行；
- 机器以运算器为中心（原始冯·诺依曼机）。

6. 画出计算机硬件组成框图，说明各部件的作用及计算机系统的主要技术指标。

答：计算机硬件组成框图如下：



各部件的作用如下：

控制器：整机的指挥中心，它使计算机的各个部件自动协调工作。

运算器：对数据信息进行处理的部分，用来进行算术运算和逻辑运算。

存储器：存放程序和数据，是计算机实现“存储程序控制”的基础。

输入设备：将人们熟悉的信息形式转换成计算机可以接受并识别的信息形式的设备。

输出设备：将计算机处理的结果（二进制信息）转换成人类或其它设备可以接收和识别的信息形式的设备。

计算机系统的主要技术指标有：

机器字长：指 CPU 一次能处理的数据的位数。通常与 CPU 的寄存器的位数有关，字长越长，数的表示范围越大，精度也越高。机器字长也会影响计算机的运算速度。

数据通路宽度：数据总线一次能并行传送的数据位数。

存储容量：指能存储信息的最大容量，通常以字节来衡量。一般包含主存容量和辅存容量。

运算速度：通常用 MIPS（每秒百万条指令）、MFLOPS（每秒百万次浮点运算）或 CPI（执行一条指令所需的时钟周期数）来衡量。CPU 执行时间是指 CPU 对特定程序的执行时间。

主频：机器内部主时钟的运行频率，是衡量机器速度的重要参数。

吞吐量：指流入、处理和流出系统的信息速率。它主要取决于主存的存取周期。

响应时间：计算机系统对特定事件的响应时间，如实时响应外部中断的时间等。

7. 解释下列概念：

主机、CPU、主存、存储单元、存储元件、存储基元、存储元、存储字、存储字长、存储容量、机器字长、指令字长。

解：P9-10

主机：是计算机硬件的主体部分，由 CPU 和主存储器 MM 合成为主机。

CPU：中央处理器，是计算机硬件的核心部件，由运算器和控制器组成；（早期的运算器和控制器不在同一芯片上，现在的 CPU 内除含有运算器和控制器外还集成了 CACHE）。

主存：计算机中存放正在运行的程序 and 数据的存储器，为计算机的主要工作存储器，可随机存取；由存储体、各种逻辑部件及控制电路组成。

存储单元：可存放一个机器字并具有特定存储地址的存储单位。

存储元件：存储一位二进制信息的物理元件，是存储器中最小的存储单位，又叫存储基元或存储元，不能单独存取。

存储字：一个存储单元所存二进制代码的逻辑单位。

存储字长：一个存储单元所存储的二进制代码的总位数。

存储容量：存储器中可存二进制代码的总量；（通常主、辅存容量分开描述）。

机器字长：指 CPU 一次能处理的二进制数据的位数，通常与 CPU 的寄存器位数有关。

指令字长：机器指令中二进制代码的总位数。

8. 解释下列英文缩写的中文含义：

CPU、PC、IR、CU、ALU、ACC、MQ、X、MAR、MDR、I/O、MIPS、CPI、FLOPS

解：全面的回答应分英文全称、中文名、功能三部分。

CPU: Central Processing Unit, 中央处理机（器），是计算机硬件的核心部件，主要由运算器和控制器组成。

PC: Program Counter, 程序计数器，其功能是存放当前欲执行指令的地址，并可自动计数形成下一条指令地址。

IR: Instruction Register, 指令寄存器，其功能是存放当前正在执行的指令。

CU: Control Unit, 控制单元（部件），为控制器的核心部件，其功能是产生微操作命令序列。

ALU: Arithmetic Logic Unit, 算术逻辑运算单元，为运算器的核心部件，其功能是进行算术、逻辑运算。

ACC: Accumulator, 累加器，是运算器中既能存放运算前的操作数，又能存放运算结果的寄存器。

MQ: Multiplier-Quotient Register, 乘商寄存器，乘法运算时存放乘数、除法时存放商的寄存器。

X: 此字母没有专指的缩写含义，可以用作任一部件名，在此表示操作数寄存器，即运算器中工作寄存器之一，用来存放操作数；

MAR: Memory Address Register, 存储器地址寄存器，在主存中用来存放欲访问的存储单元的地址。

MDR: Memory Data Register, 存储器数据缓冲寄存器，在主存中用来存放从某单元读出、或要写入某存储单元的数据。

I/O: Input/Output equipment, 输入/输出设备，为输入设备和输出设备的总称，用于计算机内部和外界信息的转换与传送。

MIPS: Million Instruction Per Second, 每秒执行百万条指令数，为计算机运算速度指标的一种计量单位。

9. 画出主机框图，分别以存数指令“STAM”和加法指令“ADD M”（M 均为主存地址）为例，在图中按序标出完成该指令（包括取指令阶段）的信息流程（如→①）。假设主存容量为 256M*32 位，在指令字长、存储字长、机器字长相等的条件下，指出图中各寄存器的位数。

解：主机框图如 P13 图 1.11 所示。

（1）STAM 指令：PC→MAR, MAR→MM, MM→MDR, MDR→IR,

OP(IR)→CU, Ad(IR)→MAR, ACC→MDR, MAR→MM,
WR

(2) ADD M 指令: PC→MAR, MAR→MM, MM→MDR, MDR→IR,
OP(IR)→CU, Ad(IR)→MAR, RD, MM→MDR,
MDR→X, ADD, ALU→ACC, ACC→MDR, WR

假设主存容量 256M*32 位, 在指令字长、存储字长、机器字长相等的条件下, ACC、X、IR、MDR 寄存器均为 32 位, PC 和 MAR 寄存器均为 28 位。

10. 指令和数据都存于存储器中, 计算机如何区分它们?

解: 计算机区分指令和数据有以下 2 种方法:

- 通过不同的时间段来区分指令和数据, 即在取指令阶段(或取指微程序)取出的为指令, 在执行指令阶段(或相应微程序)取出的即为数据。

- 通过地址来源区分, 由 PC 提供存储单元地址的取出的是指令, 由指令地址码部分提供存储单元地址的取出的是操作数。

第 2 章 计算机的发展及应用

1. 通常计算机的更新换代以什么为依据？

答：P22

主要以组成计算机基本电路的元器件为依据，如电子管、晶体管、集成电路等。

2. 举例说明专用计算机和通用计算机的区别。

答：按照计算机的效率、速度、价格和运行的经济性和实用性可以将计算机划分为通用计算机和专用计算机。通用计算机适应性强，但牺牲了效率、速度和经济性，而专用计算机是最有效、最经济和最快的计算机，但适应性很差。例如个人电脑和计算器。

3. 什么是摩尔定律？该定律是否永远生效？为什么？

答：P23，否，P36

第3章 系统总线

1. 什么是总线？总线传输有何特点？为了减轻总线负载，总线上的部件应具备什么特点？

答：P41.总线是一种能由多个部件分时共享的公共信息传送线路。

总线传输的特点是：某一时刻只允许有一个部件向总线发送信息，但多个部件可以同时从总线上接收相同的信息。

为了减轻总线负载，总线上的部件应通过三态驱动缓冲电路与总线连通。

2. 总线如何分类？什么是系统总线？系统总线又分为几类，它们各有何作用，是单向的，还是双向的，它们与机器字长、存储字长、存储单元有何关系？

答：按照连接部件的不同，总线可以分为片内总线、系统总线和通信总线。

系统总线是连接 CPU、主存、I/O 各部件之间的信息传输线。

系统总线按照传输信息不同又分为地址线、数据线和控制线。地址线是单向的，其根数越多，寻址空间越大，即 CPU 能访问的存储单元的个数越多；数据线是双向的，其根数与存储字长相同，是机器字长的整数倍。

3. 常用的总线结构有几种？不同的总线结构对计算机的性能有什么影响？举例说明。

答：略。见 P52-55。

4. 为什么要设置总线判优控制？常见的集中式总线控制有几种？各有何特点？哪种方式响应时间最快？哪种方式对电路故障最敏感？

答：总线判优控制解决多个部件同时申请总线时的使用权分配问题；

常见的集中式总线控制有三种：链式查询、计数器定时查询、独立请求；

特点：链式查询方式连线简单，易于扩充，对电路故障最敏感；计数器定时查询方式优先级设置较灵活，对故障不敏感，连线及控制过程较复杂；独立请求方式速度最快，但硬件器件用量大，连线多，成本较高。

5. 解释下列概念：总线宽度、总线带宽、总线复用、总线的主设备（或主模块）、总线的从设备（或从模块）、总线的传输周期和总线的通信控制。

答：P46。

总线宽度：通常指数据总线的根数；

总线带宽：总线的数据传输率，指单位时间内总线上传输数据的位数；

总线复用：指同一条信号线可以分时传输不同的信号。

总线的主设备（主模块）：指一次总线传输期间，拥有总线控制权的设备（模块）；

总线的从设备（从模块）：指一次总线传输期间，配合主设备完成数据传输的设备（模块），它只能被动接受主设备发来的命令；

总线的传输周期：指总线完成一次完整而可靠的传输所需时间；

总线的通信控制：指总线传送过程中双方的时间配合方式。

6. 试比较同步通信和异步通信。

答：同步通信：指由统一时钟控制的通信，控制方式简单，灵活性差，当系统中

各部件工作速度差异较大时，总线工作效率明显下降。适合于速度差别不大的场合。

异步通信：指没有统一时钟控制的通信，部件间采用应答方式进行联系，控制方式较同步复杂，灵活性高，当系统中各部件工作速度差异较大时，有利于提高总线工作效率。

7. 画图说明异步通信中请求与回答有哪几种互锁关系？

答：见 P61-62，图 3.86。

8. 为什么说半同步通信同时保留了同步通信和异步通信的特点？

答：半同步通信既能像同步通信那样由统一时钟控制，又能像异步通信那样允许传输时间不一致，因此工作效率介于两者之间。

9. 分离式通讯有何特点，主要用于什么系统？

答：分离式通讯的特点是：（1）各模块欲占用总线使用权都必须提出申请；（2）在得到总线使用权后，主模块在先定的时间内向对方传送信息，采用同步方式传送，不再等待对方的回答信号；（3）各模块在准备数据的过程中都不占用总线，使总线可接受其它模块的请求；（4）总线被占用时都在做有效工作，或者通过它发送命令，或者通过它传送数据，不存在空闲等待时间，充分利用了总线的占用，从而实现了总线在多个主、从模块间进行信息交叉重叠并行传送。

分离式通讯主要用于大型计算机系统。

10. 为什么要设置总线标准？你知道目前流行的总线标准有哪些？什么叫 plug and play？哪些总线有这一特点？

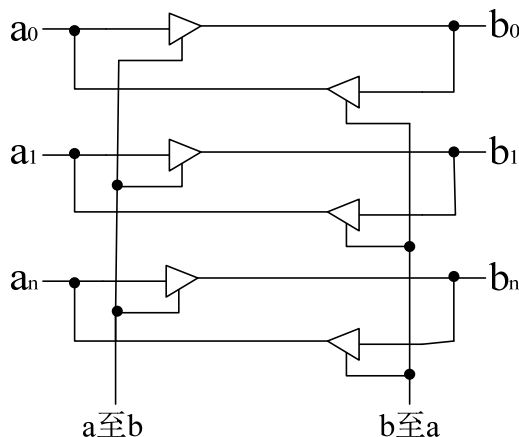
答：总线标准的设置主要解决不同厂家各类模块化产品的兼容问题；

目前流行的总线标准有：ISA、EISA、PCI 等；

plug and play：即插即用，EISA、PCI 等具有此功能。

11. 画一个具有双向传输功能的总线逻辑图。

答：在总线的两端分别配置三态门，就可以使总线具有双向传输功能。



12. 设数据总线上接有 A、B、C、D 四个寄存器，要求选用合适的 74 系列芯片，完成下列逻辑设计：

- (1) 设计一个电路，在同一时间实现 $D \rightarrow A$ 、 $D \rightarrow B$ 和 $D \rightarrow C$ 寄存器间的传送；
- (2) 设计一个电路，实现下列操作：
 - T0 时刻完成 $D \rightarrow$ 总线；
 - T1 时刻完成 总线 $\rightarrow A$ ；
 - T2 时刻完成 $A \rightarrow$ 总线；
 - T3 时刻完成 总线 $\rightarrow B$ 。

解：(1) 由 T 打开三态门将 D 寄存器中的内容送至总线 bus，由 cp 脉冲同时将总线上的数据打入到 A、B、C 寄存器中。T 和 cp 的时间关系如图 (1) 所示。

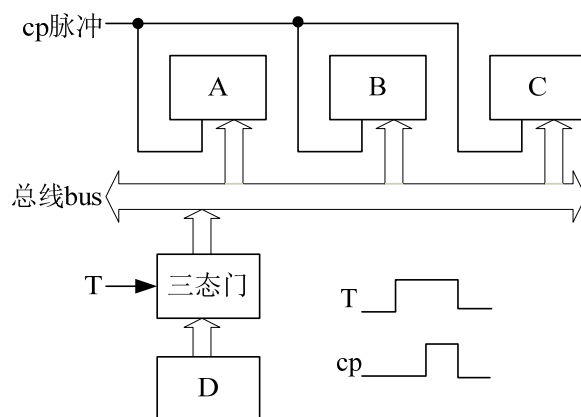
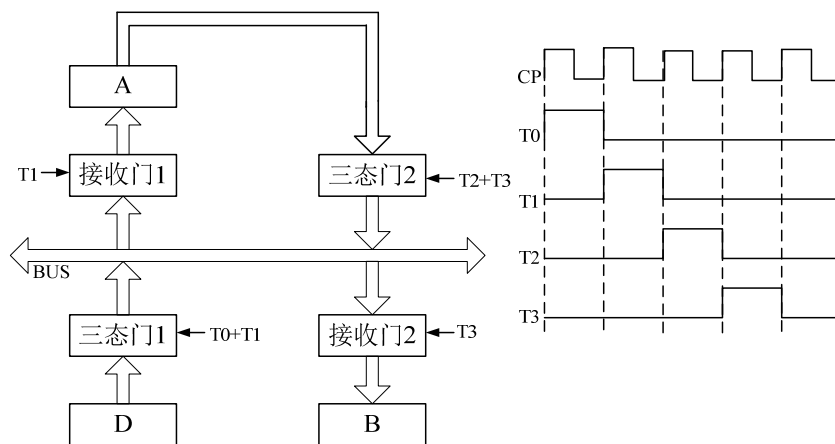


图 (1)

(2) 三态门 1 受 $T_0 + T_1$ 控制，以确保 T0 时刻 $D \rightarrow$ 总线，以及 T1 时刻总线 \rightarrow 接收门 1 $\rightarrow A$ 。三态门 2 受 $T_2 + T_3$ 控制，以确保 T2 时刻 $A \rightarrow$ 总线，以及 T3 时刻总线 \rightarrow 接收门 2 $\rightarrow B$ 。T0、T1、T2、T3 波形图如图 (2) 所示。



图(2)

13. 什么是总线的数据传输率，它与哪些因素有关？

答：总线数据传输率即总线带宽，指单位时间内总线上传输数据的位数，通常用每秒传输信息的字节数来衡量。它与总线宽度和总线频率有关，总线宽度越宽，频率越快，数据传输率越高。

14. 设总线的时钟频率为 8MHZ，一个总线周期等于一个时钟周期。如果一个总线周期中并行传送 16 位数据，试问总线的带宽是多少？

解：由于： $f=8\text{MHz}$, $T=1/f=1/8\text{M}$ 秒，一个总线周期等于一个时钟周期

所以：总线带宽= $16 / (1/8M) = 128Mbps$

15. 在一个 32 位的总线系统中，总线的时钟频率为 66MHZ，假设总线最短传输周期为 4 个时钟周期，试计算总线的最大数据传输率。若想提高数据传输率，可采取什么措施？

解：总线传输周期= $4 * 1/66M$ 秒

总线的最大数据传输率= $32 / (4/66M) = 528Mbps$

若想提高数据传输率，可以提高总线时钟频率、增大总线宽度或者减少总线传输周期包含的时钟周期个数。

16. 在异步串行传送系统中，字符格式为：1 个起始位、8 个数据位、1 个校验位、2 个终止位。若要求每秒传送 120 个字符，试求传送的波特率和比特率。

解：一帧包含： $1+8+1+2=12$ 位

故波特率为： $(1+8+1+2) * 120 = 1440bps$

比特率为： $8 * 120 = 960bps$

第4章 存储器

1. 解释概念：主存、辅存、Cache、RAM、SRAM、DRAM、ROM、PROM、EPROM、EEPROM、CDROM、Flash Memory。

答：主存：主存储器，用于存放正在执行的程序和数据。CPU 可以直接进行随机读写，访问速度较高。

辅存：辅助存储器，用于存放当前暂不执行的程序和数据，以及一些需要永久保存的信息。

Cache：高速缓冲存储器，介于 CPU 和主存之间，用于解决 CPU 和主存之间速度不匹配问题。

RAM：半导体随机存取存储器，主要用作计算机中的主存。

SRAM：静态半导体随机存取存储器。

DRAM：动态半导体随机存取存储器。

ROM：掩膜式半导体只读存储器。由芯片制造商在制造时写入内容，以后只能读出而不能写入。

PROM：可编程只读存储器，由用户根据需要确定写入内容，只能写入一次。

EPROM：紫外线擦写可编程只读存储器。需要修改内容时，现将其全部内容擦除，然后再编程。擦除依靠紫外线使浮动栅极上的电荷泄露而实现。

EEPROM：电擦写可编程只读存储器。

CDROM：只读型光盘。

Flash Memory：闪速存储器。或称快擦型存储器。

2. 计算机中哪些部件可以用于存储信息？按速度、容量和价格/位排序说明。

答：计算机中寄存器、Cache、主存、硬盘可以用于存储信息。

按速度由高至低排序为：寄存器、Cache、主存、硬盘；

按容量由小至大排序为：寄存器、Cache、主存、硬盘；

按价格/位由高至低排序为：寄存器、Cache、主存、硬盘。

3. 存储器的层次结构主要体现在什么地方？为什么要分这些层次？计算机如何管理这些层次？

答：存储器的层次结构主要体现在 Cache-主存和主存-辅存这两个存储层次上。

Cache-主存层次在存储系统中主要对 CPU 访存起加速作用，即从整体运行的效果分析，CPU 访存速度加快，接近于 Cache 的速度，而寻址空间和位价却接近于主存。

主存-辅存层次在存储系统中主要起扩容作用，即从程序员的角度看，他所使用的存储器其容量和位价接近于辅存，而速度接近于主存。

综合上述两个存储层次的作用，从整个存储系统来看，就达到了速度快、容量大、位价低的优化效果。

主存与 CACHE 之间的信息调度功能全部由硬件自动完成。而主存与辅存层次的调度目前广泛采用虚拟存储技术实现，即将主存与辅存的一部分通过软硬结合的技术组成虚拟存储器，程序员可使用这个比主存实际空间（物理地址空间）大得多的虚拟地址空间（逻辑地址空间）编程，当程序运行时，再由软、硬件自动配合完成虚拟地址空间与主存实际物理空间的转换。因此，这两个层次上的调度或转换操作对于程序员来说都是透明的。

4. 说明存取周期和存取时间的区别。

解：存取周期和存取时间的主要区别是：存取时间仅为完成一次操作的时间，而存取周期不仅包含操作时间，还包含操作后线路的恢复时间。即：

$$\text{存取周期} = \text{存取时间} + \text{恢复时间}$$

5. 什么是存储器的带宽？若存储器的数据总线宽度为 32 位，存取周期为 200ns，则存储器的带宽是多少？

解：存储器的带宽指单位时间内从存储器进出信息的最大数量。

$$\text{存储器带宽} = 1/200\text{ns} \times 32 \text{ 位} = 160\text{M 位/秒} = 20\text{MB/秒} = 5\text{M 字/秒}$$

注意：字长 32 位，不是 16 位。（注： $1\text{ns}=10^{-9}\text{s}$ ）

6. 某机字长为 32 位，其存储容量是 64KB，按字编址它的寻址范围是多少？若主存以字节编址，试画出主存字地址和字节地址的分配情况。

解：存储容量是 64KB 时，按字节编址的寻址范围就是 64K，

如按字编址，其寻址范围为： $64\text{K} / (32/8) = 16\text{K}$

字节地址		字地址
0000H		0000H
0001H		
0002H		
0003H		
0004H		0001H
0005H		
0006H		
0007H		
0008H		0002H
0009H		

主存字地址和字节地址的分配情况：如图

7. 一个容量为 16K×32 位的存储器，其地址线和数据线的总和是多少？当选用下列不同规格的存储芯片时，各需要多少片？

1K×4 位，2K×8 位，4K×4 位，16K×1 位，4K×8 位，8K×8 位

解：地址线和数据线的总和 = $14 + 32 = 46$ 根；

选择不同的芯片时，各需要的片数为：

$$1\text{K} \times 4: (16\text{K} \times 32) / (1\text{K} \times 4) = 16 \times 8 = 128 \text{ 片}$$

$$2\text{K} \times 8: (16\text{K} \times 32) / (2\text{K} \times 8) = 8 \times 4 = 32 \text{ 片}$$

$$4\text{K} \times 4: (16\text{K} \times 32) / (4\text{K} \times 4) = 4 \times 8 = 32 \text{ 片}$$

$$16\text{K} \times 1: (16\text{K} \times 32) / (16\text{K} \times 1) = 1 \times 32 = 32 \text{ 片}$$

$$4\text{K} \times 8: (16\text{K} \times 32) / (4\text{K} \times 8) = 4 \times 4 = 16 \text{ 片}$$

$$8\text{K} \times 8: (16\text{K} \times 32) / (8\text{K} \times 8) = 2 \times 4 = 8 \text{ 片}$$

8. 试比较静态 RAM 和动态 RAM。

答：略。（参看课件）

9. 什么叫刷新？为什么要刷新？说明刷新有几种方法。

解：刷新：对 DRAM 定期进行的全部重写过程；

刷新原因：因电容泄漏而引起的 DRAM 所存信息的衰减需要及时补充，因此安排了定期刷新操作；

常用的刷新方法有三种：集中式、分散式、异步式。

集中式：在最大刷新闻隔时间内，集中安排一段时间进行刷新，存在 CPU 访存死时间。

分散式：在每个读/写周期之后插入一个刷新周期，无 CPU 访存死时间。

异步式：是集中式和分散式的折衷。

10. 半导体存储器芯片的译码驱动方式有几种？

解：半导体存储器芯片的译码驱动方式有两种：线选法和重合法。

线选法：地址译码信号只选中同一个字的所有位，结构简单，费器材；

重合法：地址分行、列两部分译码，行、列译码线的交叉点即为所选单元。这种方法通过行、列译码信号的重合来选址，也称矩阵译码。可大大节省器材用量，是最常用的译码驱动方式。

11. 一个 $8K \times 8$ 位的动态 RAM 芯片，其内部结构排列成 256×256 形式，存取周期为 $0.1\mu s$ 。试问采用集中刷新、分散刷新和异步刷新三种方式的刷新闻隔各为多少？

解：采用集中刷新方式刷新闻隔为:2ms，其中刷新死时间为： $256 \times 0.1\mu s = 25.6\mu s$

采用分散刷新方式刷新闻隔为： $256 \times (0.1\mu s + 0.1\mu s) = 51.2\mu s$

采用异步刷新方式刷新闻隔为:2ms

12. 画出用 1024×4 位的存储芯片组成一个容量为 $64K \times 8$ 位的存储器逻辑框图。要求将 $64K$ 分成 4 个页面，每个页面分 16 组，指出共需多少片存储芯片。

解：设采用 SRAM 芯片，则：

总片数 = $(64K \times 8 \text{ 位}) / (1024 \times 4 \text{ 位}) = 64 \times 2 = 128$ 片

