目录

**[第1章 计算机系统概论 4](#_Toc8440)**

[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？ 4](#_Toc2004)

[3. 说明高级语言、汇编语言和机器语言的差别及其联系。 5](#_Toc18086)

[4. 如何理解计算机组成和计算机体系结构？ 5](#_Toc23790)

[5. 冯•诺依曼计算机的特点是什么？ 5](#_Toc23312)

[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。 6](#_Toc12841)

[7. 解释下列概念： 7](#_Toc15116)

[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。 7](#_Toc11724)

[8. 解释下列英文缩写的中文含义： 8](#_Toc17795)

[CPU、PC、IR、CU、ALU、ACC、MQ、X、MAR、MDR、I/O、MIPS、CPI、FLOPS 8](#_Toc9704)

[10. 指令和数据都存于存储器中，计算机如何区分它们？ 10](#_Toc27003)

**[第2章 计算机的发展及应用 11](#_Toc23904)**

[1. 通常计算机的更新换代以什么为依据？ 11](#_Toc23681)

[2. 举例说明专用计算机和通用计算机的区别。 11](#_Toc27810)

[3. 什么是摩尔定律？该定律是否永远生效？为什么？ 11](#_Toc31205)

**[第3章 系统总线 12](#_Toc9886)**

[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？ 12](#_Toc15004)

[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。 12](#_Toc10517)

[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。 13](#_Toc13182)

[6. 试比较同步通信和异步通信。 14](#_Toc21299)

[7. 画图说明异步通信中请求与回答有哪几种互锁关系？ 14](#_Toc13814)

[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？ 14](#_Toc3066)

[9. 分离式通讯有何特点，主要用于什么系统？ 14](#_Toc817)

[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？ 15](#_Toc5431)

[11. 画一个具有双向传输功能的总线逻辑图。 15](#_Toc3544)

[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： 16](#_Toc5834)

[13. 什么是总线的数据传输率，它与哪些因素有关？ 17](#_Toc4648)

**[第4章 存储器 19](#_Toc25123)**

[1. 解释概念：主存、辅存、Cache、RAM、SRAM、DRAM、ROM、PROM、EPROM、EEPROM、CDROM、Flash Memory。 19](#_Toc15105)

[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。 20](#_Toc20329)

[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？ 20](#_Toc2268)

[4. 说明存取周期和存取时间的区别。 21](#_Toc15725)

[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？ 21](#_Toc28203)

[8. 试比较静态RAM和动态RAM。 22](#_Toc23824)

[9. 什么叫刷新？为什么要刷新？说明刷新有几种方法。 23](#_Toc1972)

[10. 半导体存储器芯片的译码驱动方式有几种？ 23](#_Toc29117)

[17. 写出1100、1101、1110、1111对应的汉明码。 29](#_Toc17904)

[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？ 35](#_Toc25055)

[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？ 35](#_Toc5673)

[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？ 36](#_Toc11257)

[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？ 36](#_Toc12718)

[33．简要说明提高访存速度可采取的措施。 41](#_Toc28047)

**[第5章 输入输出系统 44](#_Toc11169)**

[1. I/O有哪些编址方式？各有何特点？ 44](#_Toc21054)

[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？ 44](#_Toc23073)

[3. I/O设备与主机交换信息时有哪几种控制方式？简述它们的特点？ 45](#_Toc13495)

[4. 比较程序查询方式，程序中断方式，和DMA方式对CPU工作效率的影响？ 45](#_Toc24036)

[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？ 46](#_Toc18558)

[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？ 46](#_Toc25467)

[12. 结合程序查询方式的接口电路，说明其工作过程。 47](#_Toc6802)

[13. 说明中断向量地址和入口地址的区别和联系。 47](#_Toc21459)

[14. 在什么条件下，I/O设备可以向CPU提出中断请求？ 48](#_Toc26734)

[15. 什么是中断允许触发器？它有何作用？ 48](#_Toc5559)

[16. 在什么条件和什么时间，CPU可以响应I/O的中断请求？ 48](#_Toc2051)

[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？ 49](#_Toc20568)

[22.程序查询方式和程序中断方式都是通过“程序”传送数据，两者的区别是什么？ 49](#_Toc12447)

[25 根据以下要求设计一个产生3个设备向量地址的电路。 50](#_Toc24362)

[26. 什么是多重中断？实现多重中断的必要条件是什么？ 50](#_Toc13270)

[27. DMA方式有何特点？什么样的I/O设备与主机交换信息时采用DMA方式，举例说明。 51](#_Toc27033)

[28. CPU对DMA请求和中断请求的响应时间是否一样？为什么？ 51](#_Toc28394)

[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。 52](#_Toc25947)

[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。 54](#_Toc16457)

**[第6章 计算机的运算方法 59](#_Toc16497)**

[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 59](#_Toc27154)

[5. 已知[x]补，求[x]原和x。 60](#_Toc6164)

[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]补=[x]原成立。    60](#_Toc20939)

[7. 若x< 0，当x= -64时，[x]补=[x]原=1,100 0000，则 [x]补=[x]原成立。 设x为真值，x\*为绝对值，说明[-x\*]补=[-x]补能否成立。 61](#_Toc19488)

[8. 讨论若[x]补>[y]补，是否有x>y？     61](#_Toc20133)