题意分析：本题设计的存储器结构上分为总体、页面、组三级，因此画图时也应分三级画。首先应确定各级的容量：

页面容量 = 总容量 / 页面数 = $64K \times 8 / 4 = 16K \times 8$ 位，4 片 $16K \times 8$ 字串联成 $64K \times 8$ 位

组容量 = 页面容量 / 组数 = $16K \times 8 \text{ 位} / 16 = 1K \times 8$ 位，16 片 $1K \times 8$ 位字串联成 $16K \times 8$ 位

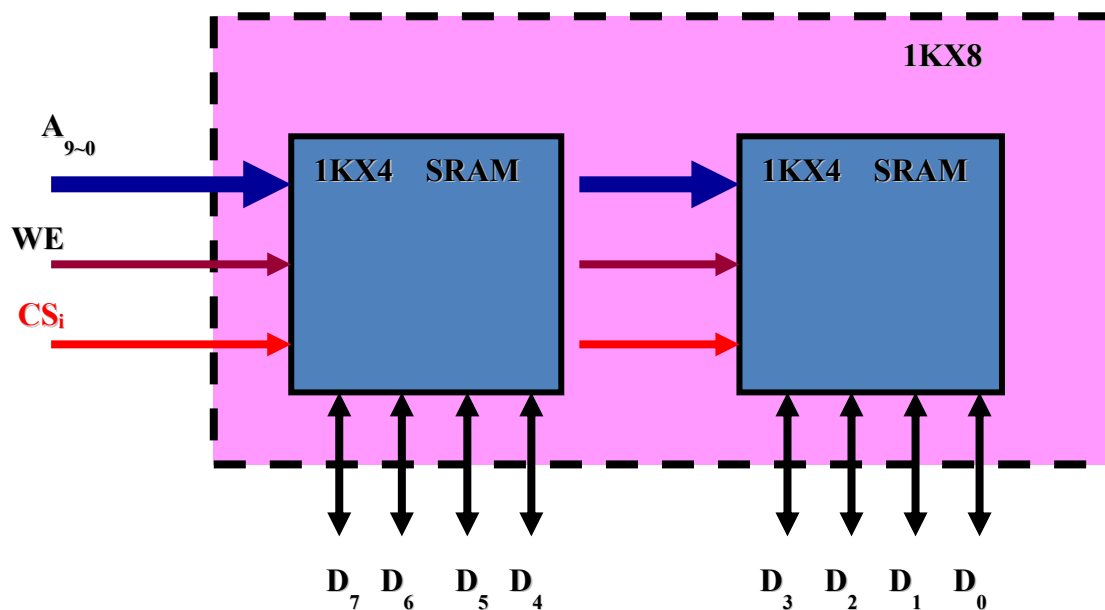
组内片数 = 组容量 / 片容量 = $1K \times 8 \text{ 位} / 1K \times 4 \text{ 位} = 2$ 片，两片 $1K \times 4$ 位芯片位并联成 $1K \times 8$ 位

存储器逻辑框图：

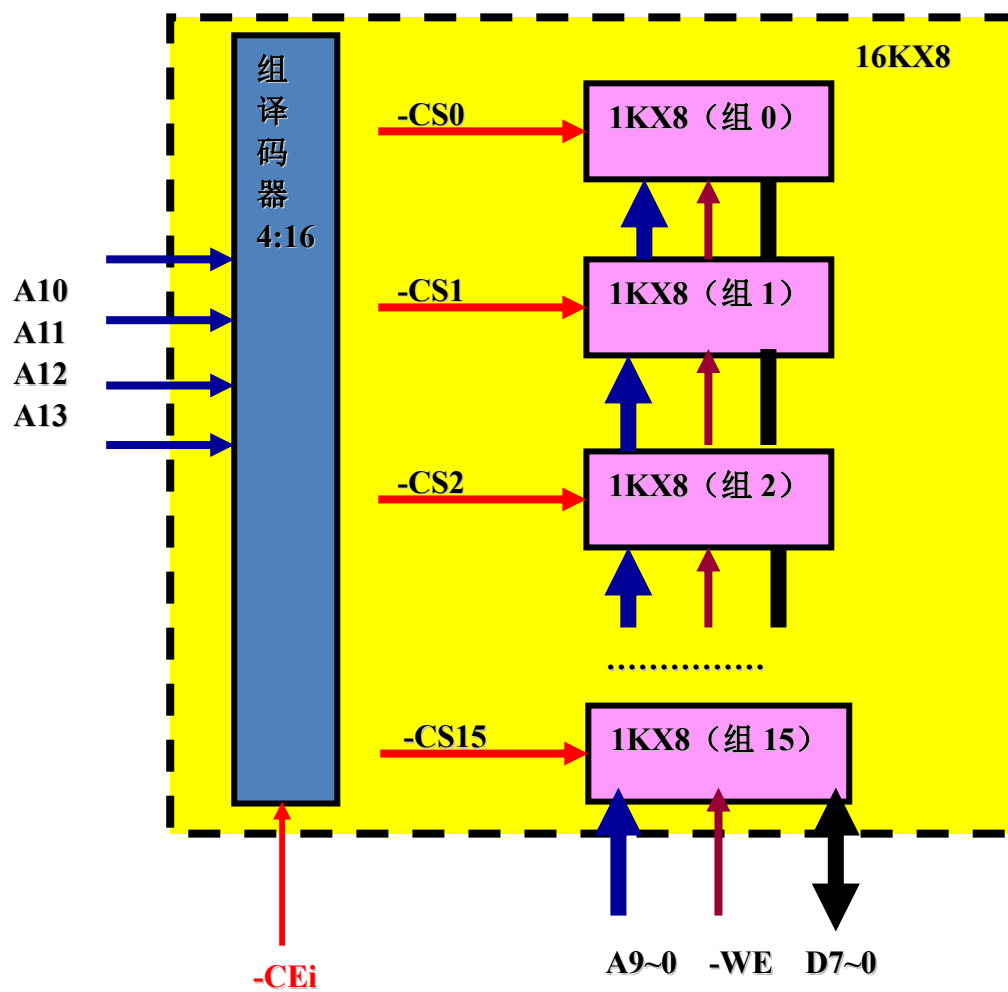
地址分配：



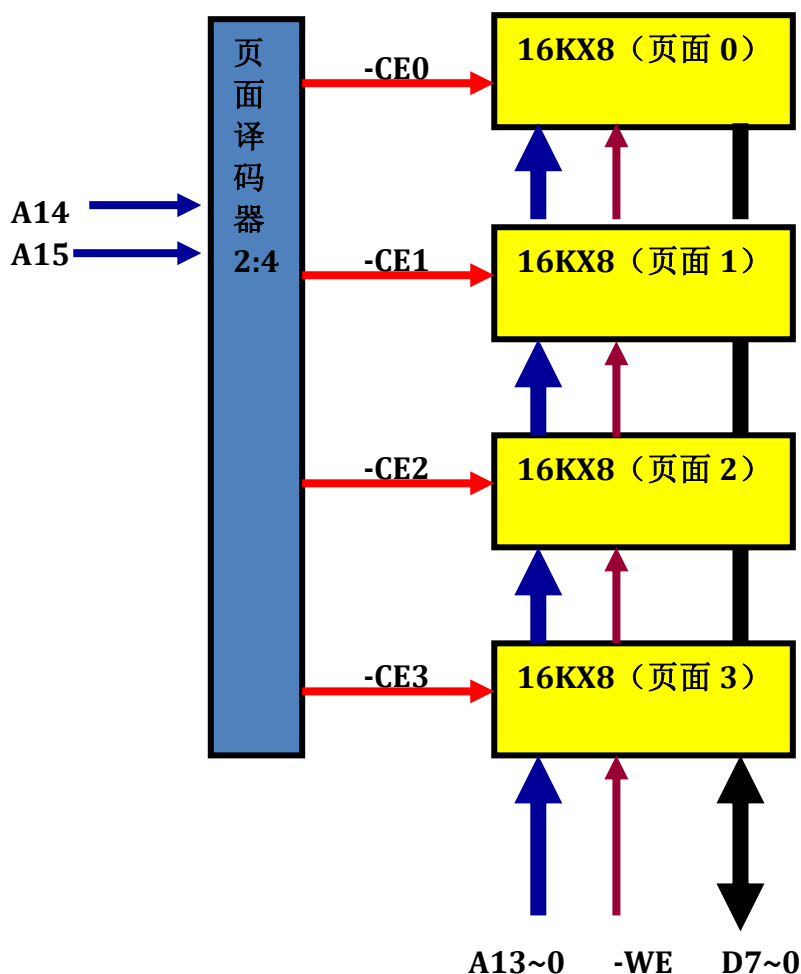
组逻辑图如下：（位扩展）



页面逻辑框图：（字扩展）



存储器逻辑框图：（字扩展）



13. 设有一个 $64\text{K} \times 8$ 位的 RAM 芯片，试问该芯片共有多少个基本单元电路（简称存储基元）？欲设计一种具有上述同样多存储基元的芯片，要求对芯片字长的选择应满足地址线和数据线的总和为最小，试确定这种芯片的地址线 and 数据线，并说明有几种解答。

解：存储基元总数 = $64\text{K} \times 8$ 位 = 512K 位 = 2^{19} 位；

思路：如要满足地址线和数据线总和最小，应尽量把存储元安排在字向，因为地址位数和字数成 2 的幂的关系，可较好地压缩线数。

设地址线根数为 a ，数据线根数为 b ，则片容量为： $2^a \times b = 2^{19}$ ； $b = 2^{19-a}$ ；

若 $a = 19$ ， $b = 1$ ，总和 = $19 + 1 = 20$ ；

$a = 18$ ， $b = 2$ ，总和 = $18 + 2 = 20$ ；

$a = 17$ ， $b = 4$ ，总和 = $17 + 4 = 21$ ；

$a = 16$ ， $b = 8$ ，总和 = $16 + 8 = 24$ ；

.....

由上可看出：芯片字数越少，芯片字长越长，引脚数越多。芯片字数减 1、芯片位数均按 2 的幂变化。

结论：如果满足地址线和数据线的总和为最小，这种芯片的引脚分配方案有两种：地址线 = 19 根，数据线 = 1 根；或地址线 = 18 根，数据线 = 2 根。

- (1) 该机所允许的最大主存空间是多少?
- (2) 若每个模块板为 $32\text{K} \times 8$ 位, 共需几个模块板?
- (3) 每个模块板内共有几片 RAM 芯片?
- (4) 共有多少片 RAM?
- (5) CPU 如何选择各模块板?

模板号 (3位)	芯片号 (3位)	片内地址 (12位)
----------	----------	------------

解：(1) 地址空间分配图：

系统程序区 (ROM 共 4KB)：0000H-0FFFH

用户程序区 (RAM 共 12KB)：1000H-3FFFH

(2) 选片：ROM：选择 4K×4 位芯片 2 片，位并联

RAM：选择 4K×8 位芯片 3 片，字串联 (RAM1 地址范围为：1000H-1FFFH, RAM2 地址范围为 2000H-2FFFH, RAM3 地址范围为：3000H-3FFFH)

[illegible]

CPU 和存储器连接逻辑图及片选逻辑如下图(3)所示:

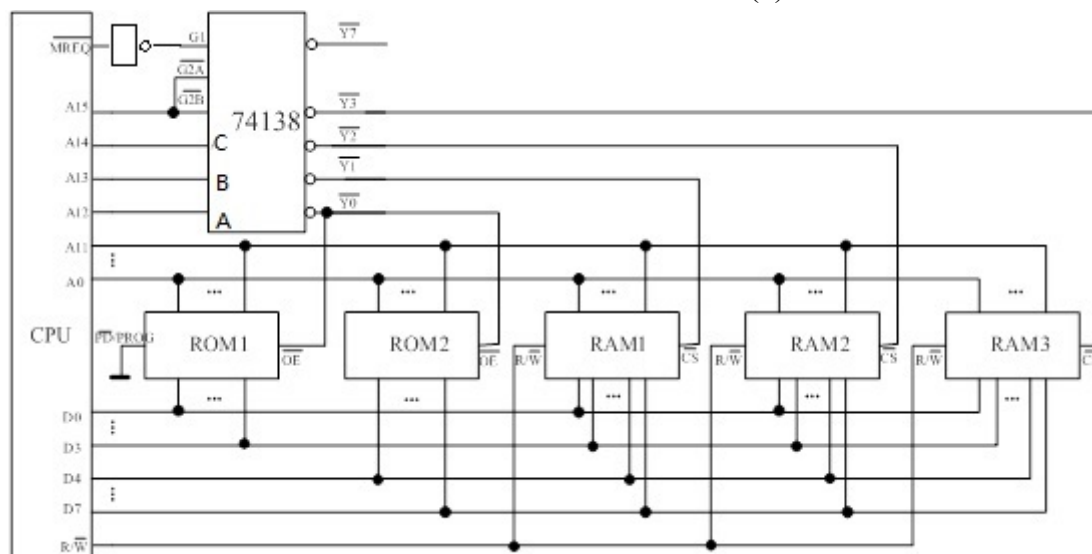


图 (3)

16. CPU 假设同上题, 现有 8 片 $8K \times 8$ 位的 RAM 芯片与 CPU 相连, 试回答:

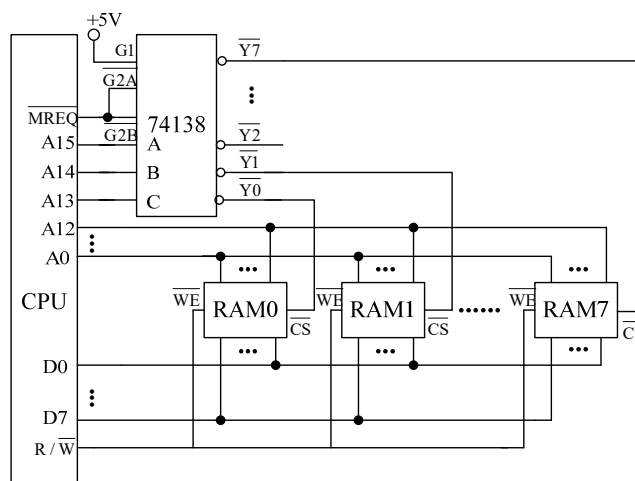
(1) 用 74138 译码器画出 CPU 与存储芯片的连接图;

(2) 写出每片 RAM 的地址范围;

(3) 如果运行时发现不论往哪片 RAM 写入数据后, 以 A000H 为起始地址的存储芯片都有与其相同的数据, 分析故障原因。

(4) 根据 (1) 的连接图, 若出现地址线 A13 与 CPU 断线, 并搭接到高电平上, 将出现什么后果?

解: (1) CPU 与存储器芯片连接逻辑图:



(2) 地址空间分配图:

RAM0: 0000H——1FFFH

RAM1: 2000H——3FFFH

RAM2: 4000H——5FFFH

RAM3: 6000H——7FFFH

RAM4: 8000H——9FFFH

RAM5: A000H——BFFFH

RAM6: C000H——DFFFH

RAM7: E000H——FFFFH

(3) 如果运行时发现不论往哪片 RAM 写入数据后, 以 A000H 为起始地址的存储芯片(RAM5)都有与其相同的数据, 则根本的故障原因为: 该存储芯片的片选输入端很可能总是处于低电平。假设芯片与译码器本身都是好的, 可能的情况有:

- 1) 该片的 \overline{CS} 端与 \overline{WE} 端错连或短路;
- 2) 该片的 \overline{CS} 端与 CPU 的 \overline{MREQ} 端错连或短路;
- 3) 该片的 \overline{CS} 端与地线错连或短路。

(4) 如果地址线 A13 与 CPU 断线, 并搭接到高电平上, 将会出现 A13 恒为“1”的情况。此时存储器只能寻址 A13=1 的地址空间(奇数片), A13=0 的另一半地址空间(偶数片)将永远访问不到。若对 A13=0 的地址空间(偶数片)进行访问, 只能错误地访问到 A13=1 的对应空间(奇数片)中去。

17. 写出 1100、1101、1110、1111 对应的汉明码。

解: 有效信息均为 $n=4$ 位, 假设有效信息用 $b_4b_3b_2b_1$ 表示

校验位位数 $k=3$ 位, ($2^k \geq n+k+1$)

设校验位分别为 c_1 、 c_2 、 c_3 , 则汉明码共 $4+3=7$ 位, 即: $c_1c_2b_4c_3b_3b_2b_1$

校验位在汉明码中分别处于第 1、2、4 位

$$c_1 = b_4 \oplus b_3 \oplus b_1$$

$$c_2 = b_4 \oplus b_2 \oplus b_1$$

$$c_3 = b_3 \oplus b_2 \oplus b_1$$

当有效信息为 1100 时, $c_3c_2c_1=110$, 汉明码为 0111100。

当有效信息为 1101 时, $c_3c_2c_1=001$, 汉明码为 1010101。

当有效信息为 1110 时, $c_3c_2c_1=000$, 汉明码为 0010110。

当有效信息为 1111 时, $c_3c_2c_1=111$, 汉明码为 1111111。

18. 已知收到的汉明码(按配偶原则配置)为 1100100、1100111、1100000、1100001, 检查上述代码是否出错? 第几位出错?

解: 假设接收到的汉明码为: $c_1'c_2'b_4'c_3'b_3'b_2'b_1'$

纠错过程如下:

$$P_1 = c_1' \oplus b_4' \oplus b_3' \oplus b_1'$$

$$P_2 = c_2' \oplus b_4' \oplus b_2' \oplus b_1'$$

$$P_3 = c_3' \oplus b_3' \oplus b_2' \oplus b_1'$$

如果收到的汉明码为 1100100, 则 $p_3p_2p_1=011$, 说明代码有错, 第 3 位(b_4')出错, 有效信息为: 1100

如果收到的汉明码为 1100111, 则 $p_3p_2p_1=111$, 说明代码有错, 第 7 位(b_1')出错, 有效信息为: 0110

如果收到的汉明码为 1100000, 则 $p_3p_2p_1=110$, 说明代码有错, 第 6 位(b_2')出错, 有效信息为: 0010

如果收到的汉明码为 1100001, 则 $p_3p_2p_1=001$, 说明代码有错, 第 1 位(c_1')出错, 有效信息为: 0001

19. 已经接收到下列汉明码，分别写出它们所对应的欲传送代码。

(1) 1100000 (按偶性配置)

(2) 1100010 (按偶性配置)

(3) 1101001 (按偶性配置)

(4) 0011001 (按奇性配置)

(5) 1000000 (按奇性配置)

(6) 1110001 (按奇性配置)

解：(一) 假设接收到的汉明码为 $C1'C2'B4'C3'B3'B2'B1'$ ，按偶性配置则：

$$P1=C1' \oplus B4' \oplus B3' \oplus B1'$$

$$P2=C2' \oplus B4' \oplus B2' \oplus B1'$$

$$P3=C3' \oplus B3' \oplus B1'$$

(1) 如接收到的汉明码为 1100000，

$$P1=1 \oplus 0 \oplus 0 \oplus 0=1$$

$$P2=1 \oplus 0 \oplus 0 \oplus 0=1$$

$$P3=0 \oplus 0 \oplus 0=0$$

$P3P2P1=011$ ，第 3 位出错，可纠正为 1110000，故欲传送的信息为 1000。

(2) 如接收到的汉明码为 1100010，

$$P1=1 \oplus 0 \oplus 0 \oplus 0=1$$

$$P2=1 \oplus 0 \oplus 1 \oplus 0=0$$

$$P3=0 \oplus 0 \oplus 0=0$$

$P3P2P1=001$ ，第 1 位出错，可纠正为 0100010，故欲传送的信息为 0010。

(3) 如接收到的汉明码为 1101001，

$$P1=1 \oplus 0 \oplus 0 \oplus 1=0$$

$$P2=1 \oplus 0 \oplus 0 \oplus 1=0$$

$$P3=1 \oplus 0 \oplus 1=0$$

$P3P2P1=000$ ，传送无错，故欲传送的信息为 0001。

(二) 假设接收到的汉明码为 $C1'C2'B4'C3'B3'B2'B1'$ ，按奇性配置则：

$$P1=C1' \oplus B4' \oplus B3' \oplus B1' \oplus 1$$

$$P2=C2' \oplus B4' \oplus B2' \oplus B1' \oplus 1$$

$$P3=C3' \oplus B3' \oplus B1' \oplus 1$$

(4) 如接收到的汉明码为 0011001，

$$P1=0 \oplus 1 \oplus 0 \oplus 1 \oplus 1=1$$

$$P2=0 \oplus 1 \oplus 0 \oplus 1 \oplus 1=1$$

$$P3=1 \oplus 0 \oplus 1 \oplus 1=1$$

$P3P2P1=111$ ，第 7 位出错，可纠正为 0011000，故欲传送的信息为 1000。

(5) 如接收到的汉明码为 1000000，

$$P1=1 \oplus 0 \oplus 0 \oplus 0 \oplus 1=0$$

$$P2=0 \oplus 1 \oplus 0 \oplus 0 \oplus 1=0$$

$$P3=0 \oplus 0 \oplus 0 \oplus 1=1$$

$P3P2P1=100$ ，第 4 位出错，可纠正为 1001000，故欲传送的信息为 0000。

(6) 如接收到的汉明码为 1110001，

$$P1=1 \oplus 1 \oplus 0 \oplus 1 \oplus 1=0$$

$$P2=1 \oplus 1 \oplus 0 \oplus 1 \oplus 1=0$$

$$P3=0 \oplus 0 \oplus 1 \oplus 1=0$$

$P_3P_2P_1=000$ ，传送无错，故欲传送的信息为 1001。

20. 欲传送的二进制代码为 1001101，用奇校验来确定其对应的汉明码，若在第 6 位出错，说明纠错过程。

解：欲传送的二进制代码为 1001101，有效信息位数为 $n=7$ 位，则汉明校验的校验位为 k 位，则： $2k \geq n+k+1$ ， $k=4$ ，进行奇校验设校验位为 $C_1C_2C_3C_4$ ，汉明码为 $C_1C_2B_7C_3B_6B_5B_4C_4B_3B_2B_1$ ，

$$C_1=1 \oplus B_7 \oplus B_6 \oplus B_4 \oplus B_3 \oplus B_1=1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1=1$$

$$C_2=1 \oplus B_7 \oplus B_5 \oplus B_4 \oplus B_2 \oplus B_1=1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1=0$$

$$C_3=1 \oplus B_6 \oplus B_5 \oplus B_4=1 \oplus 0 \oplus 0 \oplus 1=0$$

$$C_4=1 \oplus B_3 \oplus B_2 \oplus B_1=1 \oplus 1 \oplus 0 \oplus 1=1$$

故传送的汉明码为 10100011101，若第 6 位 (B_5) 出错，即接收的码字为 10100111101，则

$$P_1=1 \oplus C_1' \oplus B_7' \oplus B_6' \oplus B_4' \oplus B_3' \oplus B_1'=1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1=0$$

$$P_2=1 \oplus C_2' \oplus B_7' \oplus B_5' \oplus B_4' \oplus B_2' \oplus B_1'=1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1=1$$

$$P_3=1 \oplus C_3' \oplus B_6' \oplus B_5' \oplus B_4'=1 \oplus 0 \oplus 0 \oplus 1 \oplus 1=1$$

$$P_4=1 \oplus C_4' \oplus B_3' \oplus B_2' \oplus B_1'=1 \oplus 1 \oplus 1 \oplus 0 \oplus 1=0$$

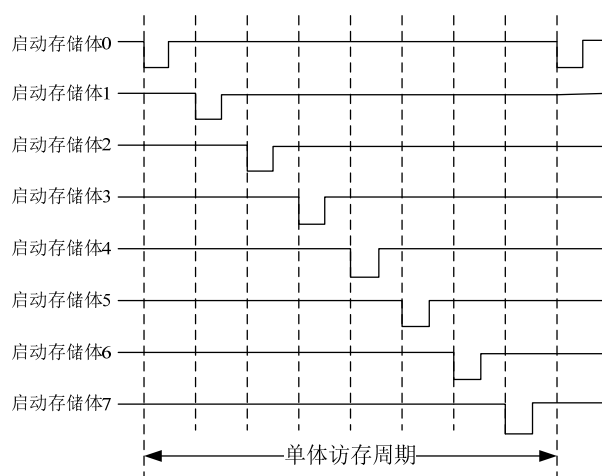
$P_4P_3P_2P_1=0110$ 说明第 6 位出错，对第 6 位取反即完成纠错。