[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）？ 61](#_Toc31417)

[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？ 62](#_Toc771)

[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。 62](#_Toc3649)

[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： 63](#_Toc25379)

[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？ 65](#_Toc6105)

[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？     66](#_Toc23060)

[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。 66](#_Toc17149)

[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。 68](#_Toc18274)

[18. 试比较逻辑移位和算术移位。     70](#_Toc21251)

[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。 70](#_Toc8992)

[21. 用原码加减交替法和补码加减交替法计算x÷y。 74](#_Toc24959)

[26.按机器补码浮点运算步骤，计算[x±y]补. 76](#_Toc17201)

**[第7章 指令系统 80](#_Toc3291)**

[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与硬件结构之间存在着密切的关系？ 80](#_Toc21515)

[2. 答：参考P300。什么叫寻址方式？为什么要学习寻址方式？ 80](#_Toc167)

[3. 答：参看P310。什么是指令字长、机器字长和存储字长？ 80](#_Toc26684)

[4. 答：略。零地址指令的操作数来自哪里？？各举一例说明。 80](#_Toc19820)

[5. 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。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。 80](#_Toc23937)

[9. 试比较间接寻址和寄存器间接寻址。 81](#_Toc29699)

[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： 84](#_Toc2864)

**[第8章 CPU的结构和功能 85](#_Toc26816)**

[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。 85](#_Toc3386)

[2. 什么是指令周期？指令周期是否有一个固定值？为什么？ 85](#_Toc13563)

[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。 85](#_Toc22559)

[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？ 87](#_Toc20910)

[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？ 87](#_Toc25980)

[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？ 87](#_Toc27723)

[17. 在中断系统中INTR、INT、EINT三个触发器各有何作用？ 88](#_Toc1573)

**[第9章 控制单元的功能 91](#_Toc3828)**

[1. 设CPU内有这些部件：PC、IR、MAR、MDR、AC、CU。 91](#_Toc3554)

[2. 控制单元的功能是什么？其输入受什么控制？ 92](#_Toc12215)

[3. 什么是指令周期、机器周期和时钟周期？三者有何关系？ 92](#_Toc28236)

[4. 能不能说机器的主频越快，机器的速度就越快，为什么？ 92](#_Toc1670)

[9.试比较同步控制、异步控制和联合控制的区别。 95](#_Toc16733)

**[第10章 控制单元的设计 101](#_Toc18425)**

[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。 101](#_Toc2477)

[12. 能否说水平型微指令就是直接编码的微指令，为什么？ 102](#_Toc4901)

# 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件，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和数据的存储器，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“STA M”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）STA M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：Gordon Moore 提出的“微芯片上集成的晶体管的数目每3年翻两番”的规律﹐这就是人们常称的 Moore(摩尔）定律。

该定律不能永远生效。原因有:其一，硅芯片的集成度要受其物理极限的制约，集成度不可能永无止境地提高﹔其二由于芯片集成度提高时，其制作成本也在不断提高﹔其三，随着集成度的提高，微处理器内部的功耗、散热﹑线延迟等一系列的问题将难以解决。所以该定律不能永久生效。

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：见P52-55。

解:总线结构通常有单总线结构和多总线结构。

( 1）单总线结构是将CPU、主存、IO设备（通过I/o接口〉都挂在一组总线上，允许IO 设备之间、Io 设备与CPU之间或I/o设备与主存之间直接交换信息○这种结构简单，也便于扩充，但是所有的传送都通过这组共享总线，易形成瓶颈。这类总线多数为小型机或微型机所采用。

(2)多总线结构又分为双总线结构、三总线结构和四总线结构。

将速率不同的 IO设备进行分类，然后将它们连接在不同的通道〈具有特殊功能的处理器〉上，那么计算机系统的利用率将会更高，由此发展成多总线结构。

**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。

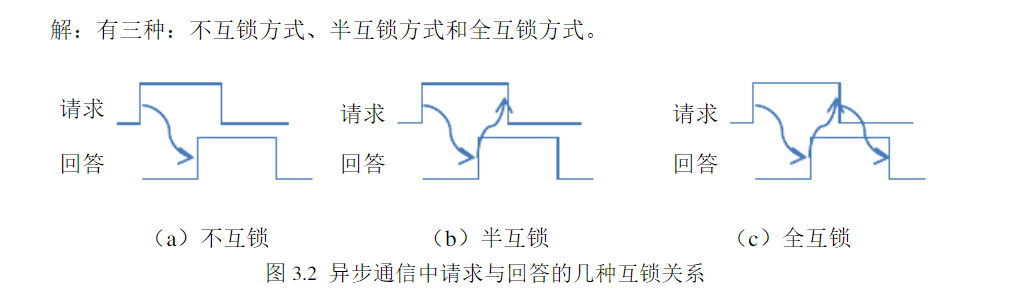
## 6. 试比较同步通信和异步通信。

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→A、D→B和D→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→总线；

T1时刻完成总线→A；

T2时刻完成A→总线；

T3时刻完成总线→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T0＋T1控制，以确保T0时刻D→总线，以及T1时刻总线→接收门1→A。三态门2受T2＋T3控制，以确保T2时刻A→总线，以及T3时刻总线→接收门2→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=8MHz,T=1/f=1/8M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：

存取周期 = 存取时间 + 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。

存储器带宽 = 1/200ns ×32位 = 160M位/秒 = 20MB/秒 = 5M字/秒

**注意**：字长32位，不是16位。（注：1ns=10-9s）

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：64K / （32/8）= 16K

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：

1K×4：（16K×32） / （1K×4） = 16×8 = 128片

2K×8：（16K×32） / （2K×8） = 8×4 = 32片

4K×4：（16K×32） / （4K×4） = 4×8 = 32片

16K×1：（16K×32）/ （16K×1） = 1×32 = 32片

4K×8：（16K×32）/ （4K×8） = 4×4 = 16片

8K×8：（16K×32） / （8K×8） = 2×4 = 8片

## 8. 试比较静态RAM和动态RAM。

答：



## 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：在最大刷新间隔时间内，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×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0.1μs。试问采用集中刷新、分散刷新和异步刷新三种方式的刷新间隔各为多少？

解：采用分散刷新方式刷新间隔为:2ms，其中刷新死时间为：256×0.1μs=25.6μs

采用分散刷新方式刷新间隔为：256×（0.1μs+×0.1μs）=51.2μs

采用异步刷新方式刷新间隔为:2ms

12. 画出用1024×4位的存储芯片组成一个容量为64K×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×8位） / （1024×4位）= 64×2 = 128片

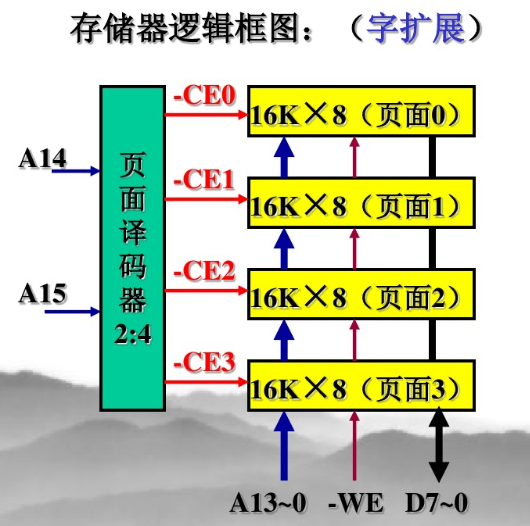
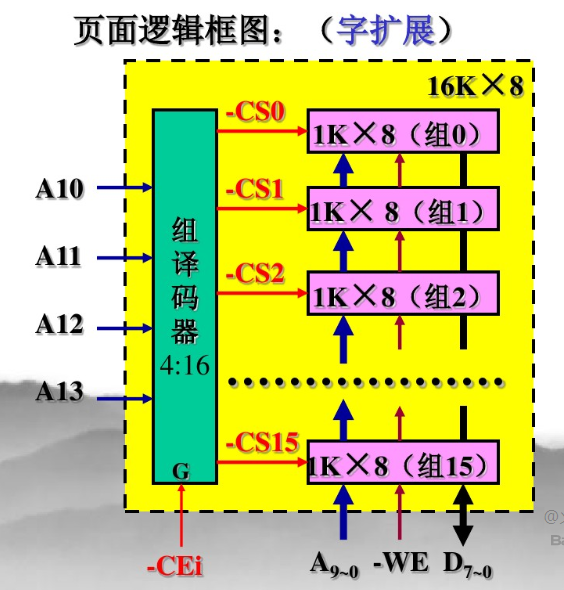
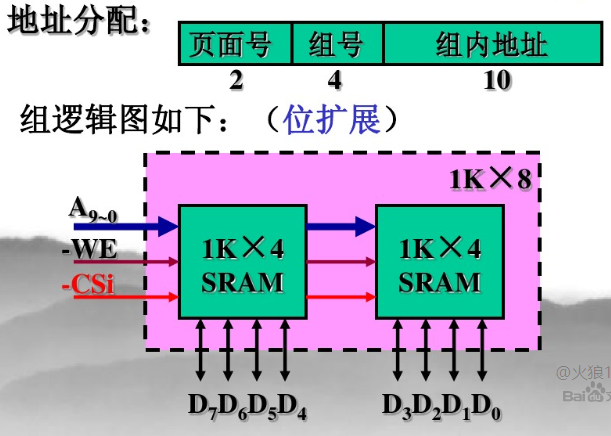
题意分析：本题设计的存储器结构上分为总体、页面、组三级，因此画图时也应分三级画。首先应确定各级的容量：

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

组容量 = 页面容量 / 组数   = 16K×8位 / 16 = 1K×8位，16片1K×8位字串联成16K×8位

组内片数 = 组容量 / 片容量 = 1K×8位 / 1K×4位 = 2片，两片1K×4位芯片位并联成1K×8位

存储器逻辑框图：（略）。



13. 设有一个64K×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和数据线，并说明有几种解答。

解：存储基元总数 = 64K×8位 = 512K位 = 219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：2a×b = 219；b = 219-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。