21. 为什么在汉明码纠错过程中，新的检测位 $P_3P_2P_1$ 的状态即指出了编码中错误的信息位？

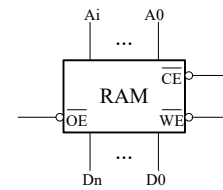
答：汉明码属于分组奇偶校验， $P_3P_2P_1=000$ ，说明接收方生成的校验位和收到的校验位相同，否则不同说明出错。由于分组时校验位只参加一组奇偶校验，有效信息参加至少两组奇偶校验，若果校验位出错， $P_3P_2P_1$ 的某一位将为 1，刚好对应位号 4、2、1；若果有效信息出错，将引起 $P_3P_2P_1$ 中至少两位为 1，如 B_1 出错，将使 P_3P_1 均为 1， $P_2=0$ ， $P_4P_2P_1=101$ ，

22. 某机字长 16 位，常规的存储空间为 64K 字，若想不改用其他高速的存储芯片，而使访存速度提高到 8 倍，可采取什么措施？画图说明。

解：若想不改用高速存储芯片，而使访存速度提高到 8 倍，可采取八体交叉存取技术，8 体交叉访问时序如下图：



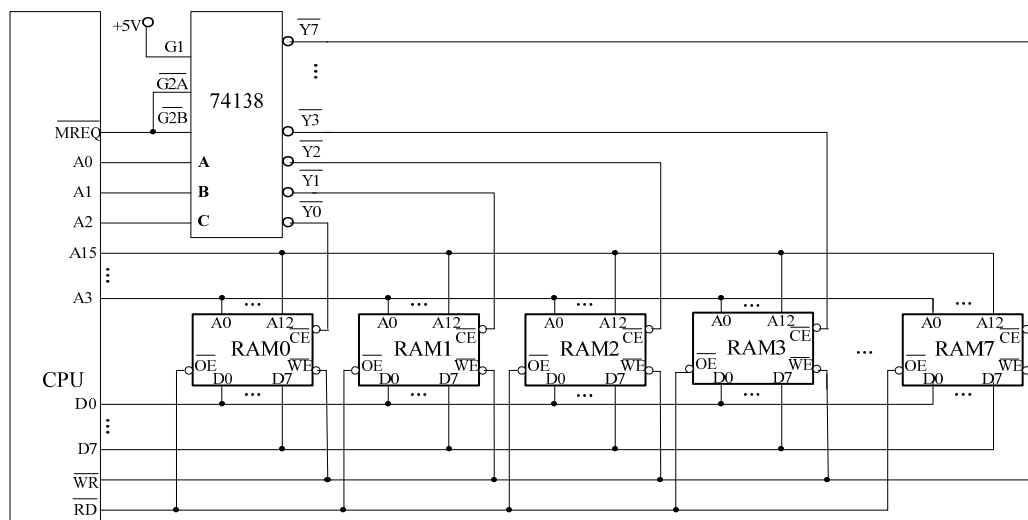
23. 设 CPU 共有 16 根地址线，8 根数据线，并用 M/\overline{IO} 作为访问存储器或 I/O 的控制信号（高电平为访存，低电平为访 I/O）， \overline{WR} （低电平有效）为写命令， \overline{RD} （低电平有效）为读命令。设计一个容量为 64KB 的采用低位交叉编址的 8 体并行结构存储器。现有下图所示的存储器芯片和 138 译码器。



行结构存储器。现有下图所示的存储器芯片和 138 译码器。

画出 CPU 和存储器芯片（芯片容量自定）的连接图，并写出图中每个存储芯片的地址范围（用十六进制数表示）。

解：8 体低位交叉并行存储器的每个存储体容量为 $64KB/8=8KB$ ，因此应选择 8KBRAM 芯片，芯片地址线 12 根（ A_0-A_{12} ），数据线 8 根（ D_0-D_7 ），用 138 译码器进行存储体的选择。设计如下：



24. 一个 4 体低位交叉的存储器，假设存储周期为 T ，CPU 每隔 $1/4$ 存取周期启动一个存储体，试问依次访问 64 个字需多少个存取周期？

解：4 体低位交叉的存储器的总线传输周期为 τ ， $\tau=T/4$ ，依次访问 64 个字所需时间为：

$$t=T+(64-1)\tau=T+63T/4=16.75T$$

25. 什么是“程序访问的局部性”？存储系统中哪一级采用了程序访问的局部性原理？

答：程序运行的局部性原理指：在一小段时间内，最近被访问过的程序和数据很可能再次被访问；在空间上，这些被访问的程序和数据往往集中在一小片存储区；在访问顺序上，指令顺序执行比转移执行的可能性大（大约 5:1）。存储系统中 Cache-主存层次和主存-辅存层次均采用了程序访问的局部性原理。

26. 计算机中设置 Cache 的作用是什么？能否将 Cache 的容量扩大，最后取代主存，为什么？

答：计算机中设置 Cache 的作用是解决 CPU 和主存速度不匹配问题。

不能将 Cache 的容量扩大取代主存，原因是：（1）Cache 容量越大成本越高，难以满足人们追求低价格的要求；（2）如果取消主存，当 CPU 访问 Cache 失败时，需要将辅存的内容调入 Cache 再由 CPU 访问，造成 CPU 等待时间太长，损失更大。

27. Cache 做在 CPU 芯片内有什么好处？将指令 Cache 和数据 Cache 分开又有什么好处？

答：Cache 做在 CPU 芯片内主要有下面几个好处：

（1）可提高外部总线的利用率。因为 Cache 在 CPU 芯片内，CPU 访问 Cache 时不必占用外部总线。

（2）Cache 不占用外部总线就意味着外部总线可更多地支持 I/O 设备与主存的信息传输，增强了系统的整体效率。

（3）可提高存取速度。因为 Cache 与 CPU 之间的数据通路大大缩短，故存取速度得以提高。

将指令 Cache 和数据 Cache 分开有如下好处：

1）可支持超前控制和流水线控制，有利于这类控制方式下指令预取操作的完成。

2）指令 Cache 可用 ROM 实现，以提高指令存取的可靠性。

3）数据 Cache 对不同数据类型的支持更为灵活，既可支持整数（例 32 位），也可支持浮点数据（如 64 位）。

补充：

Cache 结构改进的第三个措施是分级实现，如二级缓存结构，即在片内 Cache（L1）和主存之间再设一个片外 Cache（L2），片外缓存既可以弥补片内缓存容量不够大的缺点，又可在主存与片内缓存间起到平滑速度差的作用，加速片内缓存的调入调出速度。

28. 设主存容量为 256K 字，Cache 容量为 2K 字，块长为 4。

（1）设计 Cache 地址格式，Cache 中可装入多少块数据？

（2）在直接映射方式下，设计主存地址格式。

（3）在四路组相联映射方式下，设计主存地址格式。

（4）在全相联映射方式下，设计主存地址格式。

（5）若存储字长为 32 位，存储器按字节寻址，写出上述三种映射方式下主存的地址格式。

解：（1）Cache 容量为 2K 字，块长为 4，Cache 共有 $2K/4=2^{11}/2^2=2^9=512$ 块，

Cache 字地址 9 位，字块内地址为 2 位

因此，Cache 地址格式设计如下：

Cache 字块地址（9 位）	字块内地址（2 位）
-----------------	------------

（2）主存容量为 256K 字= 2^{18} 字，主存地址共 18 位，共分 $256K/4=2^{16}$ 块，主存字块标记为 $18-9-2=7$ 位。

直接映射方式下主存地址格式如下：

主存字块标记（7 位）	Cache 字块地址（9 位）	字块内地址（2 位）
-------------	-----------------	------------

(3)根据四路组相联的条件,一组内共有 4 块,得 Cache 共分为 $512/4=128=2^7$ 组,

主存字块标记为 $18-7-2=9$ 位,主存地址格式设计如下:

主存字块标记 (9 位)	组地址 (7 位)	字块内地址(2 位)
--------------	-----------	------------

(4) 在全相联映射方式下,主存字块标记为 $18-2=16$ 位,其地址格式如下:

主存字块标记 (16 位)	字块内地址 (2 位)
---------------	-------------

(5)若存储字长为 32 位,存储器按字节寻址,则主存容量为 $256K*32/8=2^{20}B$,
Cache 容量为 $2K*32/8=2^{13}B$,块长为 $4*32/8=16=2^4B$,字块内地址为 4 位,

在直接映射方式下,主存字块标记为 $21-9-5=7$ 位,主存地址格式为:

主存字块标记 (7 位)	Cache 字块地址 (9 位)	字块内地址(4 位)
--------------	------------------	------------

在四路组相联映射方式下,主存字块标记为 $21-7-5=9$ 位,主存地址格式为:

主存字块标记 (9 位)	组地址 (7 位)	字块内地址(4 位)
--------------	-----------	------------

在全相联映射方式下,主存字块标记为 $21-5=16$ 位,主存地址格式为:

主存字块标记 (16 位)	字块内地址 (4 位)
---------------	-------------

29. 假设 CPU 执行某段程序时共访问 Cache 命中 4800 次,访问主存 200 次,已知 Cache 的存取周期为 30ns,主存的存取周期为 150ns,求 Cache 的命中率以及 Cache-主存系统的平均访问时间和效率,试问该系统的性能提高了多少倍?

解: Cache 被访问命中率为: $4800/(4800+200)=24/25=96\%$

则 Cache-主存系统的平均访问时间为: $t_a=0.96*30ns+(1-0.96)*150ns=34.8ns$

Cache-主存系统的访问效率为: $e=t_c/t_a*100\%=30/34.8*100\%=86.2\%$

性能为原来的 $150ns/34.8ns=4.31$ 倍,即提高了 3.31 倍。

30. 一个组相连映射的 CACHE 由 64 块组成,每组内包含 4 块。主存包含 4096 块,每块由 128 字组成,访存地址为字地址。试问主存和高速存储器的地址各为几位?画出主存地址格式。

解: cache 组数: $64/4=16$, Cache 容量为: $64*128=2^{13}$ 字, cache 地址 13 位

主存共分 $4096/16=256$ 区,每区 16 块

主存容量为: $4096*128=2^{19}$ 字,主存地址 19 位,地址格式如下:

主存字块标记 (8 位)	组地址 (4 位)	字块内地址 (7 位)
--------------	-----------	-------------

31. 设主存容量为 1MB,采用直接映射方式的 Cache 容量为 16KB,块长为 4,每字 32 位。试问主存地址为 ABCDEH 的存储单元在 Cache 中的什么位置?

解: 主存和 Cache 按字节编址,

Cache 容量 $16KB=2^{14}B$ ，地址共格式为 14 位，分为 $16KB/(4*32/8B)=2^{10}$ 块，每块 $4*32/8=16B=2^4B$ ，Cache 地址格式为：

Cache 字块地址（10 位）	字块内地址（4 位）
------------------	------------

主存容量 $1MB=2^{20}B$ ，地址共格式为 20 位，分为 $1MB/(4*32/8B)=2^{16}$ 块，每块 2^4B ，采用直接映射方式，主存字块标记为 $20-14=6$ 位，主存地址格式为：

主存字块标记（6 位）	Cache 字块地址（10 位）	字块内地址（4 位）
-------------	------------------	------------

主存地址为 ABCDEH=1010 1011 1100 1101 1110B，主存字块标记为 101010，Cache 字块地址为 11 1100 1101，字块内地址为 1110，故该主存单元应映射到 Cache 的 101010 块的第 1110 字节，即第 42 块第 14 字节位置。或者在 Cache 的第 11 1100 1101 1110=3CDEH 字节位置。

32. 设某机主存容量为 4MB，Cache 容量为 16KB，每字块有 8 个字，每字 32 位，设计一个四路组相联映射（即 Cache 每组内共有 4 个字块）的 Cache 组织。

（1）画出主存地址字段中各段的位数。

（2）设 Cache 的初态为空，CPU 依次从主存第 0，1，2，...，89 号单元读出 90 个字（主存一次读出一个字），并重复按此次序读 8 次，问命中率是多少？

（3）若 Cache 的速度是主存的 6 倍，试问有 Cache 和无 Cache 相比，速度约提高多少倍？

解：（1）根据每字块有 8 个字，每字 32 位（4 字节），得出主存地址字段中字块内地址为 $3+2=5$ 位。

根据 Cache 容量为 $16KB=2^{14}B$ ，字块大小为 $8*32/8=32=2^5B$ ，得 Cache 地址共 14 位，Cache 共有 $2^{14-5}=2^9$ 块。

根据四路组相联映射，Cache 共分为 $2^9/2^2=2^7$ 组。

根据主存容量为 $4MB=2^{22}B$ ，得主存地址共 22 位，主存字块标记为 $22-7-5=10$ 位，故主存地址格式为：

主存字块标记（10 位）	组地址（7 位）	字块内地址（5 位）
--------------	----------	------------

（2）由于每个字块中有 8 个字，而且初态为空，因此 CPU 读第 0 号单元时，未命中，必须访问主存，同时将该字所在的主存块调入 Cache 第 0 组中的任一块内，接着 CPU 读第 1~7 号单元时均命中。同理，CPU 读第 8，16，...，88 号时均未命中。可见，CPU 在连续读 90 个字中共有 12 次未命中，而后 8 次循环读 90 个字全部命中，命中率为：

$$\frac{90 \times 8 - 12}{90 \times 8} = 0.984$$

（3）设 Cache 的周期为 t ，则主存周期为 $6t$ ，没有 Cache 的访问时间为 $6t \times 90 \times 8$ ，有 Cache 的访问时间为 $t(90 \times 8 - 12) + 6t \times 12$ ，则有 Cache 和无 Cache 相比，速度提高的倍数为：

$$\frac{6t \times 90 \times 8}{(90 \times 8 - 12)t + 6t \times 12} - 1 \approx 5.54$$

33. 简要说明提高访存速度可采取的措施。

答：提高访存速度可采取三种措施：

（1）采用高速器件。即采用存储周期短的芯片，可提高访存速度。

（2）采用 Cache。CPU 最近要使用的信息先调入 Cache，而 Cache 的速度比

主存快得多，这样 CPU 每次只需从 Cache 中读写信息，从而缩短访存时间，提高访存速度。

(3) 调整主存结构。如采用单体多字或采用多体结构存储器。

38. 磁盘组有 6 片磁盘，最外两侧盘面可以记录，存储区域内径 22cm，外径 33cm，道密度为 40 道/cm，内层密度为 400 位/cm，转速 3600 转/分，问：

- (1) 共有多少存储面可用？
- (2) 共有多少柱面？
- (3) 盘组总存储容量是多少？
- (4) 数据传输率是多少？

解：(1) 共有： $6 \times 2 = 12$ 个存储面可用。

(2) 有效存储区域 = $(33 - 22) / 2 = 5.5 \text{ cm}$

柱面数 = $40 \text{ 道/cm} \times 5.5 = 220 \text{ 道}$

(3) 内层道周长 = $\pi \times 22 = 69.08 \text{ cm}$

道容量 = $400 \text{ 位/cm} \times 69.08 \text{ cm} = 3454 \text{ B}$

面容量 = $3454 \text{ B} \times 220 \text{ 道} = 759,880 \text{ B}$

盘组总容量 = $759,880 \text{ B} \times 12 \text{ 面} = 9,118,560 \text{ B}$

(4) 转速 = $3600 \text{ 转} / 60 \text{ 秒} = 60 \text{ 转/秒}$

数据传输率 = $3454 \text{ B} \times 60 \text{ 转/秒} = 207,240 \text{ B/S}$

39. 某磁盘存储器转速为 3000 转/分，共有 4 个记录盘面，每毫米 5 道，每道记录信息 12 288 字节，最小磁道直径为 230mm，共有 275 道，求：

- (1) 磁盘存储器的存储容量。
- (2) 最高位密度（最小磁道的位密度）和最低位密度。
- (3) 磁盘数据传输率。
- (4) 平均等待时间。

解：(1) 存储容量 = $275 \text{ 道} \times 12\,288 \text{ B/道} \times 4 \text{ 面} = 13\,516\,800 \text{ B}$

(2) 最高位密度 = $12\,288 \text{ B} / (\pi \times 230) = 17 \text{ B/mm} = 136 \text{ 位/mm}$ （向下取整）

最大磁道直径 = $230 \text{ mm} + 2 \times 275 \text{ 道} / (5 \text{ 道/mm}) = 230 \text{ mm} + 110 \text{ mm} = 340 \text{ mm}$

最低位密度 = $12\,288 \text{ B} / (\pi \times 340) = 11 \text{ B/mm} = 92 \text{ 位/mm}$ （向下取整）

(3) 磁盘数据传输率 = $12\,288 \text{ B} \times 3000 \text{ 转/分} = 12\,288 \text{ B} \times 50 \text{ 转/秒} = 614\,400 \text{ B/s}$

(4) 平均等待时间 = $1 \text{ s} / 50 / 2 = 10 \text{ ms}$

第5章 输入输出系统

1. I/O 有哪些编址方式？各有何特点？

解：常用的 I/O 编址方式有两种：I/O 与内存统一编址和 I/O 独立编址。

特点：I/O 与内存统一编址方式的 I/O 地址采用与主存单元地址完全一样的格式，I/O 设备和主存占用同一个地址空间，CPU 可像访问主存一样访问 I/O 设备，不需要安排专门的 I/O 指令。

I/O 独立编址方式时机器为 I/O 设备专门安排一套完全不同于主存地址格式的编址方式，此时 I/O 地址与主存地址是两个独立的空间，CPU 需要通过专门的 I/O 指令来访问 I/O 地址空间。

2. 简要说明 CPU 与 I/O 之间传递信息可采用哪几种联络方式？它们分别用于什么场合？

答：CPU 与 I/O 之间传递信息常采用三种联络方式：直接控制（立即响应）、同步、异步。适用场合分别为：

直接控制适用于结构极简单、速度极慢的 I/O 设备，CPU 直接控制外设处于某种状态而无须联络信号。

同步方式采用统一的时钟进行联络，适用于 CPU 与 I/O 速度差并不大，近距离传送的场合。

异步方式采用应答机制进行联络，适用于 CPU 与 I/O 速度差较大、远距离传送的场合。

6. 字符显示器的接口电路中配有缓冲存储器和只读存储器，各有何作用？

解：显示缓冲存储器的作用是支持屏幕扫描时的反复刷新；只读存储器作为字符发生器使用，他起着将字符的 ASCII 码转换为字形点阵信息的作用。

8. 某计算机的 I/O 设备采用异步串行传送方式传送字符信息。字符信息的格式为 1 位起始位、7 位数据位、1 位校验位和 1 位停止位。若要求每秒钟传送 480 个字符，那么该设备的数据传送速率为多少？

解： $480 \times 10 = 4800$ 位/秒 = 4800 波特

波特——是数据传送速率波特率的单位。

10. 什么是 I/O 接口，与端口有何区别？为什么要设置 I/O 接口？I/O 接口如何分类？

解：I/O 接口一般指 CPU 和 I/O 设备间的连接部件，而端口是指 I/O 接口内 CPU 能够访问的寄存器，端口加上相应的控制逻辑即构成 I/O 接口。

I/O 接口分类方法很多，主要有：

（1）按数据传送方式分有并行接口和串行接口两种；

（2）按数据传送的控制方式分有程序控制接口、程序中断接口、DMA 接口三种。

12. 结合程序查询方式的接口电路，说明其工作过程。

解：程序查询接口工作过程如下（以输入为例）：

1) CPU 发 I/O 地址设备开始工作；地址总线 → 接口 → 设备选择器译码 → 选

中→发 SEL 信号； 2) CPU 发启动命令 DBR→开命令接收门； →D 置 0, B 置 1 → 接口向设备发启动命令； 3) CPU 等待，输入设备读出数据； 4) 外设工作完成, B 置 0, D 置 1； 5) 准备就绪信号→接口→完成信号→控制总线→CPU； 6) 输入：CPU 通过输入指令 (IN) 将 DBR 中的数据取走。

若为输出，除数据传送方向相反以外，其他操作与输入类似。工作过程如下：

开命令接收门；→选中，发 SEL 信号→设备选择器译码→接口→地址总线→1) CPU 发 I/O 地址 2) 输出：CPU 通过输出指令 (OUT) 将数据放入接口 DBR 中；设备开始工作；→接口向设备发启动命令→D 置 0, B 置 1 → 3) CPU 发启动命令 4) CPU 等待，输出设备将数据从 DBR 取走； B 置 0, D 置 1；→接口→ 5) 外设工作完成，完成信号 CPU，CPU 可通过指令再次向接口 DBR 输出数据，进行第二次传送。→控制总线→6) 准备就绪信号。

13. 说明中断向量地址和入口地址的区别和联系。

解：中断向量地址和入口地址的区别：

中断向量地址是硬件电路（向量编码器）产生的中断源的内存地址编号，即存放中断向量的地址编号；中断入口地址是中断服务程序首址，即中断向量。

中断向量地址和入口地址的联系：

中断向量地址可理解为中断服务程序入口地址指示器（入口地址的地址），通过它访存可获得中断服务程序入口地址。（两种方法：在向量地址所指单元内放一条 JMP 指令；主存中设向量地址表。参考 8.4.3）

14. 在什么条件下，I/O 设备可以向 CPU 提出中断请求？

解：I/O 设备向 CPU 提出中断请求的条件是：I/O 接口中的设备工作完成状态为 1 (D=1)，中断屏蔽码为 0 (MASK=0)，且 CPU 查询中断时，中断请求触发器状态为 1 (INTR=1)。

15. 什么是中断允许触发器？它有何作用？

解：中断允许触发器是 CPU 中断系统中的一个部件，他起着开关中断的作用（即中断总开关，则中断屏蔽触发器可视为中断的分开关）。

16. 在什么条件和什么时间，CPU 可以响应 I/O 的中断请求？

解：CPU 响应 I/O 中断请求的条件和时间是：当中断允许状态为 1 (EINT=1)，且至少有一个中断请求被查到，则在一条指令执行完时，响应中断。

17. 某系统对输入数据进行取样处理，每抽取一个输入数据，CPU 就要中断处理一次，将取样的数据存至存储器的缓冲区中，该中断处理需 P 秒。此外，缓冲区内每存储 N 个数据，主程序就要将其取出进行处理，这个处理需 Q 秒。试问该系统可以跟踪到每秒多少次中断请求？

解：这是一道求中断饱和度的题，要注意主程序对数据的处理不是中断处理，因此 Q 秒不能算在中断次数内。

N 个数据所需的处理时间= $P \times N + Q$ 秒

平均每个数据所需处理时间= $(P \times N + Q) / N$ 秒

求倒数得：该系统跟踪到的每秒中断请求数= $N / (P \times N + Q)$ 次。

19. 在程序中断方式中，磁盘申请中断的优先权高于打印机。当打印机正在进行打印时，磁盘申请中断请求。试问是否要将打印机输出停下来，等磁盘操作结束后，打印机输出才能继续进行？为什么？

解：这是一道多重中断的题，由于磁盘中断的优先权高于打印机，因此应将打印

机输出停下来，等磁盘操作结束后，打印机输出才能继续进行。因为打印机的速度比磁盘输入输出的速度慢，并且暂停打印不会造成数据丢失。

26. 什么是多重中断？实现多重中断的必要条件是什么？

解：多重中断是指：当 CPU 执行某个中断服务程序的过程中，发生了更高级、更紧迫的事件，CPU 暂停现行中断服务程序的执行，转去处理该事件的中断，处理完返回现行中断服务程序继续执行的过程。