14. 某8位微型机地址码为18位，若使用4K×4位的RAM芯片组成模块板结构的存储器，试问：

（1）该机所允许的最大主存空间是多少？

（2）若每个模块板为32K×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？

解：（1）该机所允许的最大主存空间是：218 × 8位 = 256K×8位 = 256KB

（2）模块板总数 = 256K×8 / 32K×8 = 8块

（3）板内片数 = 32K×8位 / 4K×4位 = 8×2 = 16片

（4）总片数 = 16片×8 = 128片

（5）CPU通过最高3位地址译码输出选择模板，次高3位地址译码输出选择芯片。地址格式分配如下：



15. 设CPU共有16根地址线，8根数据线，并用（低电平有效）作访存控制信号，作读写命令信号（高电平为读，低电平为写）。现有下列存储芯片：ROM（2K×8位，4K×4位，8K×8位），RAM（1K×4位，2K×8位，4K×8位），及74138译码器和其他门电路（门电路自定）。试从上述规格中选用合适芯片，画出CPU和存储芯片的连接图。要求：

（1）最小4K地址为系统程序区，4096~16383地址范围为用户程序区。

（2）指出选用的存储芯片类型及数量。

（3）详细画出片选逻辑。

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)

  （3）各芯片二进制地址分配如下：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| ROM1,2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| RAM1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| RAM2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| RAM3 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

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×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）该片的端与端错连或短路；

2）该片的端与CPU的端错连或短路；

3）该片的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b4b3b2b1表示

校验位位数k=3位，（2k>=n+k+1）

设校验位分别为c1、c2、c3，则汉明码共4+3=7位，即：c1c2b4c3b3b2b1

校验位在汉明码中分别处于第1、2、4位

c1=b4⊕b3⊕b1

c2=b4⊕b2⊕b1

c3=b3⊕b2⊕b1

当有效信息为1100时，c3c2c1=110,汉明码为0111100。

当有效信息为1101时，c3c2c1=001,汉明码为1010101。

当有效信息为1110时，c3c2c1=000,汉明码为0010110。

当有效信息为1111时，c3c2c1=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：c1’c2’b4’c3’b3’b2’b1’

纠错过程如下：

P1=c1’⊕b4’⊕b3’⊕b1’

P2=c2’⊕b4’⊕b2’⊕b1’

P3=c3’⊕b3’⊕b2’⊕b1’

如果收到的汉明码为1100100，则p3p2p1=011，说明代码有错，第3位（b4’）出错，有效信息为：1100

如果收到的汉明码为1100111，则p3p2p1=111，说明代码有错，第7位（b1’）出错，有效信息为：0110

如果收到的汉明码为1100000，则p3p2p1=110，说明代码有错，第6位（b2’）出错，有效信息为：0010

如果收到的汉明码为1100001，则p3p2p1=001，说明代码有错，第1位（c1’）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’⊕B4’⊕B3’⊕B1’

P2=C2’⊕B4’⊕B2’⊕B1’

P3=C3’⊕B3’⊕B1’

（1）如接收到的汉明码为1100000，

P1=1⊕0⊕0⊕0=1

P2=1⊕0⊕0⊕0=1

P3=0⊕0⊕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⊕0⊕0⊕0=1

P2=1⊕0⊕1⊕0=0

P3=0⊕0⊕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⊕0⊕0⊕1=0

P2=1⊕0⊕0⊕1=0

P3=1⊕0⊕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’⊕B4’⊕B3’⊕B1’⊕1

P2=C2’⊕B4’⊕B2’⊕B1’⊕1

P3=C3’⊕B3’⊕B1’⊕1

（4）如接收到的汉明码为0011001，

P1=0⊕1⊕0⊕1⊕1=1

P2=0⊕1⊕0⊕1⊕1=1

P3=1⊕0⊕1⊕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⊕0⊕0⊕0⊕1=0

P2=0⊕1⊕0⊕0⊕1=0

P3=0⊕0⊕0⊕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⊕1⊕0⊕1⊕1=0

P2=1⊕1⊕0⊕1⊕1=0

P3=0⊕0⊕1⊕1=0

P3P2P1=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>=n+k+1，k=4，进行奇校验设校验位为C1C2C3C4，汉明码为C1C2B7C3B6B5B4C4B3B2B1，

C1=1⊕B7⊕B6⊕B4⊕B3⊕B1=1⊕1⊕0⊕1⊕1⊕1=1

C2=1⊕B7⊕B5⊕B4⊕B2⊕B1=1⊕1⊕0⊕1⊕0⊕1=0

C3=1⊕B6⊕B5⊕B4=1⊕0⊕0⊕1=0

C4=1⊕B3⊕B2⊕B1=1⊕1⊕0⊕1=1

故传送的汉明码为10100011101，若第6位(B5)出错，即接收的码字为10100111101，则

P1=1⊕C1’⊕B7’⊕B6’⊕B4’⊕B3’⊕B1’=1⊕1⊕1⊕0⊕1⊕1⊕1=0

P2=1⊕C2’⊕B7’⊕B5’⊕B4’⊕B2’⊕B1’=1⊕0⊕1⊕1⊕1⊕0⊕1=1

P3=1⊕C3’⊕B6’⊕B5’⊕B4’=1⊕0⊕0⊕1⊕1=1

P4=1⊕C4’⊕B3’⊕B2’⊕B1’=1⊕1⊕1⊕0⊕1=0

P4P3P2P1=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P4P2P1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，P4P2P1=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，P4P2P1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P4P2P1中至少两位为1，如B1出错，将使P4P1均为1，P2=0,P4P2P1=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），（低电平有效）为写命令，（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。

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（A0-A12），数据线8根（D0-D7），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，τ=T/4，依次访问64个字所需时间为：

t=T+(64-1) τ=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=211/22=29=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=218字，主存地址共18位，共分256K/4=216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=27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\*32/4=221B，

Cache容量为2K\*32/4=214B，块长为4\*32/4=32B=25B，字块内地址为5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（5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（5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（5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： ta=0.96\*30ns+(1-0.96)\*150ns=34.8ns

Cache-主存系统的访问效率为：e=tc/ta\*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=213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=219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=214B，地址共格式为14位，分为16KB/(4\*32/8B)=210块，每块4\*32/8=16B=24B，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=220B，地址共格式为20位，分为1MB/(4\*32/8B)=216块，每块24B，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=214B，字块大小为8\*32/8=32=25B，得Cache地址共14位，Cache共有214-5=29块。

根据四路组相联映射，Cache共分为29/22=27组。

根据主存容量为4MB=222B，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：



（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\*90\*8，有Cache的访问时间为t（90\*8-12）+6t\*12，则有Cache和无Cache相比，速度提高的倍数为：



## 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：6×2=12个存储面可用。

  （2）有效存储区域=（33-22）/ 2 = 5.5cm

        柱面数 = 40道/cm × 5.5= 220道

  （3）内层道周长=π×22=69.08cm

       道容量=400位/cm×69.08cm= 3454B

      面容量=3454B×220道=759，880B

      盘组总容量=759，880B ×12面= 9，118，560B

（4）转速 = 3600转 / 60秒 = 60转/秒

  数据传输率 = 3454B × 60转/秒 = 207，240 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道×12 288B/道×4面 = 13 516 800B

（2）最高位密度 = 12 288B/（π×230）= 17B/mm = 136位/mm（向下取整）

   最大磁道直径=230mm+2×275道/(5道/mm) = 230mm + 110mm = 340mm

  最低位密度 = 12 288B /(π×340)= 11B/mm = 92位 / mm （向下取整）

（3）磁盘数据传输率= 12 288B × 3000转/分=12 288B × 50转/秒=614 400B/s

（4）平均等待时间 = 1s/50 / 2 = 10ms

# 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地址编码，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标进行联络，适用于CPU与I/O速度差不大，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。

## 3. I/O设备与主机交换信息时有哪几种控制方式？简述它们的特点？

答：5种控制方式：程序查询，程序中断，DMA，I/O通道，I/O处理机。特点如下，

（1）程序查询方式：CPU通过程序不断查询I/O设备是否准备好，I/O接口上必须设有状态标记以供CPU查询；I/O设备与CPU处于串行工作状态，CPU存在踏步现象，工作效率低。

（2）程序中断方式：CPU在启动I/O设备后继续执行自身程序；只有当外设准备好，向CPU发出中断请求且CPU响应后，才中断当前执行程序，转去处理中断请求，中断处理完再返回原程序。这种方式消除踏步现象，只有在中断处理时占用CPU，CPU工作效率较高。

（3）DMA方式：主存与外设之间有一条数据通道；主存与外设交换数据时无需CPU介入，而由DMA控制器来控制；CPU将总线使用权让给DMA控制器。

（4）通道方式。通道是一个具有特殊功能的处理器，CPU把部分权力下放给通道，由它实现对外围设备的统一管理和外围设备与主存之间的数据交换，大大提高了CPU的效率，但它是以花费更多的硬件为代价。

（5）I/O处理机方式。它是通道方式的进一步发展，CPU将I/O操作及外围设备的管理权全部交给I/O处理机，其实质是多机系统，因而效率有更大提高

答案参见P162~165。

## 4. 比较程序查询方式，程序中断方式，和DMA方式对CPU工作效率的影响？

答：程序查询方式：I/O设备准备时，CPU通过程序不断查询I/O设备是否准备好；I/O设备与CPU处于串行工作状态，CPU工作效率低。

程序中断方式： 当I/O设备准备时无需占用CPU，CPU继续执行当前程序，这时CPU与I/O设备并行工作；当外设准备好发送(接收)数据时，向CPU发出中断请求时，才需要CPU，CPU处理中断并转去执行中断服务子程序时，中断处理完CPU返回断点处继续执行原程序。与程序查询方式相比，CPU工作效率高很多。

DMA方式：外设在准备数据以及与主存交换数据时，均无需占用CPU，数据交换由DMA控制器来控制，而CPU将总线使用权让给DMA控制器。在数据交换时，CPU虽不能访存，但可以做一些内部操作。在数据交换结束时，需要CPU以中断方式介入进行数据后处理工作。3种方式中，DMA方式下CPU的工作效率最高。

## 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：480×10=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：

    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，中断入口地址是中断服务程序首址。

    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×N+Q秒

  平均每个数据所需处理时间= （P×N+Q）/N秒

  求倒数得：该系统跟踪到的每秒中断请求数=N/（P×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。

## 22.程序查询方式和程序中断方式都是通过“程序”传送数据，两者的区别是什么？

答：程序查询方式通过“程序”传送数据时，程序对I/O的控制包括了I/O准备和I/O传送两段时间。由于I/O的工作速度比CPU低得多，因此程序中要反复询问I/O的状态，造成“踏步等待”，严重浪费了CPU的工作时间。 而程序中断方式虽然也是通过“程序”传送数据，但程序仅对I/O传送阶段进行控制，I/O准备阶段不需要CPU查询。故CPU此时照样可以运行现行程序，与I/O并行工作，大大提高了CPU的工作效率。