实现多重中断的必要条件是：在现行中断服务期间，中断允许触发器为 1，即开中断。

28. CPU 对 DMA 请求和中断请求的响应时间是否一样？为什么？

解：CPU 对 DMA 请求和中断请求的响应时间不一样，因为两种方式的交换速度相差很大，因此 CPU 必须以更短的时间间隔查询并响应 DMA 请求。响应中断请求是在每条指令执行周期结束的时刻，而响应 DMA 请求是在存取周期结束的时刻。

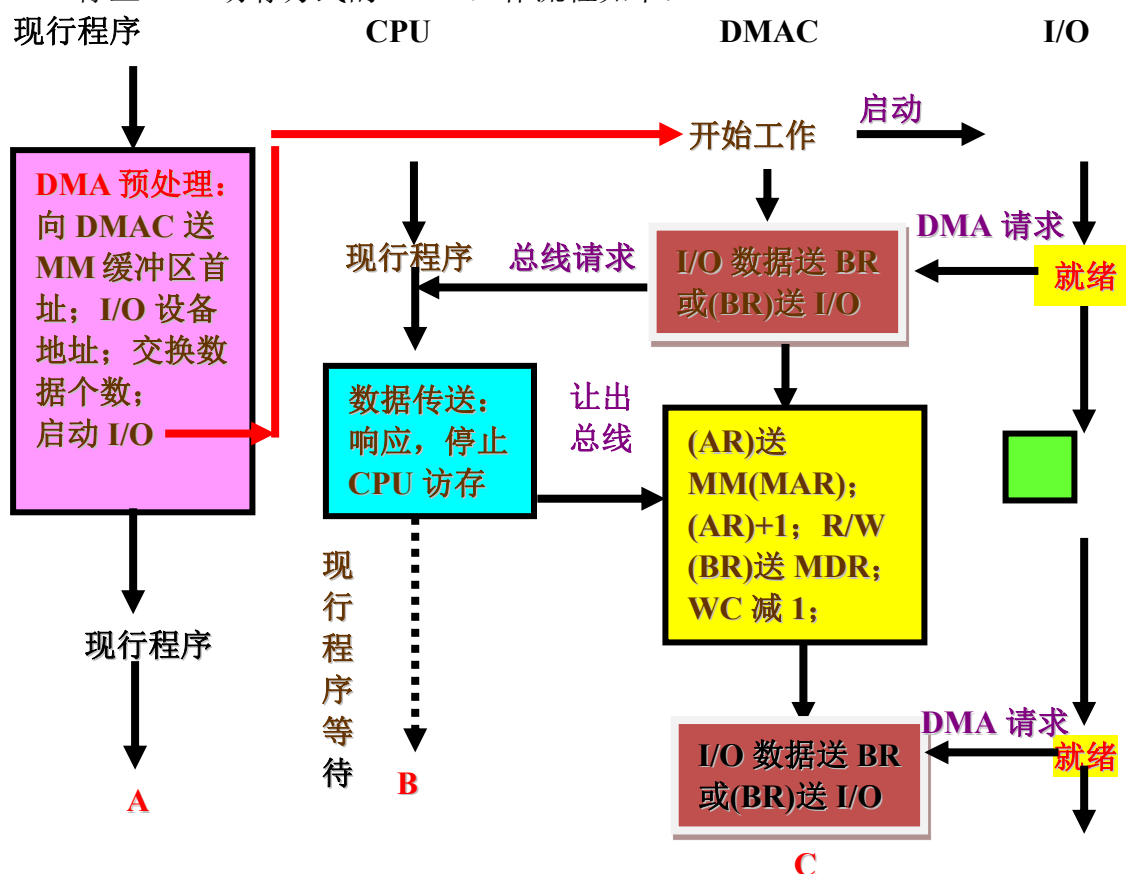
中断方式是程序切换，而程序又是由指令组成，所以必须在一条指令执行完毕才能响应中断请求，而且 CPU 只有在每条指令执行周期结束的时刻才发出查询信号，以获取中断请求信号，若此时条件满足，便能响应中断请求。

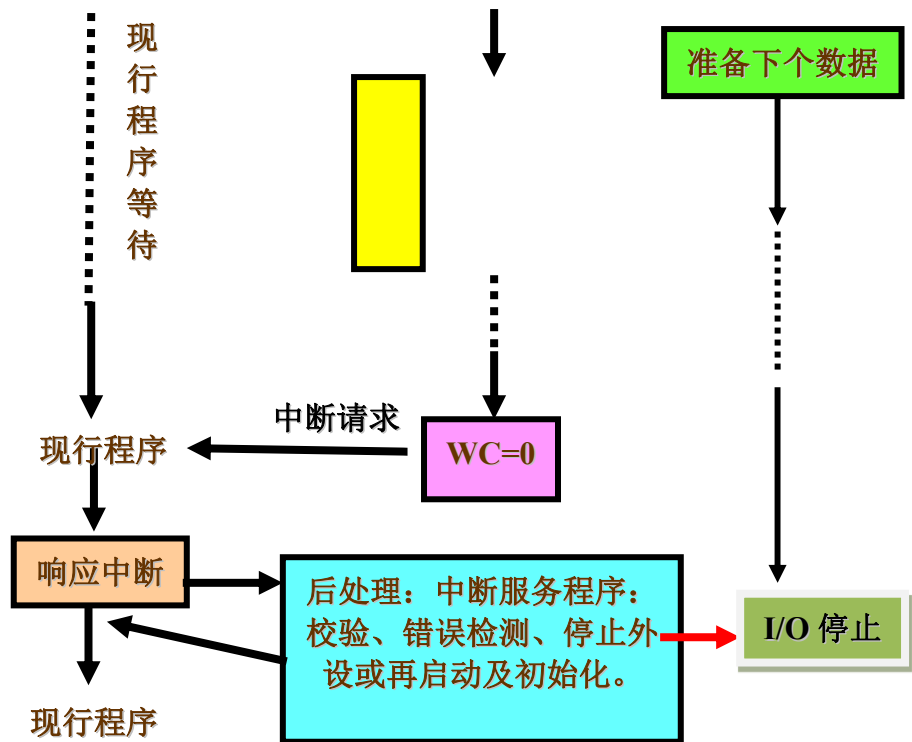
DMA 请求是由 DMA 接口根据设备的工作状态向 CPU 申请占用总线，此时只要总线未被 CPU 占用，即可立即响应 DMA 请求；若总线正被 CPU 占用，则必须等待该存取周期结束时，CPU 才交出总线的使用权。

30. DMA 的工作方式中，CPU 暂停方式和周期挪用方式的数据传送流程有何不同？画图说明。

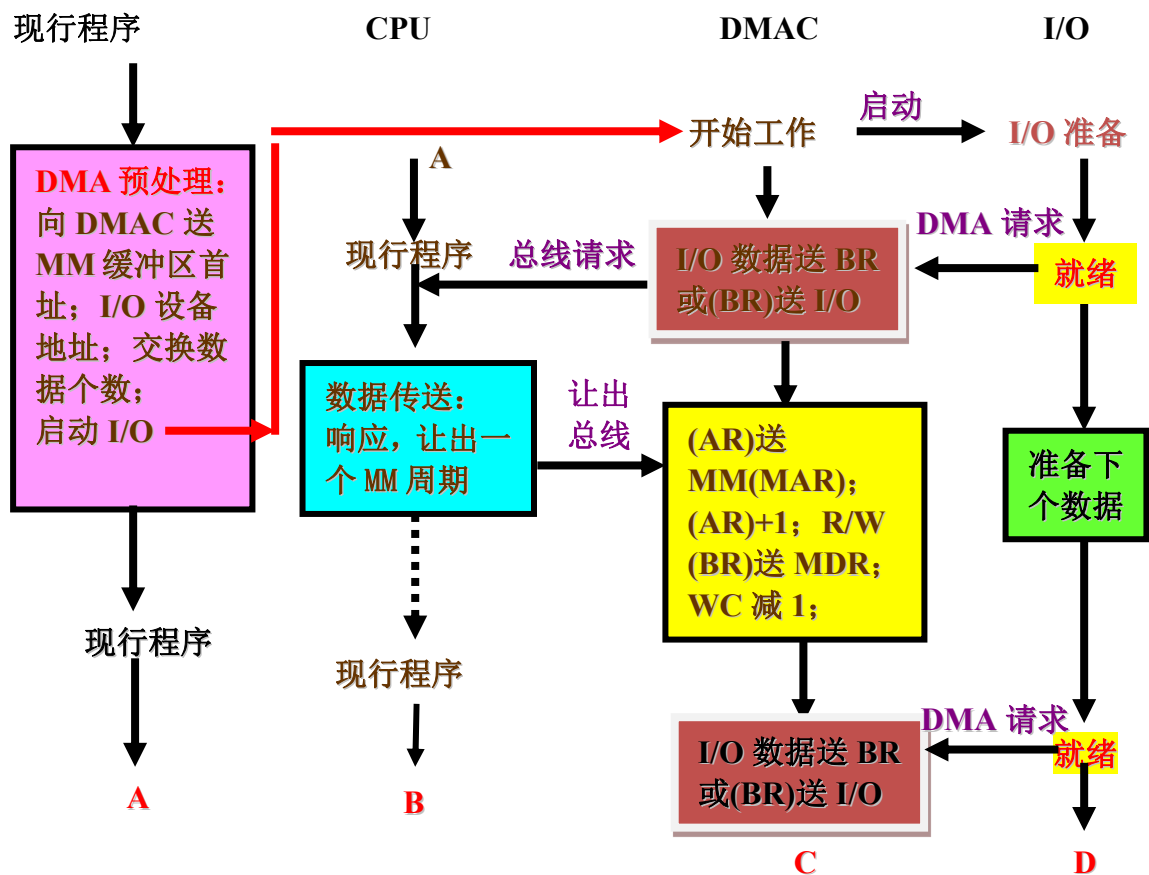
解：两种 DMA 方式的工作流程，其主要区别在于传送阶段，现行程序是否完全停止访存。

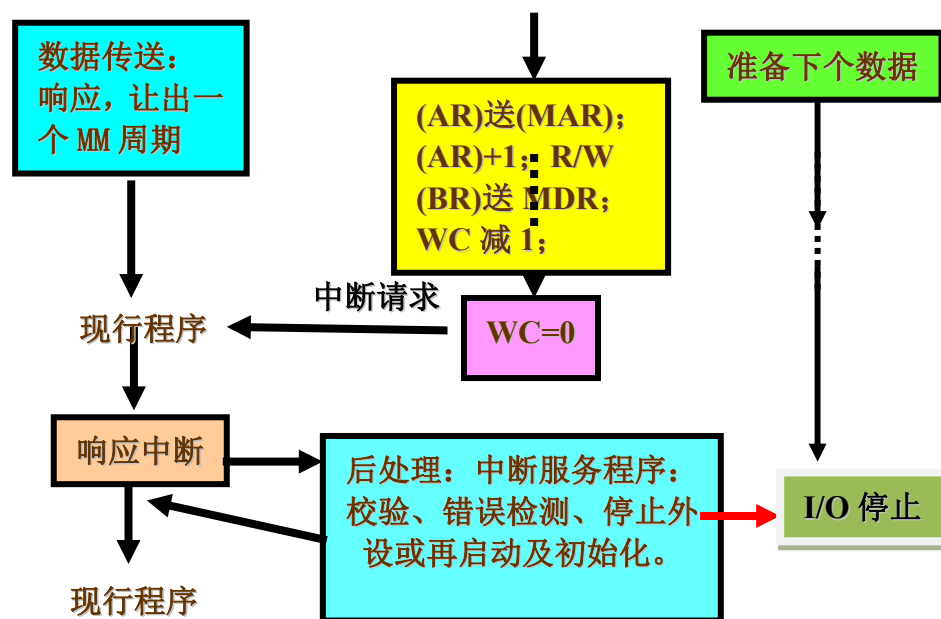
停止 CPU 访存方式的 DMA 工作流程如下：





周期窃取方式的 DMA 工作流程如下：





31. 假设某设备向 CPU 传送信息的最高频率是 40 000 次/秒，而相应的中断处理程序其执行时间为 $40\mu\text{s}$ ，试问该外设是否可用程序中断方式与主机交换信息，为什么？

解：该设备向 CPU 传送信息的时间间隔 $=1/40000=0.025\times 10^{-3}=25\mu\text{s} < 40\mu\text{s}$

则：该外设不能用程序中断方式与主机交换信息，因为其中断处理程序的执行速度比该外设的交换速度慢。

32. 设磁盘存储器转速为 3000 转/分，分 8 个扇区，每扇区存储 1K 字节，主存与磁盘存储器数据传送的宽度为 16 位（即每次传送 16 位）。假设一条指令最长执行时间是 $25\mu\text{s}$ ，是否可采用一条指令执行结束时响应 DMA 请求的方案，为什么？若不行，应采取什么方案？

解：先算出磁盘传送速度，然后和指令执行速度进行比较得出结论。

道容量 $= 1\text{K} \times 8 \times 8 \text{ 位} = 8\text{KB} = 4\text{K 字}$

数传率 $= 4\text{K 字} \times 3000 \text{ 转/分} = 4\text{K 字} \times 50 \text{ 转/秒} = 200\text{K 字/秒}$

一个字的传送时间 $= 1/200\text{K 秒} \approx 5\mu\text{s}$ （注：在此 $1\text{K}=1024$ ，来自数据块单位缩写。）

因为 $5\mu\text{s} \ll 25\mu\text{s}$ ，所以不能采用一条指令执行结束时响应 DMA 请求的方案，应采取每个 CPU 机器周期末查询及响应 DMA 请求的方案（通常安排 CPU 机器周期 $= \text{MM 存取周期}$ ）。

33. 试从下面七个方面比较程序查询、程序中断和 DMA 三种方式的综合性能。

- (1) 数据传送依赖软件还是硬件。
- (2) 传送数据的基本单位。
- (3) 并行性。
- (4) 主动性。
- (5) 传输速度。
- (6) 经济性。
- (7) 应用对象。

解：比较如下：

(1) 程序查询、程序中断方式的数据传送主要依赖软件，DMA 主要依赖硬件。（注意：这里指主要的趋势）

(2) 程序查询、程序中断传送数据的基本单位为字或字节，DMA 为数据块。

(3) 程序查询方式传送时，CPU 与 I/O 设备串行工作；程序中断方式时，CPU 与 I/O 设备并行工作，现行程序与 I/O 传送串行进行；DMA 方式时，CPU 与 I/O 设备并行工作，现行程序与 I/O 传送并行进行。

(4) 程序查询方式时，CPU 主动查询 I/O 设备状态；程序中断及 DMA 方式时，CPU 被动接受 I/O 中断请求或 DMA 请求。

(5) 程序中断方式由于软件额外开销时间比较大，因此传输速度最慢；程序查询方式软件额外开销时间基本没有，因此传输速度比中断快；DMA 方式基本由硬件实现传送，因此速度最快；

注意：程序中断方式虽然 CPU 运行效率比程序查询高，但传输速度却比程序查询慢。

(6) 程序查询接口硬件结构最简单，因此最经济；程序中断接口硬件结构稍微复杂一些，因此较经济；DMA 控制器硬件结构最复杂，因此成本最高；

(7) 程序中断方式适用于中、低速设备的 I/O 交换；程序查询方式适用于中、低速实时处理过程；DMA 方式适用于高速设备的 I/O 交换；

第 6 章 计算机的运算方法

2. 已知 $X=0.a_1a_2a_3a_4a_5a_6$ (a_i 为 0 或 1), 讨论下列几种情况时 a_i 各取何值。

(1) $x > \frac{1}{2}$

(2) $x \geq \frac{1}{8}$

(3) $\frac{1}{4} \geq x > \frac{1}{16}$

解: (1) 若要 $x > \frac{1}{2}$, 只要 $a_1=1$, $a_2 \sim a_6$ 不全为 0 即可。

(2) 若要 $x \geq \frac{1}{8}$, 只要 $a_1 \sim a_3$ 不全为 0 即可。

(3) 若要 $\frac{1}{4} \geq x > \frac{1}{16}$, 只要 $a_1=0$, a_2 可任取 0 或 1;

当 $a_2=0$ 时, 若 $a_3=0$, 则必须 $a_4=1$, 且 a_5 、 a_6 不全为 0;

若 $a_3=1$, 则 $a_4 \sim a_6$ 可任取 0 或 1;

当 $a_2=1$ 时, $a_3 \sim a_6$ 均取 0。

3. 设 x 为整数, $[x]_{\text{补}}=1, x_1x_2x_3x_4x_5$, 若要求 $x < -16$, 试问 $x_1 \sim x_5$ 应取何值?

解: 若要 $x < -16$, 需 $x_1=0$, $x_2 \sim x_5$ 任意。(注: 负数绝对值大的补码码值反而小。)

4. 设机器数字长为 8 位 (含 1 位符号位在内), 写出对应下列各真值的原码、补码和反码。 -13/64, 29/128, 100, -87

解: 真值与不同机器码对应关系如下:

真值	-13/64	29/128	100	-87
二进制	-0.001101	0.0011101	1100100	-1010111
原码	1.001 1010	0.001 1101	0110 0100	1101 0111
补码	1.1100110	0.001 1101	0110 0100	10101001
反码	1.1100101	0.001 1101	0110 0100	10101000

5. 已知 $[x]_{\text{补}}$, 求 $[x]_{\text{原}}$ 和 x 。

$[x_1]_{\text{补}}=1.1100$; $[x_2]_{\text{补}}=1.1001$; $[x_3]_{\text{补}}=0.1110$; $[x_4]_{\text{补}}=1.0000$;

$[x_5]_{\text{补}}=1,0101$; $[x_6]_{\text{补}}=1,1100$; $[x_7]_{\text{补}}=0,0111$; $[x_8]_{\text{补}}=1,0000$;

解: $[x]_{\text{补}}$ 与 $[x]_{\text{原}}$ 、 x 的对应关系如下:

$[x]_{\text{补}}$	1.1100	1.1001	0.1110	1.0000	1,0101	1,1100	0,0111	1,0000
$[x]_{\text{原}}$	1.0100	1.0111	0.1110	无	1,1011	1,0100	0,0111	无
x	-0.0100	-0.0111	0.1110	-1	-1011	-100	0,0111	-10000

6. 设机器数字长为 8 位 (含 1 位符号位在内), 分整数和小数两种情况讨论真值 x 为何值时, $[x]_{\text{补}}=[x]_{\text{原}}$ 成立。

解: 当 x 为小数时, 若 $x \geq 0$, 则 $[x]_{\text{补}}=[x]_{\text{原}}$ 成立;

若 $x < 0$, 当 $x = -1/2$ 时, $[x]_{\text{补}}=[x]_{\text{原}}=1.100 0000$, 则 $[x]_{\text{补}}=[x]_{\text{原}}$ 成立。

当 x 为整数时，若 $x \geq 0$ ，则 $[x]_{\text{补}} = [x]_{\text{原}}$ 成立；
 若 $x < 0$ ，当 $x = -64$ 时， $[x]_{\text{补}} = [x]_{\text{原}} = 1,100\ 0000$ ，则 $[x]_{\text{补}} = [x]_{\text{原}}$ 成立。

7. 设 x 为真值， x^* 为绝对值，说明 $[-x^*]_{\text{补}} = [-x]_{\text{补}}$ 能否成立。

解：当 x 为真值， x^* 为绝对值时， $[-x^*]_{\text{补}} = [-x]_{\text{补}}$ 不能成立。原因如下：

(1) 当 $x < 0$ 时，由于 $[-x^*]_{\text{补}}$ 是一个负值，而 $[-x]_{\text{补}}$ 是一个正值，因此此时 $[-x^*]_{\text{补}} = [-x]_{\text{补}}$ 不成立；

(2) 当 $x \geq 0$ 时，由于 $-x^* = -x$ ，因此此时 $[-x^*]_{\text{补}} = [-x]_{\text{补}}$ 的结论成立。

8. 讨论若 $[x]_{\text{补}} > [y]_{\text{补}}$ ，是否有 $x > y$ ？

解：若 $[x]_{\text{补}} > [y]_{\text{补}}$ ，不一定有 $x > y$ 。 $[x]_{\text{补}} > [y]_{\text{补}}$ 时 $x > y$ 的结论只在 $x > 0$ 且 $y > 0$ ，及 $x < 0$ 且 $y < 0$ 时成立。

由于正数补码的符号位为 0，负数补码的符号位为 1，当 $x > 0$ 、 $y < 0$ 时，有 $x > y$ ，但则 $[x]_{\text{补}} < [y]_{\text{补}}$ ；同样，当 $x < 0$ 、 $y > 0$ 时，有 $x < y$ ，但 $[x]_{\text{补}} > [y]_{\text{补}}$ 。

9. 当十六进制数 9B 和 FF 分别表示为原码、补码、反码、移码和无符号数时，所对应的十进制数各为多少（设机器数采用一位符号位）？

解：真值和机器数的对应关系如下：

9BH	原码	补码	反码	移码	无符号数
对应十进制数	-27	-101	-100	+27	155
FFH	原码	补码	反码	移码	无符号数
对应十进制数	-127	-1	-0	+127	255

10. 在整数定点机中，设机器数采用 1 位符号位，写出 ± 0 的原码、补码、反码和移码，得出什么结论？

解：0 的机器数形式如下：（假定机器数共 8 位，含 1 位符号位在内）

真值	原码	补码	反码	移码
+0	0 000 0000	0 000 0000	0 000 0000	1 000 0000
-0	1 000 0000	0 000 0000	1 111 1111	1 000 0000

结论：0 的原码和反码分别有 +0 和 -0 两种形式，补码和移码只有一种形式，且补码和移码数值位相同，符号位相反。

11. 已知机器数字长为 4 位（含 1 位符号位），写出整数定点机和小数定点机中原码、补码和反码的全部形式，并注明其对应的十进制真值。

整数定点机				小数定点机			
原码	补码	反码	真值	原码	补码	反码	真值
0,000	0,000	0,000	+0	0.000	0.000	0.000	+0
0,001	0,001	0,001	1	0.001	0.001	0.001	0.125
0,010	0,010	0,010	2	0.010	0.010	0.010	0.250
0,011	0,011	0,011	3	0.011	0.011	0.011	0.375
0,100	0,100	0,100	4	0.100	0.100	0.100	0.500
0,101	0,101	0,101	5	0.101	0.101	0.101	0.625

0,110	0,110	0,110	6	0.110	0.110	0.110	0.750
0,111	0,111	0,111	7	0.111	0.111	0.111	0.875
1,000	0,000	1,111	-0	1.000	0.000	1.111	-0
1,001	1,111	1,110	-1	1.001	1.111	1.110	-0.125
1,010	1,110	1,101	-2	1.010	1.110	1.101	-0.250
1,011	1,101	1,100	-3	1.011	1.101	1.100	-0.375
1,100	1,100	1,011	-4	1.100	1.100	1.011	-0.500
1,101	1,011	1,010	-5	1.101	1.011	1.010	-0.625
1,110	1,010	1,001	-6	1.110	1.010	1.001	-0.750
1,111	1,001	1,000	-7	1.111	1.001	1.000	-0.875
无	1,000	无	-8	无	1.000	无	-1

12. 设浮点数格式为：阶码 5 位（含 1 位阶符），尾数 11 位（含 1 位数符）。写出 51/128、-27/1024、7.375、-86.5 所对应的机器数。要求如下：