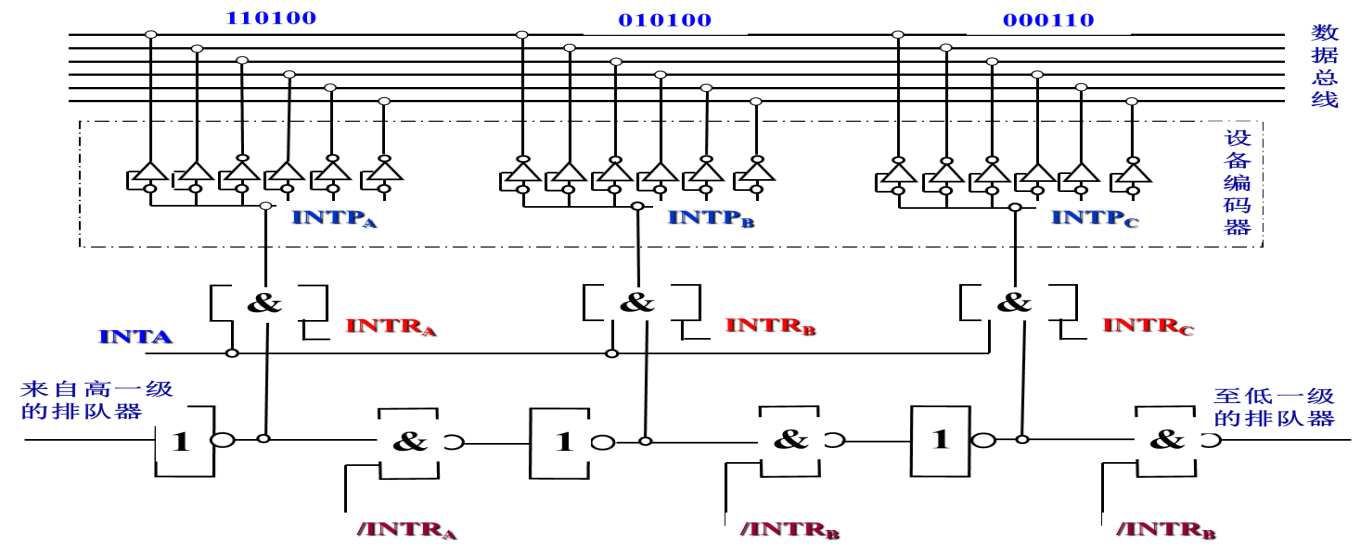
## 25 根据以下要求设计一个产生3个设备向量地址的电路。

(1) 3个设备的优先级A→B→C降序排列。

(2) A、B、C的向量地址分别为110100、010100、000110。

(3) 排队器采用链式排队电路。

(4) 当CPU发来中断响应信号INTA时，可将向量地址取至CPU。



## 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。

## 27. DMA方式有何特点？什么样的I/O设备与主机交换信息时采用DMA方式，举例说明。

由于主存和DMA接口之间有一条数据通路，因此主存和设备交换信息是，不通过CPU，也不需要CPU暂停现行程序为设备服务，省去了保护和恢复现场，因此工作速度比程序中断方式的高。

通常DMA与主存交换数据是采用如下三种方法：

（1）停止CPU访问主存

（2）周期挪用（周期窃取）

（3）DMA与CPU交替访问

当I/O设备与硬盘、U盘等辅存交换信息时，即可采用DMA方式。因为在此传输过程中不需要CPU对信息进行处理，故不需要CPU的参与，故可以采用DMA方式。

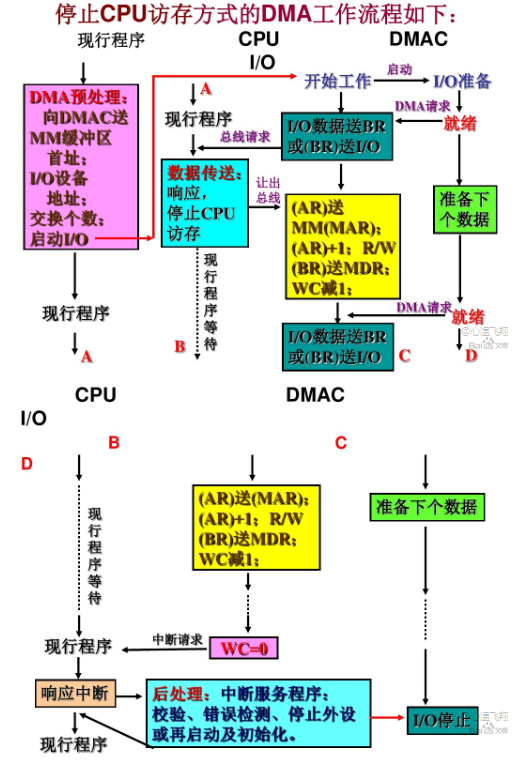
## 28. CPU对DMA请求和中断请求的响应时间是否一样？为什么？

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。

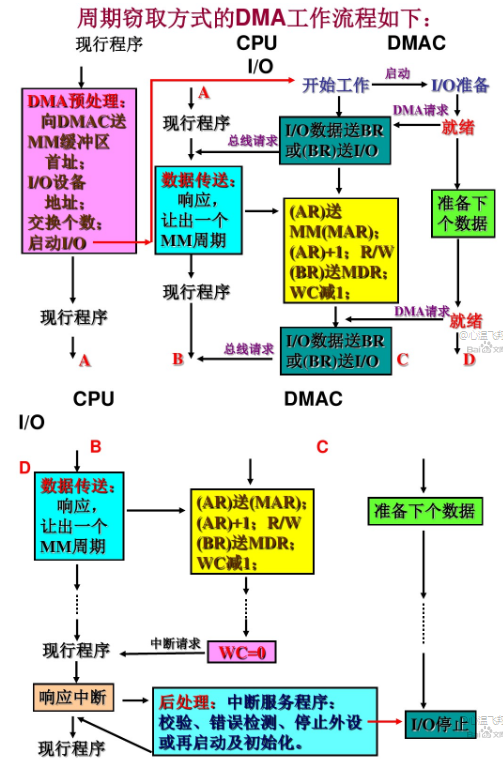
中断方式是程序切换，而程序又是由指令组成，所以必须在一条指令执行完毕才能响应中断请求，而且CPU只有在每条指令执行周期结束的时刻才发出查询信号，以获取中断请求信号，若此时条件满足，便能响应中断请求。

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。



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



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40μs，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=0.025×10-3=25 μ s < 40μs

    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25μs，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= 1K ×8 ×8 位= 8KB = 4K字

数传率=4K字×3000转/分=4K字×50转/秒 =200K字/秒

一个字的传送时间=1/200K秒≈ 5μs （注：在此1K=1024，来自数据块单位缩写。）

因为5 μs<<25μs，所以不能采用一条指令执行结束响应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=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；

讨论：

问题1：这里的传送速度指I/O设备与主存间，还是I/O与CPU之间？

答：视具体传送方式而定，程序查询、程序中断为I/O与CPU之间交换，DMA为I/O与主存间交换。

问题2：主动性应以CPU的操作方式看，而不是以I/O的操作方式看。

**补充题：**

一、某CRT显示器可显示64种ASCII字符，每帧可显示72字×24排；每个字符字形采用7×8点阵，即横向7点，字间间隔1点，纵向8点，排间间隔6点；帧频50Hz，采取逐行扫描方式。假设不考虑屏幕四边的失真问题，且行回扫和帧回扫均占扫描时间的20%，问：