- (1) 阶码和尾数均为原码。
- (2) 阶码和尾数均为补码。
- (3) 阶码为移码，尾数为补码。

解：据题意画出该浮点数的格式：

阶符 1 位	阶码 4 位	数符 1 位	尾数 10 位
--------	--------	--------	---------

将十进制数转换为二进制：x1=51/128=0.0110011B=2⁻¹*0.110 011B

$$x2=-27/1024=-0.0000011011B=$$

$$2^{-5}*(-0.11011B)$$

$$x3=7.375=111.011B=2^3*0.111011B$$

$$x4=-86.5=-1010110.1B=2^7*(-0.10101101B)$$

则以上各数的浮点规格化数为：

- (1) [x1]浮=1, 0001; 0.110 011 000 0
[x2]浮=1, 0101; 1.110 110 000 0
[x3]浮=0, 0011; 0.111 011 000 0
[x4]浮=0, 0111; 1.101 011 010 0
- (2) [x1]浮=1, 1111; 0.110 011 000 0
[x2]浮=1, 1011; 1.001 010 000 0
[x3]浮=0, 0011; 0.111 011 000 0
[x4]浮=0, 0111; 1.010 100 110 0
- (3) [x1]浮=0, 1111; 0.110 011 000 0
[x2]浮=0, 1011; 1.001 010 000 0
[x3]浮=1, 0011; 0.111 011 000 0
[x4]浮=1, 0111; 1.010 100 110 0

13. 浮点数格式同上题，当阶码基值分别取 2 和 16 时：

- (1) 说明 2 和 16 在浮点数中如何表示。
- (2) 基值不同对浮点数有什么影响？
- (3) 当阶码和尾数均用补码表示，且尾数采用规格化形式，给出两种情况下所能表示的最大正数和非零最小正数真值。

解：(1) 阶码基值不论取何值，在浮点数中均为隐含表示，即：2 和 16 不出现

在浮点格式中，仅为人为的约定。

(2) 当基值不同时，对数的表示范围和精度都有影响。即：在浮点格式不变的情况下，基越大，可表示的浮点数范围越大，但浮点数精度越低。

(3) $r=2$ 时，

最大正数的浮点格式为：0, 1111; 0.111 111 111 1

其真值为： $N_{+max}=2^{15} \times (1-2^{-10})$

非零最小规格化正数浮点格式为：1, 0000; 0.100 000 000 0

其真值为： $N_{+min}=2^{-16} \times 2^{-1}=2^{-17}$

$r=16$ 时，

最大正数的浮点格式为：0, 1111; 0.1111 1111 11

其真值为： $N_{+max}=16^{15} \times (1-2^{-10})$

非零最小规格化正数浮点格式为：1, 0000; 0.0001 0000 00

其真值为： $N_{+min}=16^{-16} \times 16^{-1}=16^{-17}$

14. 设浮点数字长为 32 位，欲表示 ± 6 万间的十进制数，在保证数的最大精度条件下，除阶符、数符各取 1 位外，阶码和尾数各取几位？按这样分配，该浮点数溢出的条件是什么？

解：若要保证数的最大精度，应取阶码的基值=2。

若要表示 ± 6 万间的十进制数，由于 $32768 (2^{15}) < 6 \text{ 万} < 65536 (2^{16})$ ，则：阶码除阶符外还应取 5 位（向上取 2 的幂）。

故：尾数位数=32-1-1-5=25 位

25 (32) 该浮点数格式如下：

阶符 (1 位)	阶码 (5 位)	数符 (1 位)	尾数 (25 位)
----------	----------	----------	-----------

按此格式，该浮点数上溢的条件为：阶码 $\geq 2^5$

15. 什么是机器零？若要求全 0 表示机器零，浮点数的阶码和尾数应采取什么机器数形式？

解：机器零指机器数所表示的零的形式，它与真值零的区别是：机器零在数轴上表示为“0”点及其附近的一段区域，即在计算机中小到机器数的精度达不到的数均视为“机器零”，而真零对应数轴上的一点（0 点）。若要求用“全 0”表示浮点机器零，则浮点数的阶码应用移码、尾数用补码表示（此时阶码为最小阶、尾数为零，而移码的最小码值正好为“0”，补码的零的形式也为“0”，拼起来正好为一串 0 的形式）。

16. 设机器数字长为 16 位，写出下列各种情况下它能表示的数的范围。设机器数采用一位符号位，答案均用十进制表示。

(1) 无符号数；

(2) 原码表示的定点小数。

(3) 补码表示的定点小数。

(4) 补码表示的定点整数。

(5) 原码表示的定点整数。

(6) 浮点数的格式为：阶码 6 位（含 1 位阶符），尾数 10 位（含 1 位数符）。分别写出其正数和负数的表示范围。

(7) 浮点数格式同 (6)，机器数采用补码规格化形式，分别写出其对应的

正数和负数的真值范围。

解：(1) 无符号整数： $0 \sim 2^{16} - 1$ ，即： $0 \sim 65535$ ；

无符号小数： $0 \sim 1 - 2^{-16}$ ，即： $0 \sim 0.999998$ ；

(2) 原码定点小数： $-1 + 2^{-15} \sim 1 - 2^{-15}$ ，即： $-0.999997 \sim 0.999997$

(3) 补码定点小数： $-1 \sim 1 - 2^{-15}$ ，即： $-1 \sim 0.999997$

(4) 补码定点整数： $-2^{15} \sim 2^{15} - 1$ ，即： $-32768 \sim 32767$

(5) 原码定点整数： $-2^{15} + 1 \sim 2^{15} - 1$ ，即： $-32767 \sim 32767$

(6) 据题意画出该浮点数格式，当阶码和尾数均采用原码，非规格化数表示时：

最大负数=1, 11 111; 1.000 000 001，即 $-2^{-9} \times 2^{-31}$

最小负数=0, 11 111; 1.111 111 111，即 $-(1-2^{-9}) \times 2^{31}$

则负数表示范围为： $-(1-2^{-9}) \times 2^{31} \sim -2^{-9} \times 2^{-31}$

最大正数=0, 11 111; 0.111 111 111，即 $(1-2^{-9}) \times 2^{31}$

最小正数=1, 11 111; 0.000 000 001，即 $2^{-9} \times 2^{-31}$

则正数表示范围为： $2^{-9} \times 2^{-31} \sim (1-2^{-9}) \times 2^{31}$

(7) 当机器数采用补码规格化形式时，若不考虑隐藏位，则

最大负数=1, 00 000; 1.011 111 111，即 $-2^{-1} \times 2^{-32}$

最小负数=0, 11 111; 1.000 000 000，即 -1×2^{31}

则负数表示范围为： $-1 \times 2^{31} \sim -2^{-1} \times 2^{-32}$

最大正数=0, 11 111; 0.111 111 111，即 $(1-2^{-9}) \times 2^{31}$

最小正数=1, 00 000; 0.100 000 000，即 $2^{-1} \times 2^{-32}$

则正数表示范围为： $2^{-1} \times 2^{-32} \sim (1-2^{-9}) \times 2^{31}$

17. 设机器数字长为8位（包括一位符号位），对下列各机器数进行算术左移一位、两位，算术右移一位、两位，讨论结果是否正确。

$[x1]_{\text{原}}=0.001\ 1010$ ； $[y1]_{\text{补}}=0.101\ 0100$ ； $[z1]_{\text{反}}=1.010\ 1111$ ；

$[x2]_{\text{原}}=1.110\ 1000$ ； $[y2]_{\text{补}}=1.110\ 1000$ ； $[z2]_{\text{反}}=1.110\ 1000$ ；

$[x3]_{\text{原}}=1.001\ 1001$ ； $[y3]_{\text{补}}=1.001\ 1001$ ； $[z3]_{\text{反}}=1.001\ 1001$ 。

解：算术左移一位：

$[x1]_{\text{原}}=0.011\ 0100$ ；正确

$[x2]_{\text{原}}=1.101\ 0000$ ；溢出（丢1）出错

$[x3]_{\text{原}}=1.011\ 0010$ ；正确

$[y1]_{\text{补}}=0.010\ 1000$ ；溢出（丢1）出错

$[y2]_{\text{补}}=1.101\ 0000$ ；正确

$[y3]_{\text{补}}=1.011\ 0010$ ；溢出（丢0）出错

$[z1]_{\text{反}}=1.101\ 1111$ ；溢出（丢0）出错

$[z2]_{\text{反}}=1.101\ 0001$ ；正确

$[z3]_{\text{反}}=1.011\ 0011$ ；溢出（丢0）出错

算术左移两位：

$[x1]_{\text{原}}=0.110\ 1000$ ；正确

$[x2]_{\text{原}}=1.010\ 0000$ ；溢出（丢11）出错

$[x3]_{\text{原}}=1.110\ 0100$ ；正确

$[y1]_{\text{补}}=0.101\ 0000$ ；溢出（丢10）出错

$[y2]_{\text{补}}=1.010\ 0000$ ；正确

$[y3]_{\text{补}}=1.110\ 0100$; 溢出 (丢 00) 出错

$[z1]_{\text{反}}=1.011\ 1111$; 溢出 (丢 01) 出错

$[z2]_{\text{反}}=1.010\ 0011$; 正确

$[z3]_{\text{反}}=1.110\ 0111$; 溢出 (丢 00) 出错

算术右移一位:

$[x1]_{\text{原}}=0.000\ 1101$; 正确

$[x2]_{\text{原}}=1.011\ 0100$; 正确

$[x3]_{\text{原}}=1.000\ 1100(1)$; 丢 1, 产生误差

$[y1]_{\text{补}}=0.010\ 1010$; 正确

$[y2]_{\text{补}}=1.111\ 0100$; 正确

$[y3]_{\text{补}}=1.100\ 1100(1)$; 丢 1, 产生误差

$[z1]_{\text{反}}=1.101\ 0111$; 正确

$[z2]_{\text{反}}=1.111\ 0100(0)$; 丢 0, 产生误差

$[z3]_{\text{反}}=1.100\ 1100$; 正确

算术右移两位:

$[x1]_{\text{原}}=0.000\ 0110\ (10)$; 产生误差

$[x2]_{\text{原}}=1.001\ 1010$; 正确

$[x3]_{\text{原}}=1.000\ 0110\ (01)$; 产生误差

$[y1]_{\text{补}}=0.001\ 0101$; 正确

$[y2]_{\text{补}}=1.111\ 1010$; 正确

$[y3]_{\text{补}}=1.110\ 0110\ (01)$; 产生误差

$[z1]_{\text{反}}=1.110\ 1011$; 正确

$[z2]_{\text{反}}=1.111\ 1010\ (00)$; 产生误差

$[z3]_{\text{反}}=1.110\ 0110\ (01)$; 产生误差

18. 试比较逻辑移位和算术移位。

解: 逻辑移位和算术移位的区别:

逻辑移位是对逻辑数或无符号数进行的移位, 其特点是不论左移还是右移, 空出位均补 0, 移位时不考虑符号位。

算术移位是对带符号数进行的移位操作, 其关键规则是移位时符号位保持不变, 空出位的补入值与数的正负、移位方向、采用的码制等有关。补码或反码右移时具有符号延伸特性。左移时可能产生溢出错误, 右移时可能丢失精度。

19. 设机器数字长为 8 位 (含 1 位符号位), 用补码运算规则计算下列各题。

(1) $A=9/64$, $B=-13/32$, 求 $A+B$ 。

(2) $A=19/32$, $B=-17/128$, 求 $A-B$ 。

(3) $A=-3/16$, $B=9/32$, 求 $A+B$ 。

(4) $A=-87$, $B=53$, 求 $A-B$ 。

(5) $A=115$, $B=-24$, 求 $A+B$ 。

解: (1) $A=9/64=0.001\ 0010B$, $B=-13/32=-0.011\ 0100B$

$[A]_{\text{补}}=0.001\ 0010$, $[B]_{\text{补}}=1.100\ 1100$

$[A+B]_{\text{补}}=0.0010010+1.1001100=1.1011110$ ——无溢出

$A+B=-0.010\ 0010B=-17/64$

(2) $A=19/32=0.100\ 1100B$, $B=-17/128=-0.001\ 0001B$

- $[A]_{\text{补}}=0.100\ 1100$, $[B]_{\text{补}}=1.110\ 1111$, $[-B]_{\text{补}}=0.001\ 0001$
 $[A-B]_{\text{补}}=0.1001100+0.0010001=0.1011101$ ——无溢出
 $A-B=0.101\ 1101B=93/128B$
- (3) $A=-3/16=-0.001\ 1000B$, $B=9/32=0.010\ 0100B$
 $[A]_{\text{补}}=1.110\ 1000$, $[B]_{\text{补}}=0.010\ 0100$
 $[A+B]_{\text{补}}=1.1101000+0.0100100=0.0001100$ —— 无溢出
 $A+B=0.000\ 1100B=3/32$
- (4) $A=-87=-101\ 0111B$, $B=53=110\ 101B$
 $[A]_{\text{补}}=1\ 010\ 1001$, $[B]_{\text{补}}=0\ 011\ 0101$, $[-B]_{\text{补}}=1\ 100\ 1011$
 $[A-B]_{\text{补}}=1\ 0101001+1\ 1001011=0\ 1110100$ —— 溢出
- (5) $A=115=111\ 0011B$, $B=-24=-11\ 000B$
 $[A]_{\text{补}}=0\ 1110011$, $[B]_{\text{补}}=1, 110\ 1000$
 $[A+B]_{\text{补}}=0\ 1110011+1\ 1101000=0\ 1011011$ ——无溢出
 $A+B=101\ 1011B=91$

20. 用原码一位乘和补码一位乘 (Booth 算法) 计算 $x \cdot y$ 。

- (1) $x=0.110\ 111$, $y=-0.101\ 110$;
 (2) $x=-0.010\ 111$, $y=-0.010\ 101$;
 (3) $x=19$, $y=35$;
 (4) $x=0.110\ 11$, $y=-0.111\ 01$ 。

解：先将数据转换成所需的机器数，然后计算，最后结果转换成真值。

- (1) $[x]_{\text{原}}=0.110111$, $[y]_{\text{原}}=1.101110$, $x^*=0.110111$, $y^*=0.101110$

原码一位乘:

部分积	乘数 y^*	说明
0.000 000 +0.000 000	101 110	部分积初值为 0，乘数为 0 加 0
0.000 000 0.000 000 +0.110 111	010 111	右移一位 乘数为 1，加上 x^*
0.110 111 0.011 011 +0.110 111	101 011	右移一位 乘数为 1，加上 x^*
1.010 010 0.101 001 +0.110 111	010 101	右移一位 乘数为 1，加上 x^*
1.100 000 0.110 000 +0.000 000	001 010	右移一位 乘数为 0，加上 0
0.110 000 0.011 000 +0.110 111	000 101	右移一位 乘数为 1，加上 x^*
1.001 111 0.100 111	100 010	右移一位

即 $x^* \times y^* = 0.100\ 111\ 100\ 010$, $z_0 = x_0 \oplus y_0 = 0 \oplus 1 = 1$,

$[x \times y]_{\text{原}} = 1.100\ 111\ 100\ 010$, $x \cdot y = -0.100\ 111\ 100\ 010$

补码一位乘: $[x]_{\text{补}} = 0.110111$, $[-x]_{\text{补}} = 1.001001$, $[y]_{\text{补}} = 1.010010$

部分积	乘数	Y_{n+1}	说明
00 . 000 000	1 010 010 0 101 001	<u>0</u> <u>0</u>	$Y_n y_{n+1} = 00$, 部分积右移 1 位 $Y_n y_{n+1} = 10$, 部分积加 $[-x]_{\text{补}}$
00 . 000 000 +11 . 001 001			
11 . 001 001			右移 1 位
11 . 100 100 +00 . 110 111	1 010 100	<u>1</u>	$Y_n y_{n+1} = 01$, 部分积加 $[x]_{\text{补}}$
00 . 011 011			右移 1 位
00 . 001 101 00 . 000 110 +11 . 001 001	1 101 010 1 110 101	<u>0</u> <u>0</u>	$Y_n y_{n+1} = 00$, 部分积右移 1 位 $Y_n y_{n+1} = 10$, 部分积加 $[-x]_{\text{补}}$
11 . 001 111			右移 1 位
11 . 100 111 +00 . 110 111	1 111 010	<u>1</u>	$Y_n y_{n+1} = 01$, 部分积加 $[x]_{\text{补}}$
00 . 011 110 00 . 001 111 +11 . 001 001	0 111 101	<u>0</u>	右移 1 位 $Y_n y_{n+1} = 10$, 部分积加 $[-x]_{\text{补}}$
11 . 011 000	0 111 10		

即 $[x \times y]_{\text{补}} = 1.011\ 000\ 011\ 110$, $x \cdot y = -0.100\ 111\ 100\ 010$

21. 用原码加减交替法计算 $x \div y$ 。

- (1) $x = 0.100111$, $y = 0.101011$;
- (2) $x = -0.10101$, $y = 0.11011$;
- (3) $x = 0.10100$, $y = -0.10001$;
- (4) $x = 13/32$, $y = -27/32$ 。

解: (1) $x^* = [x]_{\text{原}} = [x]_{\text{补}} = x = 0.100\ 111$

$$y^*=[y]_{\text{原}}=[y]_{\text{补}}=y=0.101 \quad 011$$

$$[-y^*]_{\text{补}}=[-y]_{\text{补}}=1.010 \quad 101$$

$$q_0=x_0 \oplus y_0=0 \oplus 0=0$$

$$x \div y = x^* \div y^* = [x \div y]_{\text{原}} = 0.111010$$

$$r^*=0.000 \quad 010 \times 2^{-6}=0.000 \quad 000 \quad 000 \quad 010$$

$$(2) \quad x=-0.101 \quad 01, \quad y=0.110 \quad 11$$

$$[x]_{\text{原}}=1.101 \quad 01 \quad x^*=0.101 \quad 01$$

$$y^*=[y]_{\text{原}}=[y]_{\text{补}}=y=0.110 \quad 11 \quad [-y^*]_{\text{补}}=[-y]_{\text{补}}=1.001 \quad 01$$

$$[x]_{\text{补}}=1.010 \quad 11 \quad q_0=x_0 \oplus y_0=1 \oplus 0=1$$

$$x^* \div y^*=0.110 \quad 00 \quad [x \div y]_{\text{原}}=1.110 \quad 00$$

$$x \div y = -0.110 \quad 00 \quad r^*=0.110 \quad 00 \times 2^{-5}=0.000001 \quad 1000$$

$$(3) \quad x=0.101 \quad 00, \quad y=-0.100 \quad 01$$

$$x^*=[x]_{\text{原}}=[x]_{\text{补}}=x=0.101 \quad 00 \quad [y]_{\text{原}}=1.100 \quad 01$$

$$y^*=0.100 \quad 01 \quad [-y^*]_{\text{补}}=1.011 \quad 11$$

$$[y]_{\text{补}}=1.011 \quad 11 \quad [-y]_{\text{补}}=0.100 \quad 01$$

$$q_0=x_0 \oplus y_0=0 \oplus 1=1 \quad x^* \div y^*=1.001 \quad 01 \quad \text{—— 溢出}$$

$$[x \div y]_{\text{原}}: \text{无定义} \quad x \div y = -1.001 \quad 01$$

$$r^*=0.010 \quad 11 \times 2^{-5}=0.000 \quad 000 \quad 101 \quad 1$$

注：由于本题中 $x^* > y^*$ ，有溢出。除法运算时一般在运算前判断是否 $x^* > y^*$ ，如果该条件成立则停止运算，转溢出处理。但此算法本身在溢出情况下仍可正常运行，此时数值位占领小数点左边的1位，商需设双符号位（变形补码），以判溢出。采用这种方法时运算前可不判溢出，直接进行运算，运算完后再判溢出。

$$(4) \quad x=13/32=(0.011 \quad 01)_2$$

$$y=-27/32=(-0.110 \quad 11)_2 \quad x^*=[x]_{\text{原}}=[x]_{\text{补}}=x=0.011 \quad 01$$

$$[y]_{\text{原}}=1.11011 \quad y^*=0.110 \quad 11$$

$$[-y^*]_{\text{补}}=1.00101 \quad [y]_{\text{补}}=1.001 \quad 01$$

$$[-y]_{\text{补}}=0.11011 \quad q_0=x_0 \oplus y_0=0 \oplus 1=1$$

$$x^* \div y^*=0.01111 \quad [x \div y]_{\text{原}}=1.011 \quad 11$$

$$x \div y = (-0.011 \quad 11)_2 = -15/32 \quad r^*=0.010 \quad 11 \times 2^{-5}=0.000 \quad 000 \quad 101 \quad 1$$

26.按机器补码浮点运算步骤，计算 $[x \pm y]_{\text{补}}$ 。

$$(1) \quad x=2^{-011} \times 0.101 \quad 100, \quad y=2^{-010} \times (-0.011 \quad 100);$$

$$(2) \quad x=2^{-011} \times (-0.100 \quad 010), \quad y=2^{-010} \times (-0.011 \quad 111);$$

$$(3) \quad x=2^{101} \times (-0.100 \quad 101), \quad y=2^{100} \times (-0.001 \quad 111)。$$

解：先将 x 、 y 转换成机器数形式：

$$(1) \quad x=2^{-011} \times 0.101 \quad 100, \quad y=2^{-010} \times (-0.011 \quad 100)$$

$$[x]_{\text{补}}=1, 101; 0.101 \quad 100, [y]_{\text{补}}=1, 110; 1.100 \quad 100$$

$$[Ex]_{\text{补}}=1, 101, [y]_{\text{补}}=1, 110, [Mx]_{\text{补}}=0.101 \quad 100, [My]_{\text{补}}=1.100 \quad 100$$

1) 对阶：

$$[\Delta E]_{\text{补}}=[Ex]_{\text{补}}+[-Ey]_{\text{补}}=11, 101+00, 010=11, 111 < 0,$$

$$Ex \text{ 向 } Ey \text{ 对齐, 则: } [Ex]_{\text{补}}+1=11, 101+00, 001=11, 110=[Ey]_{\text{补}}$$

$$[x]_{\text{补}}=1, 110; 0.010 \quad 110$$

2) 尾数运算：

$$[Mx]_{\text{补}}+[My]_{\text{补}}=0.010 \quad 110+11.100 \quad 100=11.111010$$

$$[Mx]_{\text{补}}+[-My]_{\text{补}}=0.010 \quad 110+00.011100=00.110 \quad 010$$

3) 结果规格化：

$[x+y]_{\text{补}}=11, 110; 11.111\ 010 = 11, 011; 11.010\ 000$ (尾数左规 3 次, 阶码减 3)

$[x-y]_{\text{补}}=11, 110; 00.110\ 010$, 已是规格化数。

4) 舍入: 无

5) 溢出: 无

则: $x+y=2^{-101} \times (-0.110\ 000)$

$x-y=2^{-010} \times 0.110\ 010$

(2) $x=2^{-011} \times (-0.100010), y=2^{-010} \times (-0.011111)$

$[x]_{\text{补}}=1, 101; 1.011\ 110, [y]_{\text{补}}=1, 110; 1.100\ 001$

1) 对阶: 过程同(1)的 1), 则

$[x]_{\text{补}}=1, 110; 1.101\ 111$

2) 尾数运算:

$[Mx]_{\text{补}}+[My]_{\text{补}}=11.101111+11.100001=11.010000$

$[Mx]_{\text{补}}+[-My]_{\text{补}}=11.101111+00.011111=00.001110$

3) 结果规格化:

$[x+y]_{\text{补}}=11, 110; 11.010\ 000$, 已是规格化数

$[x-y]_{\text{补}}=11, 110; 00.001\ 110=11, 100; 00.111000$ (尾数左规 2 次, 阶码减 2)

4) 舍入: 无

5) 溢出: 无

则: $x+y=2^{-010} \times (-0.110\ 000)$

$x-y=2^{-100} \times 0.111\ 000$

(3) $x=2^{101} \times (-0.100\ 101), y=2^{100} \times (-0.001\ 111)$

$[x]_{\text{补}}=0, 101; 1.011\ 011, [y]_{\text{补}}=0, 100; 1.110\ 001$

1) 对阶:

$[\Delta E]_{\text{补}}=00, 101+11, 100=00, 001 > 0$, 应 E_y 向 E_x 对齐, 则:

$[E_y]_{\text{补}}+1=00, 100+00, 001=00, 101=[E_x]_{\text{补}}$

$[y]_{\text{补}}=0, 101; 1.111\ 000$ (1)

2) 尾数运算:

$[Mx]_{\text{补}}+[My]_{\text{补}}=11.011011+11.111000(1)=11.010011(1)$

$[Mx]_{\text{补}}+[-My]_{\text{补}}=11.011011+00.000111(1)=11.100010(1)$

3) 结果规格化:

$[x+y]_{\text{补}}=00, 101; 11.010\ 011(1)$, 已是规格化数

$[x-y]_{\text{补}}=00, 101; 11.100\ 010(1)=00, 100; 11.000\ 101$ (尾数左规 1 次, 阶码减 1)

4) 舍入:

$[x+y]_{\text{补}}=00, 101; 11.010\ 011$ (舍)

$[x-y]_{\text{补}}$ 不变

5) 溢出: 无

则: $x+y=2^{101} \times (-0.101\ 101)$

$x-y=2^{100} \times (-0.111\ 011)$

27. 设浮点数阶码取 3 位，尾数取 6 位（均不包括符号位），要求阶码用移码运算，尾数用补码运算，计算 $x \cdot y$ ，且结果保留 1 倍字长。

$$(1) x=2^{-100} \times 0.101101, \quad y=2^{-011} \times (-0.110101);$$

$$(2) x=2^{-011} \times (-0.100111), \quad y=2^{101} \times (-0.101011)。$$

解：先将 x 、 y 转换成机器数形式：

$$(1) [x]_{\text{阶移尾补}}=0, 100; 0.101 \quad 101 \quad [y]_{\text{阶移尾补}}=0, 101; 1.001 \quad 011$$

1) 阶码相加： $[Ex]_{\text{移}}+[Ey]_{\text{补}}=00, 100+11, 101=00, 001$ （无溢出）

2) 尾数相乘： $[Mx \times My]_{\text{补}}=1.011 \quad 010$ （101 111 00）

3) 结果规格化：已是规格化数。

4) 舍入：设采用 0 舍 1 入法，应入：

$$[x \times y]_{\text{阶移尾补}}=0, 001; 1.011 \quad 011$$

5) 溢出：无 $x \times y=2^{-111} \times (-0.100 \quad 101)$

$$(2) x=2^{-011} \times (-0.100 \quad 111) \quad y=2^{101} \times (-0.101 \quad 011)$$

$$[x]_{\text{阶移尾补}}=0, 101; 1.011 \quad 001 \quad [y]_{\text{阶移尾补}}=1, 101; 1.010 \quad 101$$

1) 阶码相加： $[Ex]_{\text{移}}+[Ey]_{\text{补}}=00, 101+00, 101=01, 010$ （无溢出）

2) 尾数相乘： $[Mx \times My]_{\text{补}}=0.011 \quad 010$ （001 101 00）

3) 结果规格化： $[x \times y]_{\text{阶移尾补}}=1, 010; 0.011 \quad 010$ （001101 00）= 1, 001; 0.110100（011 010 0）

4) 舍入：设采用 0 舍 1 入法，应舍： $[x \times y]_{\text{阶移尾补}}=1, 001; 0.110 \quad 100$

5) 溢出：无 $x \times y=2^{001} \times 0.110 \quad 100$

32. 设机器字长为 16 位，分别按 4、4、4、4 和 5、5、3、3 分组后，

(1) 画出按两种分组方案的单重分组并行进位链框图，并比较哪种方案运算速度快。

(2) 画出按两种分组方案的双重分组并行进位链框图，并对这两种方案进行比较。

(3) 用 74181 和 74182 画出单重和双重分组的并行进位链框图。

解：(1) 4—4—4—4 分组的 16 位单重分组并行进位链框图见教材 286 页图 6.22。

5—5—3—3 分组的 16 位单重分组并行进位链框图如下：

(2) 4—4—4—4 分组的 16 位双重分组并行进位链框图见教材 289 页图 6.26。

5—5—3—3 分组的 16 位双重分组并行进位链框图如下：

$$5—5—3—3 \text{ 分组的进位时间}=2.5t_y \times 3=7.5t_y;$$

$$4—4—4—4 \text{ 分组的进位时间}=2.5t_y \times 3=7.5t_y;$$

可见，两种分组方案最长加法时间相同。

结论：双重分组并行进位的最长进位时间只与组数和级数有关，与组内位数无关。

(3) 单重分组 16 位并行加法器逻辑图如下（正逻辑）：

注意： 1) 74181 芯片正、负逻辑的引脚表示方法；

2) 为强调可比性，5-5-3-3 分组时不考虑扇入影响；

3) 181 芯片只有最高、最低两个进位输入/输出端，组内进位无引脚；

4) 181 为 4 位片，无法 5-5-3-3 分组，只能 4-4-4-4 分组；

5) 单重分组跳跃进位只用到 181, 使用 182 的一定是双重以上分组跳跃进位;

6) 单重分组跳跃进位是并行进位和串行进位技术的结合; 双重分组跳跃进位是二级并行进位技术; 特别注意在位数较少时, 双重分组跳跃进位可以采用全先行进位技术实现; 位数较多时, 可采用双重分组跳跃进位和串行进位技术结合实现。

第7章 指令系统

1. 什么叫机器指令？什么叫指令系统？为什么说指令系统与机器的主要功能以及硬件结构之间存在着密切的关系？

答：参考 P300。

2. 什么叫寻址方式？为什么要学习寻址方式？

答：参看 P310。

3. 什么是指令字长、机器字长和存储字长？

答：略。

4. 零地址指令的操作数来自哪里？各举一例说明。

答：零地址指令的操作数来自 ACC，为隐含约定。

在一地址指令中，另一个操作数的地址通常可采用 ACC 隐含寻址方式获得。

5. 对于二地址指令而言，操作数的物理地址可安排在什么地方？举例说明。

答：对于二地址指令而言，操作数的物理地址可安排在寄存器内、指令中或内存单元内等。

8. 某机指令字长 16 位，每个操作数的地址码为 6 位，设操作码长度固定，指令分为零地址、一地址和二地址三种格式。若零地址指令有 M 条，一地址指令有 N 种，则二地址指令最多有几种？若操作码位数可变，则二地址指令最多允许有几种？

解：1) 若采用定长操作码时，二地址指令格式如下：

OP (4 位)	A1 (6 位)	A2 (6 位)
----------	----------	----------

设二地址指令有 K 种，则： $K=2^4-M-N$ ($M=[(2^4-K)*2^6]-N*2^6$)

当 $M=1$ (最小值), $N=1$ (最小值) 时，二地址指令最多有： $K_{\max}=16-1-1=14$ 种

2) 若采用变长操作码时，二地址指令格式仍如 1) 所示，但操作码长度可随地址码的个数而变。此时， $K=2^4-(N/2^6+M/2^{12})$;

当 $(N/2^6+M/2^{12}) \leq 1$ 时 ($N/2^6+M/2^{12}$ 向上取整)， K 最大，则二地址指令最多有：

$K_{\max}=16-1=15$ 种 (只留一种编码作扩展标志用。)

9. 试比较间接寻址和寄存器间接寻址。

答：略。

10. 试比较基址寻址和变址寻址。

解：比较如下：

1) 都可有效地扩大指令寻址范围。

2) 基址寻址时，基准地址由基址寄存器给出，地址的改变反映在位移量 A 的取值上；变址寻址时，基准地址由 A 给出，地址的改变反映在变址值的自动修改上，变址值由变址寄存器给出。

- 3) 基址寄存器内容通常由系统程序设定, 变址寄存器内容通常由用户设定。
- 4) 基址寻址适用于程序的动态重定位, 变址寻址适用于数组或字符串处理, 适用场合不同。

11. 画出先变址再间址及先间址再变址的寻址过程示意图。

解: 1) 先变址再间址寻址过程简单示意如下:

$$EA=[(IX)+A], IX \rightarrow (IX)+1$$

2) 先间址再变址寻址过程简单示意如下: $EA=(IX)+(A), IX \rightarrow (IX)+1$

13. 举例说明哪几种寻址方式在指令的执行阶段不访问存储器? 哪几种寻址方式在指令的执行阶段只需访问一次存储器? 完成什么样的指令, 包括取指令在内共访问四次存储器?

解: 举例如下:

- 1) 一地址指令在执行阶段不访存的寻址方式有: 寄存器寻址、立即寻址。
- 2) 一地址指令在执行阶段只访存一次的寻址方式有: 寄存器间接寻址、直接寻址、基址寻址、变址寻址、相对寻址、页面寻址。
- 3) 包括取指在内共访存四次的指令有: 二重间址的一地址指令; 一重间址的二地址指令, 当另一操作数采用直接、基址、变址、相对、页面、寄存器间接寻址时。

16. 某机主存容量为 $4M \times 16$ 位, 且存储字长等于指令字长, 若该机指令系统可完成 108 种操作, 操作码位数固定, 且具有直接、间接、变址、基址、相对、立即等六种寻址方式, 试回答: (1) 画出一地址指令格式并指出各字段的作用;

- (2) 该指令直接寻址的最大范围;
- (3) 一次间址和多次间址的寻址范围;
- (4) 立即数的范围 (十进制表示);
- (5) 相对寻址的位移量 (十进制表示);
- (6) 上述六种寻址方式的指令哪一种执行时间最短? 哪一种最长? 为什么? 哪一种便于程序浮动? 哪一种最适合处理数组问题?
- (7) 如何修改指令格式, 使指令的寻址范围可扩大到 $4M$?
- (8) 为使一条转移指令能转移到主存的任一位置, 可采取什么措施? 简要说明之。

解: (1) 单字长一地址指令格式:

OP (7 位)	M (3 位)	A (6 位)
----------	---------	---------

OP 为操作码字段, 共 7 位, 可反映 108 种操作;

M 为寻址方式字段, 共 3 位, 可反映 6 种寻址操作;

A 为地址码字段, 共 $16-7-3=6$ 位。

- (2) 直接寻址的最大范围为 $2^6=64$ 。
- (3) 由于存储字长为 16 位, 故一次间址的寻址范围为 2^{16} ; 若多次间址, 需用存储字的最高位来区别是否继续间接寻址, 故寻址范围为 2^{15} 。
- (4) 立即数的范围为 -32——31 (有符号数), 或 0——63 (无符号数)。
- (5) 相对寻址的位移量为 -32——31。
- (6) 上述六种寻址方式中, 因立即数由指令直接给出, 故立即寻址的指令执行时间最短。间接寻址在指令的执行阶段要多次访存 (一次间接寻址要两次访存, 多次间接寻址要多次访存), 故执行时间最长。变址寻址由于变址寄存器的

内容由用户给定，而且在程序的执行过程中允许用户修改，而其形式地址始终不变，故变址寻址的指令便于用户编制处理数组问题的程序。相对寻址操作数的有效地址只与当前指令地址相差一定的位移量，与直接寻址相比，更有利于程序浮动。

(7) **方案一**：为使指令寻址范围可扩大到 4M，需要有效地址 22 位，此时可将单字长一地址指令的格式改为双字长，如下图示：

OP (7 位)	MOD (3 位)	A (高 6 位)
A (低 16 位)		

方案二：如果仍采用单字长指令 (16 位) 格式，为使指令寻址范围扩大到 4M，可通过段寻址方案实现。安排如下：

硬件设段寄存器 DS (16 位)，用来存放段地址。在完成指令寻址方式所规定的寻址操作后，得有效地址 EA (6 位)，再由硬件自动完成段寻址，最后得 22 位物理地址。即：物理地址 = (DS) × 2⁶ + EA

注：段寻址方式由硬件隐含实现。在编程指定的寻址过程完成、EA 产生之后由硬件自动完成，对用户是透明的。

方案三：在采用单字长指令 (16 位) 格式时，还可通过页面寻址方案使指令寻址范围扩大到 4M。安排如下：

硬件设页面寄存器 PR (16 位)，用来存放页面地址。指令寻址方式中增设页面寻址。当需要使指令寻址范围扩大到 4M 时，编程选择页面寻址方式，则：EA = (PR) || A (有效地址 = 页面地址“拼接”6 位形式地址)，这样得到 22 位有效地址。

(8) 为使一条转移指令能转移到主存的任一位置，寻址范围须达到 4M，除了采用(7) 方案一中的双字长一地址指令的格式外，还可配置 22 位的基址寄存器或 22 位的变址寄存器，使 EA = (BR) + A (BR 为 22 位的基址寄存器)或 EA = (IX) + A (IX 为 22 位的变址寄存器)，便可访问 4M 存储空间。还可以通过 16 位的基址寄存器左移 6 位再和形式地址 A 相加，也可达到同样的效果。

总之，不论采取何种方式，最终得到的实际地址应是 22 位。

19. 某 CPU 内有 32 个 32 位的通用寄存器，设计一种能容纳 64 种操作的指令系统。假设指令字长等于机器字长，试回答以下问题：

(1) 如果主存可直接或间接寻址，采用寄存器—存储器型指令，能直接寻址的最大存储空间是多少？画出指令格式并说明各字段的含义。

(2) 在满足 (1) 的前提下，如果采用通用寄存器作基址寄存器，则上述寄存器—存储器型指令的指令格式有何特点？画出指令格式并指出这类指令可访问多大的存储空间？

解：(1) 如采用 RS 型指令，则此指令一定是二地址以上的地址格式，指令格式如下：

OP (6 位)	R (5 位)	I (1 位)	A (20 位)
----------	---------	---------	----------

操作码字段 OP 占 6 位，因为 2⁶ ≥ 64；

寄存器编号 R 占 5 位，因为 2⁵ ≥ 32；

间址位 I 占 1 位，当 I=0，存储器寻址的操作数为直接寻址，当 I=1 时为间接寻址；

形式地址 A 占 20 位，可以直接寻址 2²⁰ 字。

(2) 如采用基址寻址，则指令格式中应给出基址寄存器号，以指定哪一个通用寄存器用作基址寄存器。指令格式变为：

OP (6 位)	源 R (5 位)	I (1 位)	X (1 位)	目标 R (5 位)	A (14 位)
----------	-----------	---------	---------	------------	----------

增加寻址特征位 X，当 X=1 时，以目标寄存器 R 作为基址寄存器进行基址寻址。

基址寻址可访问存储空间为： 2^{32} 字。

第 8 章 CPU 的结构和功能

1. CPU 有哪些功能？画出其结构框图并简要说明各个部件的作用。

答：参考 P328 和图 8.2。

2. 什么是指令周期？指令周期是否有一个固定值？为什么？

解：指令周期是指取出并执行完一条指令所需的时间。

由于计算机中各种指令执行所需的时间差异很大，因此为了提高 CPU 运行效率，即使在同步控制的机器中，不同指令的指令周期长度都是不一致的，也就是说指令周期对于不同的指令来说不是一个固定值。

3. 画出指令周期的流程图，分析说明图中每个子周期的作用。

答：参看 P343 及图 8.8。

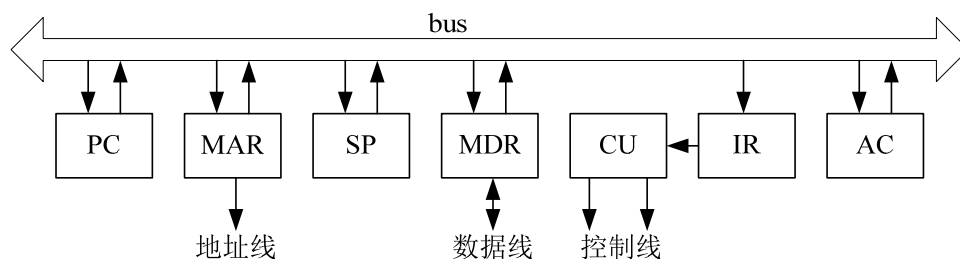
4. 设 CPU 内有下列部件：PC、IR、SP、AC、MAR、MDR 和 CU。

(1) 画出完成间接寻址的取数指令 LDA@X（将主存某地址单元 X 的内容取至 AC 中）的数据流（从取指令开始）。

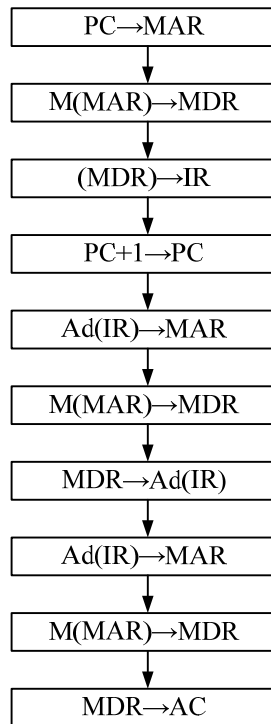
(2) 画出中断周期的数据流。

解：CPU 中的数据流向与所采用的数据通路结构直接相关，不同的数据通路中的数据流是不一样的。常用的数据通路结构方式有直接连线、单总线、双总线、三总线等形式，目前大多采用总线结构，直接连线方式仅适用于结构特别简单的机器中。

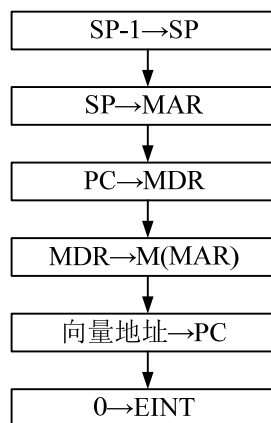
为简单起见，本题采用单总线将题中所给部件连接起来，框图如下：



(1) LDA@X 指令周期数据流程图：



(2) 中断周期流程图如下：



注：解这道题有两个要素，首先要根据所给部件设计好数据通路，即确定信息流动的载体。其次选择好描述数据流的方法，无论采用什么样的表达方式，其关键都要能清楚地反映数据在通路上流动的顺序，即强调一个“流”字。较好的表达方式是流程图的形式。

5. 中断周期前是什么阶段？中断周期后又是什么阶段？在中断周期 CPU 应完成什么操作？

答：中断周期前是执行周期，中断周期后是取指周期。在中断周期，CPU 应完成保存断点、将中断向量送 PC 和关中断等工作。

7. 什么叫系统的并行性？粗粒度并行和细粒度并行有何区别？

答：所谓并行性包含同时性和并发性。同时性是指两个或两个以上的事件在同一时刻发生，并发性是指两个或多个事件在同一时间段发生。即在同一时刻或同一时间段内完成两个或两个以上性质相同或性质不同的功能，只要在时间上存在相

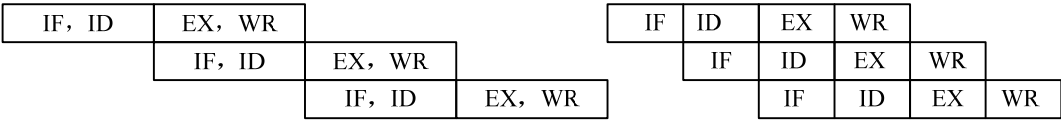
互重叠，就存在并行性。

并行性又分为粗粒度并行和细粒度并行两类。粗粒度并行是指在多个处理机上分别运行多个进程，由多台处理机合作完成一个程序，一般用算法实现。细粒度并行是指在处理机的指令级和操作级的并行性。

8. 什么是指令流水？画出指令二级流水和四级流水的示意图，它们中哪个更能提高处理机速度，为什么？

答：指令流水是指将一条指令的执行过程分为 n 个操作时间大致相等的阶段，每个阶段由一个独立的功能部件来完成，这样 n 个部件就可以同时执行 n 条指令的不同阶段，从而大大提高 CPU 的吞吐率。

指令二级流水和四级流水示意图如下：



二级指令流水示意图

四级指令流水示意图

四级流水更能提高处理机的速度。分析如下：

假设 IF、ID、EX、WR 每个阶段耗时为 t ，则连续执行 n 条指令

采用二级流水线时，耗时为： $4t+(n-1)2t=(2n+2)t$

采用四级流水线时，耗时为： $4t+(n-1)t=(n+3)t$

在 $n>1$ 时， $n+3<2n+2$ ，可见四级流水线耗时比二级流水线耗时短，因此更能提高处理机速度。

17. 在中断系统中 INTR、INT、EINT 三个触发器各有何作用？

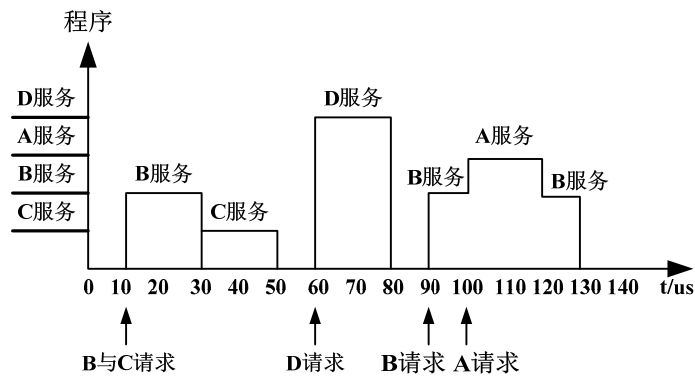
解：INTR——中断请求触发器，用来登记中断源发出的随机性中断请求信号，以便为 CPU 查询中断及中断排队判优线路提供稳定的中断请求信号。

EINT——中断允许触发器，CPU 中的中断总开关。当 $EINT=1$ 时，表示允许中断（开中断），当 $EINT=0$ 时，表示禁止中断（关中断）。其状态可由开、关中断等指令设置。

INT——中断标记触发器，控制器时序系统中周期状态分配电路的一部分，表示中断周期标记。当 $INT=1$ 时，进入中断周期，执行中断隐指令的操作。

24. 现有 A、B、C、D 四个中断源，其优先级由高向低按 A、B、C、D 顺序排列。若中断服务程序的执行时间为 $20\mu s$ ，请根据下图所示时间轴给出的中断源请求中断的时刻，画出 CPU 执行程序的轨迹。

解：A、B、C、D 的响优先级即处理优先级。CPU 执行程序的轨迹图如下：



25. 某机有五个中断源 L0、L1、L2、L3、L4，按中断响应的优先次序由高向低排序为 L0→L1→L2→L3→L4，根据下示格式，现要求中断处理次序改为 L1→L4→L2→L0→L3，根据下面的格式，写出各中断源的屏蔽字。

解：各中断源屏蔽状态见下表：

中断源	屏蔽字				
	0	1	2	3	4
I0	1	0	0	1	0
I1	1	1	1	1	1
I2	1	0	1	1	0
I3	0	0	0	1	0
I4	1	0	1	1	1

表中：设屏蔽位=1，表示屏蔽；屏蔽位=0，表示中断开放。

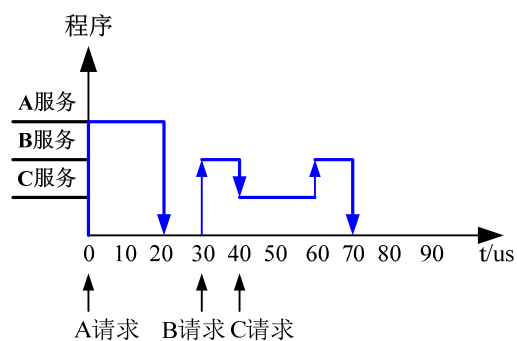
26. 设某机配有 A、B、C 三台设备，其优先顺序按 A→B→C 降序排列，为改变中断处理次序，它们的中断屏蔽字设置如下：

设备	屏蔽字
A	111
B	010
C	011

请按下图所示时间轴给出的设备请求中断的时刻，画出 CPU 执行程序的轨迹。

设 A、B、C 中断服务程序的执行时间均为 $20\mu s$ 。

解：A、B、C 设备的响应优先级为 A 最高、B 次之、C 最低，处理优先级为 A 最高、C 次之、B 最低。CPU 执行程序的轨迹图如下：



第 9 章 控制单元的功能

2. 控制单元的功能是什么？其输入受什么控制？

答：控制单元的主要功能是发出各种不同的控制信号。其输入受时钟信号、指令寄存器的操作码字段、标志和来自系统总线的控制信号的控制。

3. 什么是指令周期、机器周期和时钟周期？三者有何关系？

答：CPU 每取出并执行一条指令所需的全部时间叫指令周期；

机器周期是在同步控制的机器中，执行指令周期中一步相对完整的操作（指令步）所需时间，通常安排机器周期长度等于主存周期；

时钟周期是指计算机主时钟的周期时间，它是计算机运行时最基本的时序单位，对应完成一个微操作所需时间，通常时钟周期等于计算机主频的倒数。

4. 能不能说机器的主频越快，机器的速度就越快，为什么？

解：不能说机器的主频越快，机器的速度就越快。因为机器的速度不仅与主频有关，还与数据通路结构、时序分配方案、ALU 运算能力、指令功能强弱等多种因素有关，要看综合效果。

5. 设机器 A 的主频为 8MHz，机器周期含 4 个时钟周期，且该机的平均指令执行速度是 0.4MIPS，试求该机的平均指令周期和机器周期，每个指令周期中含几个机器周期？如果机器 B 的主频为 12MHz，且机器周期也含 4 个时钟周期，试问 B 机的平均指令执行速度为多少 MIPS？

解：先通过 A 机的平均指令执行速度求出其平均指令周期，再通过主频求出时钟周期，然后进一步求出机器周期。B 机参数的算法与 A 机类似。计算如下：

$$A \text{ 机平均指令周期} = 1/0.4\text{MIPS} = 2.5\mu\text{s}$$

$$A \text{ 机时钟周期} = 1/8\text{MHz} = 125\text{ns}$$

$$A \text{ 机机器周期} = 125\text{ns} \times 4 = 500\text{ns} = 0.5\mu\text{s}$$

$$A \text{ 机每个指令周期中含机器周期个数} = 2.5\mu\text{s} \div 0.5\mu\text{s} = 5 \text{ 个}$$

$$B \text{ 机时钟周期} = 1/12\text{MHz} \approx 83\text{ns}$$

$$B \text{ 机机器周期} = 83\text{ns} \times 4 = 332\text{ns}$$

设 B 机每个指令周期也含 5 个机器周期，则：

$$B \text{ 机平均指令周期} = 332\text{ns} \times 5 = 1.66\mu\text{s}$$

$$B \text{ 机平均指令执行速度} = 1/1.66\mu\text{s} = 0.6\text{MIPS}$$

结论：主频的提高有利于机器执行速度的提高。

6. 设某机主频为 8MHz，每个机器周期平均含 2 个时钟周期，每条指令平均有 4 个机器周期，试问该机的平均指令执行速度为多少 MIPS？若机器主频不变，但每个机器周期平均含 4 个时钟周期，每条指令平均有 4 个机器周期，则该机的平均指令执行速度又是多少 MIPS？由此可得出什么结论？

解：先通过主频求出时钟周期，再求出机器周期和平均指令周期，最后通过平均指令周期的倒数求出平均指令执行速度。计算如下：

$$\text{时钟周期} = 1/8\text{MHz} = 0.125 \times 10^{-6}\text{s}$$

$$\text{机器周期} = 0.125 \times 10^{-6}\text{s} \times 2 = 0.25 \times 10^{-6}\text{s}$$

$$\text{平均指令周期} = 0.25 \times 10^{-6}\text{s} \times 4 = 10^{-6}\text{s}$$

平均指令执行速度= $1/10^{-6}\text{s}=1\text{MIPS}$
 当参数改变后：机器周期= $0.125\times10^{-6}\text{s}\times4=0.5\times10^{-6}\text{s}$
 平均指令周期= $0.5\times10^{-6}\text{s}\times4=2\times10^{-6}\text{s}$
 平均指令执行速度= $1/(2\times10^{-6}\text{s})=0.5\text{MIPS}$
 结论：两个主频相同的机器，执行速度不一定一样。

7. 某 CPU 的主频为 10MHz，若已知每个机器周期平均包含 4 个时钟周期，该机的平均指令执行速度为 1MIPS，试求该机的平均指令周期及每个指令周期含几个机器周期？若改用时钟周期为 0.4μs 的 CPU 芯片，则计算机的平均指令执行速度为多少 MIPS？若要得到平均每秒 80 万次的指令执行速度，则应采用主频为多少的 CPU 芯片？

解：先通过主频求出时钟周期时间，再进一步求出机器周期和平均指令周期。

时钟周期= $1/10\text{MHz}=0.1\times10^{-6}\text{s}$
 机器周期= $0.1\times10^{-6}\text{s}\times4=0.4\times10^{-6}\text{s}$
 平均指令周期= $1/1\text{MIPS}=10^{-6}\text{s}$
 每个指令周期所含机器周期个数= $10^{-6}\text{s}/0.4\times10^{-6}\text{s}=2.5$ 个
 当芯片改变后：机器周期= $0.4\mu\text{s}\times4=1.6\mu\text{s}$
 平均指令周期= $1.6\mu\text{s}\times2.5=4\mu\text{s}$
 平均指令执行速度= $1/4\mu\text{s}=0.25\text{MIPS}$
 若要得到平均每秒 80 万次的指令执行速度，则：
 平均指令周期= $1/0.8\text{MIPS}=1.25\times10^{-6}=1.25\mu\text{s}$
 机器周期= $1.25\mu\text{s}\div2.5=0.5\mu\text{s}$
 时钟周期= $0.5\mu\text{s}\div4=0.125\mu\text{s}$
 CPU 主频= $1/0.125\mu\text{s}=8\text{MHz}$

8. 某计算机的主频为 6MHz，各类指令的平均执行时间和使用频度如下表所示，试计算该机的速度（单位用 MIPS 表示），若上述 CPU 芯片升级为 10MHz，则该机的速度又为多少？

指令类别	存取	加、减、比较、转移	乘除	其它
平均指令执行时间	0.6μs	0.8μs	10μs	1.4μs
使用频度	35%	45%	5%	15%

解：(1)指令平均运行时间 = $(0.6\times0.35+0.8\times0.45+10\times0.05+1.4\times0.15)\mu\text{s}=1.28\mu\text{s}$

机器平均运行速度 = $1/1.28\mu\text{s}\approx0.78\text{MIPS}$

(2)时钟周期 = $1/6\text{MHz}\approx0.167\mu\text{s}$

指令平均运行周期数 = $1.28\mu\text{s}\div0.167\mu\text{s}\approx7.66\text{CPI}$

若 CPU 芯片升级为 10MHz，时钟周期 = $1/10\text{MHz}=0.1\mu\text{s}$

指令平均运行时间 = $0.1\mu\text{s}\times7.66=0.766\mu\text{s}$

机器平均运行速度 = $1/0.766\mu\text{s}\approx1.3\text{MIPS}$

10. 试比较同步控制、异步控制和联合控制的区别。

答：同步控制是指任何一条指令或指令中任何一个微操作的执行都是事先确定的，并且都受同一基准时标的时序信号所控制的方式。异步控制无基准时标信号，

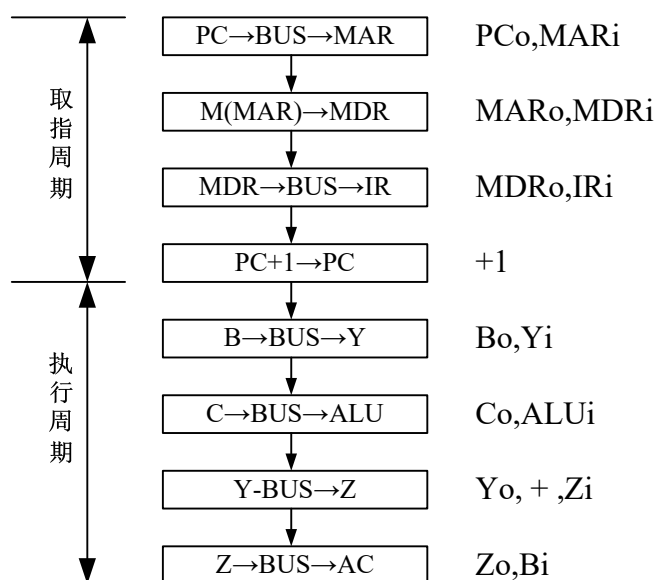
微操作的时序是由专门的应答线路控制，即控制单元发出执行某一微操作的控制信号后，等待执行部件完成了该操作后发回“回答”或“结束”信号，再开始新的微操作。联合控制是同步控制和异步控制相结合的方式，即大多数操作（如 CPU 内部各操作）在同步时序信号的控制下进行，少数时间难以确定的微操作（如涉及 I/O 操作）采用异步控制。

11. 设 CPU 内部结构如图 9.4 所示，此外还设有 B、C、D、E、H、L 六个寄存器，它们各自的输入和输出端都与内部总线相通，并分别受控制信号控制（如 B_i 为寄存器 B 的输入控制； B_o 为 B 的输出控制）。要求从取指令开始，写出完成下列指令所需的全部微操作和控制信号。

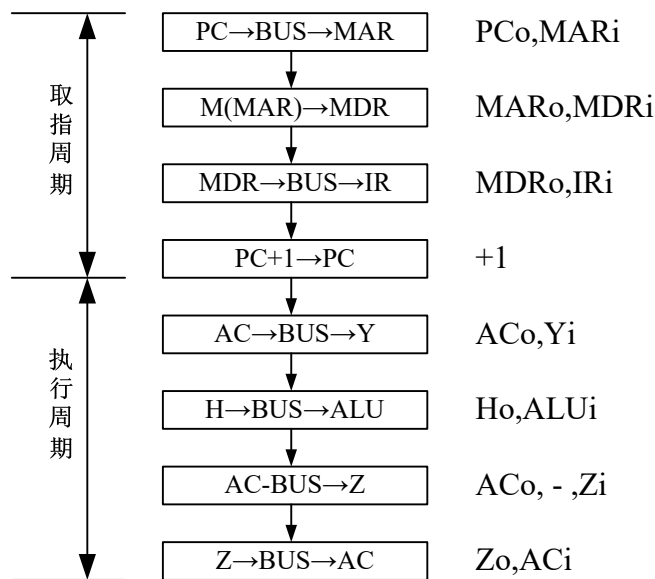
- (1) ADD B, C $((B)+(C) \rightarrow B)$
 (2) SUB A, H $((AC)-(H) \rightarrow AC)$

解：先画出相应指令的流程图，然后将图中每一步数据通路操作分解成相应的微操作，再写出同名的微命令即可。

(1) ADD B, C 指令流程及微命令序列如下：



(2) SUB A, H 指令流程及微命令序列如下：

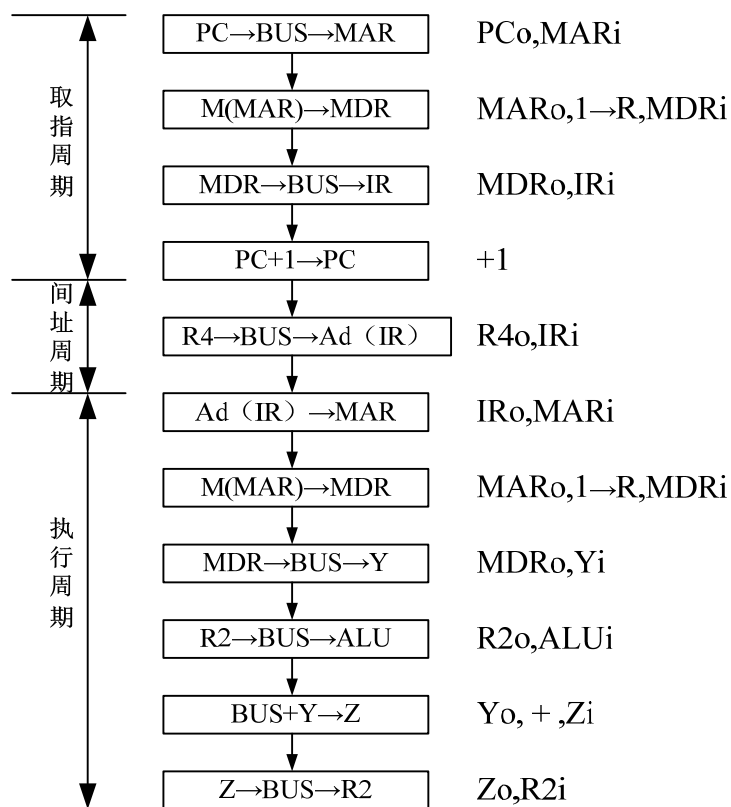


13. 设 CPU 内部结构如图 9.4 所示，此外还设有 R1~R4 四个寄存器，它们各自的输入和输出端都与内部总线相通，并分别受控制信号控制（如 R2i 为寄存器 R2 的输入控制；R2o 为 R2 的输出控制）。要求从取指令开始，写出完成下列指令所需的全部微操作和控制信号。

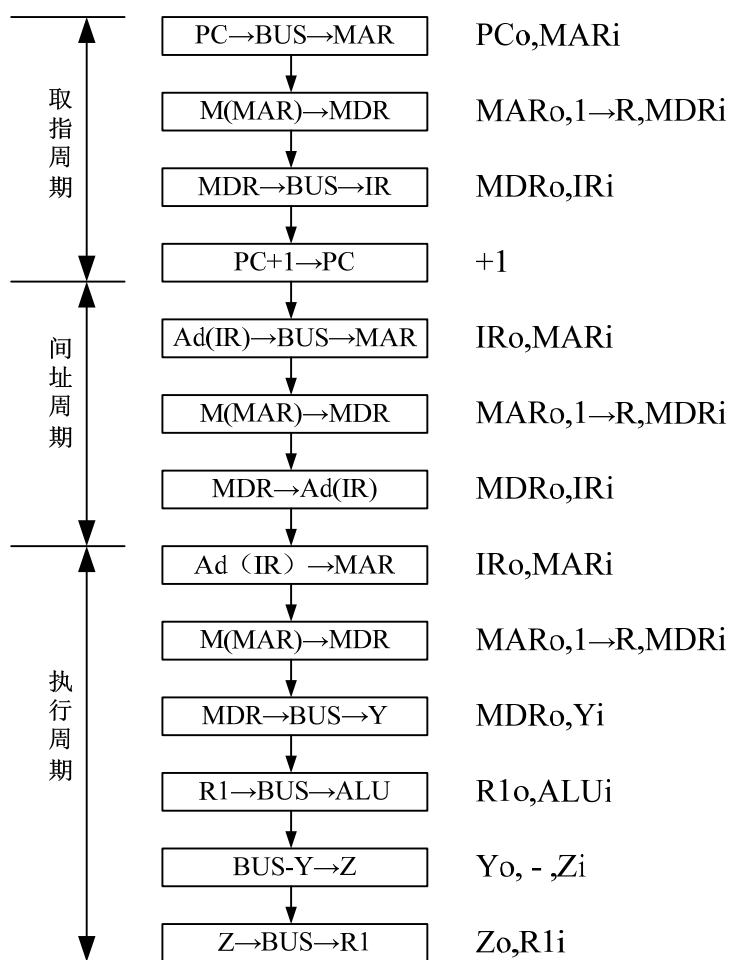
(1) ADD R2, @R4 ; ((R2)+((R4)) → R2, 寄存器间接寻址)

(2) SUB R1, @mem ; ((R1)-((mem)) → R1, 存储器间接寻址)

解：(1) ADD R2, @R4 的指令周期信息流程图及微操作控制信号如下：



(2) SUB R1,@mem 指令周期信息流程图及微操作控制信号如下:



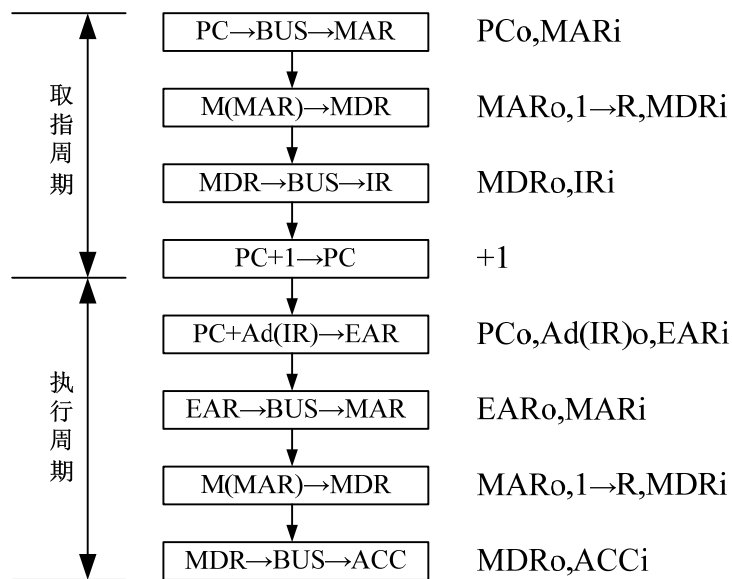
14. 设单总线计算机结构如图 9.5 所示, 其中 M 为主存, XR 为变址寄存器, EAR 为有效地址寄存器, LATCH 为锁存器。假设指令地址已存于 PC 中, 画出“LDA *D”和“SUB X,D”指令周期信息流程图, 并列出相应的控制信号序列。

说明:

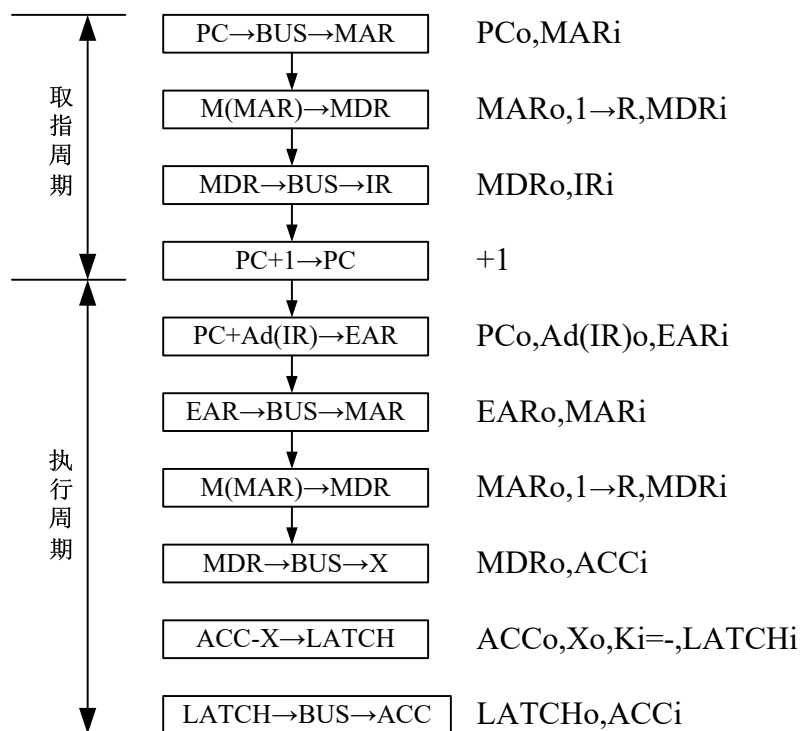
- (1) “LDA *D”指令字中*表示相对寻址, D 为相对位移量。
- (2) “SUB X,D”指令字中 X 为变址寄存器 XR, D 为形式地址。
- (3) 寄存器的输入和输出均受控制信号控制, 例如, PCi 表示 PC 的输入控制信号, MDRo 表示 MDR 的输出控制信号。

(4) 凡是需要经过总线实现寄存器之间的传送, 需要在流程图中注明, 如 PC → Bus → MAR, 相应控制信号为 PCo 和 MARi。

解: (1) “LDA *D”指令周期信息流程图及微操作控制信号如下:



(2) “SUB X,D” 指令周期信息流程图及微操作控制信号如下：



第 10 章 控制单元的设计

1. 假设响应中断时，要求将程序断点存在堆栈内，并且采用软件办法寻找中断服务程序的入口地址，试写出中断隐指令的微操作及节拍安排。

解：设软件查询程序首址为 0 号内存单元，则中断隐指令的微操作命令及节拍安排如下：

T0 0 SP

T1 , SP→MAR

T2 SP→W, SP+1 M (MAR)

T3 PSW→MAR, MDR→SP

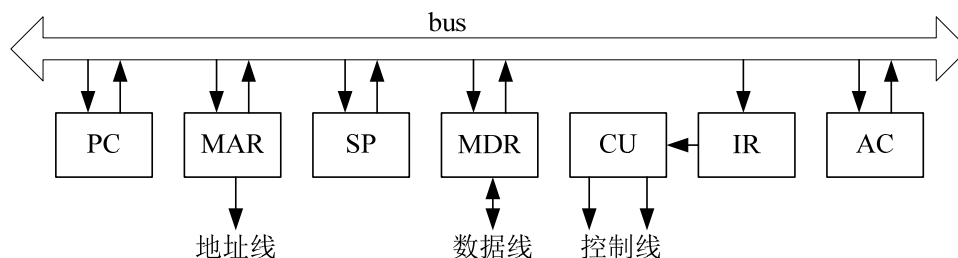
T4 1→W, SP+1→MDR, →M (MAR) PC→EINT 由于题意中没有给出确切的数据通路结构，故上述节拍分配方案的并行性较低。→PC, MDR→0 →MDR, 1→MAR

2. 写出完成下列指令的微操作及节拍安排（包括取指操作）。

(1) 指令 ADD R1, X 完成将 R1 寄存器的内容和主存 X 单元的内容相加，结果存于 R1 的操作。

(2) 指令 ISZ X 完成将主存 X 单元的内容增 1，并根据其结果若为 0，则跳过下一条指令执行。

解：设采用单总线结构的 CPU 数据通路如下图所示，且 ALU 输入端设两个暂存器 C、D（见 17 题图）。并设采用同步控制，每周期 3 节拍：



(1) 指令 ADD R1, X 的微操作及节拍安排如下：

取指周期：T0 PC→MAR, 1→R

T1 M(MAR)→MDR, PC+1→PC

T2 MDR→IR, OP(IR)→ID

执行周期 1：T0 Ad (IR)→MAR, 1→R

T1 M(MAR)→MDR

T2 MDR→D

执行周期 2：T0 R1→C

T1 +

T2 ALU→R1

(2) 指令 ISZ X 的微操作及节拍安排：

取指周期同 (1) :略

执行周期 1：T0 Ad (IR)→MAR, 1→R

T1 M(MAR)→MDR

T2 MDR→C, +1→ALU

执行周期 2：T0 ALU→MDR, 1→W

T1 (PC+1)·Z+ PC· \bar{Z} →PC

12. 能否说水平型微指令就是直接编码的微指令，为什么？

解：不能说水平型微指令就是直接编码的微指令，因为符合水平型微指令特征
的微指令都属于水平型微指令，常见的有：直接编码、字段直接编码、字段间接编
码，及混合编码等。直接编码的微指令只是最典型的一种。

15. 设控制存储器的容量为 512×48 位，微程序可在整个控存空间实现转移，而
控制微程序转移的条件共有 4 个（采用直接控制），微指令格式如下：

解：因为控制存储器共有 $512 \times 48 = 2^9 \times 48$

所以，下址字段应有 9 位，微指令字长 48 位

又因为控制微程序转移的条件有 4 个， $4+1 \leq 2^3$

所以判断测试字段占 3 位

因此控制字段位数为： $48-9-3=36$

微指令格式为：

48	13	12	10	9	1
控制字段		测试字段	下址字段		

21. 下表给出 8 条微指令 I1~I8 及所包含的微命令控制信号，设计微指令操作控
制字段格式，要求所使用的控制位最少，而且保持微指令本身内在的并行性。

解：为使设计出的微指令操作控制字段最短，并且保持微指令本身内在的并行性，
应采用混合编码法。首先找出互斥的微命令组，为便于分析，将微命令表重画如
下：

由表中微命令的分布情况可看出：a、b、c、d、e 微命令的并行性太高，因
此不能放在同一字段中。另外，由分析可知，在 2、3、4 分组的互斥组中，3 个
一组的微命令互斥组对控制位的压缩作用最明显。因此，应尽可能多的找出 3
个一组的互斥组。现找出的互斥组有：cfj, dij, efh, fhi, bgj, ehj, efj.....等等。

从中找出互不相重的互斥组有两个：dij, efh。则：微指令操作控制字段格
式安排如下：

1	1	1	1	2	2
A	b	c	g	efh	dij

各字段编码分配如下：a: 0 无操作，1 a 微命令；

b: 0 无操作，1 b 微命令；

c: 0 无操作，1 c 微命令；

g: 0 无操作，1 g 微命令；

dij : 00 无操作；01 d 微命令；10 i 微命令；11 j 微命令；

efh: 00 无操作；01 e 微命令；10 f 微命令；11 h 微命令

与采用直接控制法比较：直接控制法：10 个微命令需 10 位操作控制位；本
方案中 10 个微命令需 8 位操作控制位，压缩了 2 位。

计算题:

1、设机器数字长为 8 位（含 1 位符号位在内），写出对应下列各真值的原码、补码和反码。

-13/64, 29/128, 100, -87

2、已知[x]补，求[x]原和 x。

1)[x]补=1.1100;

5) [x]补=1,0101;

3、当十六进制数 9BH 和 FFH 分别表示为原码、补码、反码、移码和无符号数时，所对应的十进制数各为多少（设机器数采用一位符号位）？

4、6.12 将十进制数转换为二进制:

x2= -27/1024

5、利用卡诺图法化简下列函数

$$F(ABCD) = \sum (0,1,2,5,6,7,8,9,13,14)$$

$$F(ABCD) = A + \overline{\overline{B}\overline{C}\overline{D}} + BCD$$

6、用原码一位乘和补码一位乘（Booth 算法）计算 $x \cdot y$ 。

(1) $x = 0.110\ 111$, $y = -0.101\ 110$;

7、按机器补码浮点运算步骤计算 $[x+y]$ 补

(1) $x = 2^{-011} \times 0.101100$, $y = 2^{-010} \times (-0.011100)$;

二、存储器部分：

8、假设 CPU 执行某段程序时共访问 Cache 命中 4800 次，访问主存 200 次，已知 Cache 的存取周期是 30ns，主存的存取周期是 150ns，求 Cache 的命中率以及 Cache-主存系统的平均访问时间和效率，试问该系统的性能提高了多少？

9、设 CPU 有 16 根地址线，8 根数据线，用（低电平有效）作访存控制信号，R/ 作读写控制信号（高电平为读，低电平为写），现有 1K×4 位、4K×8 位的 RAM 芯片，2K×8 位，4K×4 位的 ROM 芯片，以及 74138 译码器和各种门电路，画出 CPU 与存储器连接图，图中标明信号线的方向、种类和条数；并写出每片 RAM 芯片的地址范围（用十六进制描述）。

要求：主存地址空间分配：A000H---A7FFH 为系统程序区； A800H---AFFFH 为用户程序区。

10、一个 4 体低位交叉的存储器，假设存取周期为 T，CPU 每隔 1/4 存取周期启动一个存储体，试问依次访问 64 个字需多少个存取周期？

11、cache 映射题

三、指令系统题

12、设指令字长为 16 位，采用扩展操作码技术，每个操作数的地址为 6 位。如果定义了 13 条二地址指令，试问还可安排多少条一地址指令？

13、某机指令字长 16 位，每个操作数的地址码为 6 位，设操作码长度固定，指令分为零地址、一地址和二地址三种格式。若零地址指令有 M 种，一地址指令有 N 种，则二地址指令最多有几种？若操作码位数可变，则二地址指令最多允许有几种？

14、设相对寻址的转移指令占两个字节，第一个字节是操作码，第二个字节是相对位移量，用补码表示。假设当前转移指令第一字节所在的地址为 2000H，且 CPU 每取出一个字节便自动完成 (PC)+1→PC 的操作。试问当执行“JMP *+8”和“JMP *-9”指令时，转移指令第二字节的内容各为多少？补充：操作数的有效地址是多少？

15、某机器指令格式如下所示：

15	10	9	8	7	0
操作码OP			X	位移量D	

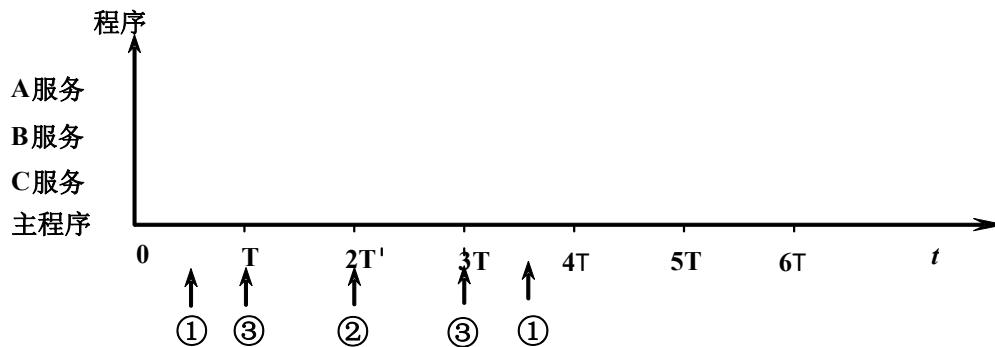
X=00	直接寻址
X=01	寄存器间接寻址，用寄存器 R ₁ 寻址
X=10	变址寻址，用变址寄存器 R ₂ 寻址
X=11	PC 相对寻址

设 (PC)=5431H，(R1)=3525H，(R2)=6783H (H 代表十六进制数)，请确定下列指令的寻址方式和有效地址。(假定主存按字节寻址处理)

- (1) 8341H (2) 1468H (3) 8100H (4) 6264H

IO 系统与中断处理部分

16. 设某机有三个中断源，其优先级次序为 $A > B > C$ ，系统允许多重中断，每个中断源的中断服务程序时间均为 T 。在下图所示时间共发生 5 次中断请求，①为 A 中断源发出的请求信号，②为 A 中断源发出的请求信号，③为 A 中断源发出的请求信号，请画出中断服务程序运行轨迹。



17. 设某机有五个中断源 L_0 、 L_1 、 L_2 、 L_3 、 L_4 ，按中断响应的优先次序由高向低排序为 $L_0 \rightarrow L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow L_4$ ，现要求中断处理次序改为 $L_1 \rightarrow L_4 \rightarrow L_2 \rightarrow L_0 \rightarrow L_3$ ，根据下面的格式，写出各中断源的屏蔽字。

18. 有四个中断源 1#、2#、3#、4#，分属四级中断 1 级、2 级、3 级、4 级，CPU 响应顺序为 1 级 \rightarrow 2 级 \rightarrow 3 级 \rightarrow 4 级，其中 1 级优先级最高，4 级优先级最低（假定 CPU 运行的现行程序优先级最低）。要求 CPU 处理顺序为 2 级 \rightarrow 4 级 \rightarrow 1 级 \rightarrow 3 级。写出各级中断屏蔽字；在某时刻四级中断源同时提出中断请求，画出 CPU 的运行轨迹。

概念问答：

1、存储器的层次结构主要体现在什么地方？为什么要分这些层次？计算机如何管理这些层次？

答：1) 存储器的层次结构主要体现在 Cache—主存和主存—辅存这两个存储层次上。

2) Cache—主存层次主要解决 CPU 和主存速度不匹配的问题，在存储系统中主要对 CPU 访存取加速作用。从 CPU 的角度看，该层次的速度接近于 Cache，而容量和每位价格却接近于主存。这就解决了存储器的高速度和低成本之间的矛盾；

主存—辅存层次主要解决存储系统的容量问题，在存储系统中主要起扩容作用。从程序员的角度看，其所使用的存储器的容量和每位价格接近于辅存，而速度接近于主存。该层次解决了大容量和低成本之间的矛盾。

3) 主存与 Cache 之间的数据调度是由硬件自动完成的，对程序员是透明的。而主存—辅存之间的数据调动，是由硬件和操作系统共同完成的。换言之，即采用虚拟存储技术实现。

2、说明存取周期和存取时间的区别。

答：存取时间是指存储器完成一次读/写所用的时间，即从本次读/写开始至本次读/写结束；而存取周期是指相邻两次读/写操作之间的时间间隔，即从本次读/写开始至下一次读/写开始。因此存取周期要略长于存取时间。

（具体要分读写两种不同操作来分析：

读周期=读出时间+片选失效到地址失效的时间

写周期=滞后时间+写入时间+写恢复时间）

3、试比较静态 RAM 和动态 RAM。

1) 静态 RAM 的特点：依靠双稳态触发器保存信息，不断电信息不丢失；功耗较大，集成度较低，速度快，每位价格高，适合于作 Cache 或存取速度要求较高的小容量主存。

2) 动态 RAM 的特点：依靠电容存储电荷来保存信息，需刷新电路进行动态刷新；功耗较小，集成度高，每位价格较低，适合于作大容量主存。

4、简述动态 RAM 的各种刷新方式及其特点。

答案要点：动态 RAM 的刷新方式有集中式刷新、分散式刷新、异步式刷新和透明式刷新等四种方式。

集中式刷新的特点：在最大刷新闻隔时间内，集中安排一段时间进行刷新。其缺点是进行刷新时必须停止读、写操作。这对主机而言是个“死区”

分散式刷新的特点：刷新工作安排在系统的存取周期内进行，对主机而言不再有“死区”。但该方法加长了系统的存取周期，存在无谓刷新，降低了整机运行效率。因此，分散方式刷新不适用于高速存储器。

异步式刷新的特点：结合了上述两种方式的优点，充分利用了最大刷新闻隔。相对于分散式刷新而言，它减少了刷新次数；相对于集中方式来说，主机的“死区”又缩短很多。因此，这种方式使用得比较多。

透明式刷新的特点：该方式不占用 CPU 时间，对 CPU 而言是透明的操作；但控制线路复杂。

5. 以全相联映射技术为例，说明在带有 Cache 的存储系统中，“读”操作是怎样完成的。

答：当 CPU 发出主存地址后，地址映射机构按照全相联映射方式将主存地址标记与 Cache 所有字块的标记进行比较，以判断出所访主存字（主存地址的内容）是否已在 Cache 中。若命中，直接访问 Cache，将该字送至 CPU；若未命中，一方面要访问主存，将该字传送给 CPU，与此同时，要按照全相联映射方式转换的 Cache 地址将该字所在的主存块装入 Cache，如果此时 Cache 已装满，就要执行替换算法，腾出空位才能将新的主存块调入。

6. 简述 Cache-主存地址映射有哪几种方式，以及各自的优缺点。

答：Cache-主存地址映射有直接映射方式、全相联映射方式和组相联映射方式三种。

直接映射方式的特点：主存的字块只可以和固定的 Cache 字块对应，优点是方式直接，硬件实现电路简单，成本低；缺点是利用率低，同时命中率和效率较低。

全相联映射方式主存中的字块可以和 Cache 的任何字块对应，优点是方式灵活，利用率高；缺点是所需逻辑电路复杂，使用成本太高。

组相联映射方式是对前两种映射方式的折衷，组间全相联，组内直接映像。其特点是集中了两个方式的优点，成本也不太高，是目前应用最为广泛的 Cache 映射方式。

7. 什么是指令周期？指令周期是否有一个固定值？为什么？

解：1) 指令周期是指 CPU 每取出并执行一条指令所需的全部时间。

2) 由于计算机中各种指令执行所需的时间差异很大，因此为了提高 CPU 运行效率，即使在同步控制的机器中，不同指令的指令周期长度都是不一致的，也就是说指令周期对于不同的指令来说不是一个固定值。

3) 指令周期长度不一致的根本原因在于设计人员，为了提高 CPU 运行效率而这样安排的，指令功能不同，需完成的微操作复杂程度亦不同，因此，不同指令的指令周期也不同。

8. 画出指令周期的流程图，分别说明图中每个子周期的作用。

解答：流程图见教材 P343。

取指周期：完成取指令和分析指令的操作。

间址周期：取操作数的有效地址。

执行周期：执行指令的操作。

中断周期：将程序断点保存到存储器。

9. 什么是指令周期、机器周期和时钟周期？三者有何关系？

解：CPU 每取出并执行一条指令所需的全部时间叫指令周期；机器周期是在同步控制的机器中，所有指令执行过程中（执行一步相对完整的操作）的一个基准时间，通常以访问一次存储器所需的时间作为一个机器周期；时钟周期是指计算机主工作时钟的周期时间，它是计算机运行时最基本的时序单位，通常时钟周期=计算机主频的倒数。

三者之间的关系：指令周期常常用若干个机器周期数来表示，机器周期也叫 CPU 周期；而一个机器周期又包含若干个时钟周期（也称为节拍脉冲或 T 周期）。

10.简述微程序控制器的工作原理和工作过程。

这是微程序控制器的工作原理：

将控制器所需要的微操作命令，以微代码的形式编成微指令，存在专门的控制存储器中，CPU 执行机器指令时，从控制存储器中取出微指令，对微指令中的操作控制字段进行解释，即产生执行机器指令所需的微操作命令序列。

微程序控制器的工作过程如下：

首先将用户程序的首地址送至 PC,然后进入取指阶段。

- 1) 取机器指令：从控制存储器中读取“取指微程序”，用产生的微命令控制 CPU 访存，读取机器指令，并送入指令寄存器 IR。
- 2) 形成微程序入口地址：根据机器指令的操作码，通过微地址形成电路，产生与该机器指令对应的微程序入口地址，并送入 CMAR。
- 3) 逐条取出机器指令对应的微程序并执行之。
- 4) 返回取指微指令，开始又一条机器指令的执行。如此不断重复，直到整个程序执行完为止。

11. 试分析比较组合逻辑控制器和微程序控制器的优缺点。

组合逻辑控制器的优点：速度快；

组合逻辑控制器的缺点：设计不规整，指令系统调整和扩充很难；

微程序控制器的优点：①结构规整，设计效率高；②易于修改和扩展指令系统功能；

微程序控制器的缺点：①执行速度较慢；②执行效率不高

12.I/O 设备与主机交换信息时，共有哪几种控制方式？简述它们的特点。

I/O 设备与主机交换信息时，共有 5 种控制方式：程序查询方式、程序中断方式、DMA 方式、I/O 通道方式和 I/O 处理机方式。其中前 3 种方式是基本的且广泛应用的控制方式。

程序查询方式的特点：控制简单，硬件开销小；CPU 与外设是串行工作的，系统效率低。适用于 CPU 不太忙且传送速度要求不太高的场合。

程序中断方式的特点：CPU 和外设可并行工作，提高了 CPU 的效率，不仅适于主机和外设之间的数据交换，还特别适于对外界随机事件的处理。适用于 CPU 较忙，传送速度不太高的系统中，尤其适合实时控制及紧急事件的处理。

DMA 方式的特点：完全由硬件（DMA 控制器）负责完成信息交换，信息传递从以 CPU 为中心，转为以内存为中心，CPU 和外设可并行工作，对高速大批量数据传送特别有用。但缺点是只能进行简单数据交换，电路结构复杂，硬件开销大。

13.CPU 响应中断的条件是什么？

CPU 响应中断的条件可以归纳为三条：

- 1) 有中断请求；
- 2) CPU 允许中断，即中断允许状态 IF=1（或 EINT=1）；
- 3) 一条指令执行结束。

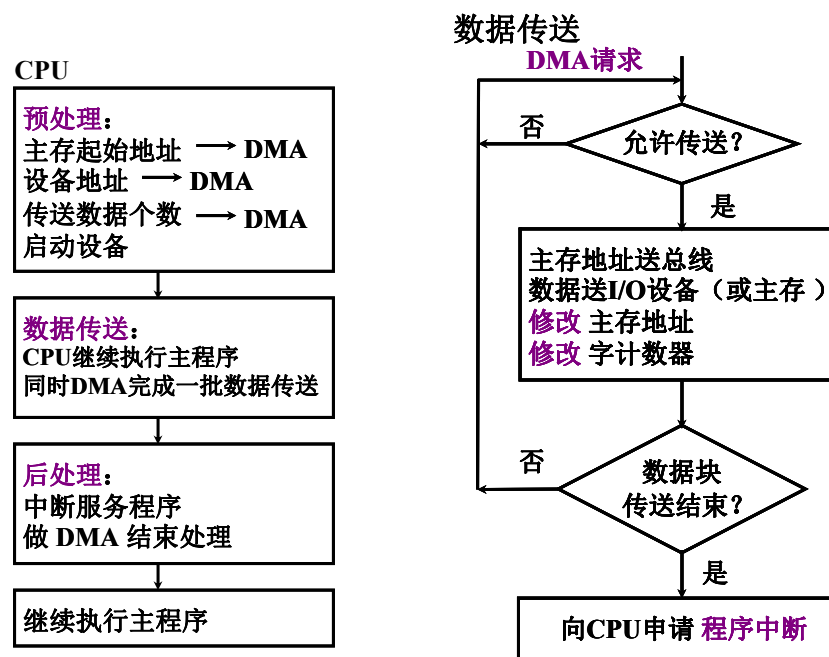
14. 试比较单重中断和多重中断服务程序的处理流程，说明它们不同的原因。

1) 二者的比较可用两种中断的服务程序流程图（见教材 P201）的对比来说明，此处略。

2) 单重中断和多重中断的区别在于“开中断”的设置时间不同。对于单重中断，开中断指令设置在最后“中断返回”之前，意味着在整个中断服务处理过程中，不能再响应其他中断源的请求。而对于多重中断，开中断指令提前至“保护现场”之后，意味着在保护现场之后，若有更高级别的中断源提出请求，CPU 也可以响应，从而实现中断嵌套，这是二者的主要区别。

15. 结合 DMA 接口电路说明其工作过程。

DMA 的数据传送过程可分为预处理、数据传送和后处理 3 个阶段。工作过程如下图所示：



各阶段完成的工作如下：

- 1) 预处理阶段：CPU 执行主程序实现 DMA 传送的初始化设置；
- 2) 数据传送阶段：由 DMA 控制器实现内存和外设间的数据传送。
- 3) 后处理阶段：中断处理程序判断传送的正误，对写入主存的数据进行校验，完成善后工作。

16. 以硬盘读写为例，说明在主机和外设之间进行数据传送，为什么需要采用 DMA 方式？

参考答案要点：

一些高速外设，如硬盘、光盘等 I/O 设备，经常需要和主存进行大批量的数据交换；若采用程序查询方式或程序中断方式来完成，即通过 CPU 执行程序来完成数据交换，速度较慢，极可能造成数据的丢失，因而不能满足批量数据的高速传递需求。因此，需要借助于硬件，比如 DMA 控制器来实现主存和高速外设之间的直接数据传送。

17. 中断周期前是什么阶段？中断周期后又是什么阶段？在中断周期 CPU 应完成什么操作？

- 1) 中断周期前是一条指令的执行周期；
- 2) 中断周期后是取指周期（取中断服务程序的第 1 条指令）；
- 3) 中断周期中，CPU 由硬件（中断隐指令）完成如下操作：

- ①保护程序的断点；
- ②关中断；
- ③转中断服务程序入口。

计组

1 假设阶码取 3 位, 尾数取 6 位 (均不包括符号位), 计算 $[2-3 \times (13/16)] - [2-4 \times (-5/8)]$ 的值 (要求写出详细计算过程)

2 补码一位乘(booth 法)要会计算和画框图。

3 设主存容量为 1MB, 采用直接映射方式的 Cache 容量为 16KB, 块长为 4, 每字 32 位。试问主存地址为 ABCDEH 的存储单元在 Cache 中的什么位置?

答案要点:

主存地址 ABCDEH, Cache 的地址为 14 位, 其中字块内地址为 4 位, Cache 字块地址为 10 位。采用直接映射方式, 只要将主存地址后 14 位地址与 Cache 的 14 位地址相同就行了。因此:

ABCDEH=1010 1011 1100 1101 1110B, 其中, 后 14 位地址为 11 1100 1101 1110, 用十六进制描述为 3CDEH, 这就是指定的主存单元在 Cache 中的位置。

4 设某机主存容量为 4MB, Cache 容量为 16KB, 每字块有 8 个字, 每字 32 位, 设计一个四路组相联映射 (即 Cache 每组内共有 4 个字块) 的 Cache 组织。

(1) 画出主存地址字段中各段的位数;

(2) 速度提高倍数和效率区分开

5 试分析比较 DMA 方式和中断方式的异同, DMA 方式能否取代中断方式, 为什么?

6 现有 A、B、C、D 四个中断源，其优先级由高向低按 A→B→C→D 顺序排列。

若中断服务程序的执行时间为 $20\mu s$ ，请根据下图所示

时间轴给出的中断源请求中断的时刻，画出 CPU 执行程序的轨迹。

7 SUB R1, @mem

取指周期：

(PC) → Bus → MAR PCO, MARi

1→R M(R) = 1(对内存的读命令有效)

M(MAR) → Bus → MDR MDRI

(MDR) → Bus → IR MDRO, IRi

(PC) + 1 → PC PCO, PCi

OP(IR) → CU 无控制信号

间址周期：

Ad(IR) → Bus → MAR MARi

1→R M(R) = 1(对内存的读命令有效)

M(MAR) → Bus → MDR MDRI

执行周期

MDR → Bus → MAR MDRO, MARi

1→R M(R) = 1(对内存的读命令有效)

M(MAR) → Bus → MDR MDRI

MDR → Bus → Y MDRO, Yi

(R1)-(Y) → Z R1O, Yo, " -" , zi

(Z)→Bus → R1

zO, R1i

8 说明微指令和微操作、微程序和机器指令、微程序和程序之间的关系。

参考答案要点：

1) 微操作是指由微命令控制实现的最基本操作，微指令是微操作控制信号代码化的结果，一条微指令对应着一个或几个微操作命令（即微命令）。

2) 微程序是一系列微指令的有序集合。在微程序控制的计算机中，一条机器指令的功能通常是由一段微程序完成的。微程序存放在控存中，而机器指令属于机器的指令集，由机器指令构成的程序存放在内存中。

3) 微程序是由计算机的设计人员编写的，存放在控制存储器中，只能读出，不能修改。而程序是一系列机器指令的有序集合，用于实际问题，可由普通程序员编写，存放在主存中，可以更新修改。

9 计算机是如何区分存储器中的指令和数据的？

参考答案要点：

CPU 可从时间和空间两个层面来区分访存取来的指令和数据。

1) 时间层面：在取指周期（或运行取指微程序）内，由 PC 提供访存地址，取来的即为指令；在执行周期（或运行执行周期相对应的微程序段）内，由指令的地址码部分提供访存地址，取来的即为操作数，也就是数据。

2) 空间层面：取来的机器指令应存放在指令寄存器，而取来的数据（或操作数）则应该存放在以累加器为代表的通用寄存器内。

10

7.4.3 指令格式设计举例

例 7.4 某机字长 16 位，存储器直接寻址空间为 128 字，变址时的位移量为 $-64 \sim +63$ ，16 个通用寄存器均可作为变址寄存器。设计一套指令系统格式，满足下列寻址类型的要求。

（1）直接寻址的二地址指令 3 条。 $16 - 2 \times 7 = 2$ 操作码 00 01 10 3 条

（2）变址寻址的一地址指令 6 条。 $16 - 11 = 5$ 11000 --- 11101 6 条

（3）寄存器寻址的二地址指令 8 条。 $16 - 8 = 8$ 11110000 - 11111111 8 条

（4）直接寻址的一地址指令 12 条。 $16 - 7 = 9$ 11111000 - 11111111 12 条

（5）零地址指令 32 条。

试问还有多少种代码未用？若安排寄存器寻址的一地址指令，还能容纳多少条？

根据题目给出直接寻址空间为 128 字，则每个地址码

11 存储器字扩展和位扩展，课本例题理解且能自己画出来就好了，难度足够了。