1）显存容量至少有多大？

2）字符发生器（ROM）容量至少有多大？

3）显存中存放的是那种信息？

4）显存地址与屏幕显示位置如何对应？

5）设置哪些计数器以控制显存访问与屏幕扫描之间的同步？它们的模各是多少？

6）点时钟频率为多少？

解：1）显存最小容量=72×24×8 =1728B

2）ROM最小容量=64×8行×8列= 512B（含字间隔1点）

3）显存中存放的是ASCII码信息。

4）显存每个地址对应一个字符显示位置，显示位置自左至右，从上到下，分别对应缓存地址由低到高。

5）设置点计数器、字计数器、行计数器、排计数器控制显存访问与屏幕扫描之间的同步。

它们的模计算如下：

点计数器模 = 7+1 = 8

行计数器模 = 8 + 6 = 14

字、排计数器的模不仅与扫描正程时间有关，而且与扫描逆程时间有关，因此计算较为复杂。

列方程： （72+x）× 0.8 = 72

（24+y）× 0.8 = 24

解方程得：x = 18，y = 6，则：

字计数器模 = 72 + 18 = 90

排计数器模 = 24 + 6 = 30

1. 点频 = 50Hz × 30排 × 14行 × 90字 × 8点   = 15 120 000Hz= 15.12MHz

二、有一编码键盘，其键阵列为8行×16列，分别对应128种ASCII码字符，采用硬件扫描方式确认按键信号，问：

1）扫描计数器应为多少位？

2）ROM容量为多大？

3）若行、列号均从0开始编排，则当第5行第7列的键表示字母“F”时，CPU从键盘读入的二进制编码应为多少（设采用奇校验）？

4）参考教材图5.15，画出该键盘的原理性逻辑框图；

5）如果不考虑校验技术，此时ROM是否可省？

解：1）扫描计数器 = 7位（与键的个数有关）

2）ROM容量 = 128 × 8 = 128B（与字符集大小有关）

3）CPU从键盘读入的应为字符“F”的ASCII码 = 01000110 （46H） ，其中最高位为奇校验位（注：不是位置码）。

4）该键盘的原理性逻辑框图见下页，与教材图5.15类似，主要需标明参数。

5）如果不考虑校验技术，并按ASCII码位序设计键阵列（注意） ，则ROM编码表可省，此时7位计数器输出值（扫描码或键位置码）即为ASCII码。

该键盘的原理性逻辑框图如下：

三、一针式打印机采用7列×9行点阵打印字符，每行可打印132个字符，共有96种可打印字符，用带偶校验位的ASCII码表示。问：

1）打印缓存容量至少有多大？

2）字符发生器容量至少有多大？

3）列计数器应有多少位？

4）缓存地址计数器应有多少位？

解：1）打印缓存最小容量 = 132×8= 132B（考虑偶校验位）

2）ROM最小容量 = 96×7列×9行= 672×9位

3）列计数器 = 3位（7列向上取2的幂）

4）缓存地址计数器 = 8位（132向上取2的幂）

讨论：

1．由于针打是按列打印，所以ROM一个存储单元中存一列的9个点，则容量为672×9位；

2．列计数器是对列号进行计数，所以模=7，3位（模不等于位数）；

3．同样缓存地址计数器模=132，8位。

第6章 计算机的运算方法

2. 已知X=0.a1a2a3a4a5a6（ai为0或1），讨论下列几种情况时ai各取何值。

（1）

（2）

（3）

解： （1）若要，只要a1=1，a2~a6不全为0即可。

（2）若要，只要a1~a3不全为0即可。

（3）若要，只要a1=0，a2可任取0或1；

当a2=0时，若a3=0，则必须a4=1，且a5、a6不全为0；

若a3=1，则a4~a6可任取0或1；

当a2=1时， a3~a6均取0。

3. 设x为整数，[x]补=1，x1x2x3x4x5，若要求 x < -16，试问 x1~x5 应取何值？

解：若要x < -16，需 x1=0，x2~x5 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]补，求[x]原和x。

[x1]补=1.1100; [x2]补=1.1001; [x3]补=0.1110;  [x4]补=1.0000;

[x5]补=1,0101; [x6]补=1,1100; [x7]补=0,0111; [x8]补=1,0000;

解：[x]补与[x]原、x的对应关系如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [x]补 | 1.1100 | 1.1001 | 0.1110 | 1.0000 | 1,0101 | 1,1100 | 0,0111 | 1,0000 |
| [x]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 |

## 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]补=[x]原成立。

解：当x为小数时，若x≥ 0，则  [x]补=[x]原成立；

        若x < 0，当x= -1/2时，[x]补=[x]原=1.100 0000，则  [x]补=[x]原成立。

当x为整数时，若x≥0，则  [x]补=[x]原成立；

## 若x< 0，当x= -64时，[x]补=[x]原=1,100 0000，则 [x]补=[x]原成立。 设x为真值，x\*为绝对值，说明[-x\*]补=[-x]补能否成立。

解：当x为真值，x\*为绝对值时，[-x\*]补=[-x]补不能成立。原因如下：

（1）当x<0时，由于[-x\*]补是一个负值，而[-x]补是一个正值，因此此时[-x\*]补=[-x]补不成立；

（2）当x≥0时，由于-x\*=-x，因此此时 [-x\*]补=[-x]补的结论成立。

## 8. 讨论若[x]补>[y]补，是否有x>y？

解：若[x]补>[y]补，不一定有x>y。 [x]补 > [y]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]补<[y]补；同样，当x<0、 y >0时，有x < y，但[x]补>[y]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=23\*0.111011B

x4=-86.5=-1010110.1B=27\*(-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）基值不同对浮点数什么有影响？

    （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=215×(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=2-16×2-1=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=1615×（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=16-16×16-1=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（215）< 6万 <65536（216），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≥25

## 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 ~ 216 - 1，即：0~ 65535；

   无符号小数：0 ~ 1 - 2-16 ，即：0 ~ 0.99998；

（2）原码定点小数：-1 + 2-15~1 - 2-15 ，即：-0.99997~0.99997

（3）补码定点小数：- 1~1 - 2-15 ，即：-1~0.99997

（4）补码定点整数：-215~215 - 1 ，即：-32768~32767

（5）原码定点整数：-215 + 1~215 - 1，即：-32767~32767

（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 -2-9×2-31

最小负数= 0，11 111；1.111 111 111，即 -（1-2-9）×231

则负数表示范围为：-（1-2-9）×231 —— -2-9×2-31

最大正数= 0，11 111；0.111 111 111，即 （1-2-9）×231

最小正数= 1，11 111；0.000 000 001，即 2-9×2-31

则正数表示范围为：2-9×2-31 ——（1-2-9）×231

（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 -2-1×2-32

最小负数=0，11 111；1.000 000 000，即 -1×231

则负数表示范围为：-1×231 —— -2-1×2-32

最大正数=0，11 111；0.111 111 111，即 （1-2-9）×231

最小正数=1，00 000；0.100 000 000，即 2-1×2-32

则正数表示范围为：2-1×2-32 ——（1-2-9）×231

## 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]原=0.001 1010；[y1]补=0.101 0100；[z1]反=1.010 1111；

[x2]原=1.110 1000；[y2]补=1.110 1000；[z2]反=1.110 1000；

[x3]原=1.001 1001；[y3]补=1.001 1001；[z3]反=1.001 1001。

解：算术左移一位：

[x1]原=0.011 0100；正确

[x2]原=1.101 0000；溢出（丢1）出错

[x3]原=1.011 0010；正确

[y1]补=0.010 1000；溢出（丢1）出错

[y2]补=1.101 0000；正确

[y3]补=1.011 0010；溢出（丢0）出错

[z1]反=1.101 1111；溢出（丢0）出错

[z2]反=1.101 0001；正确

[z3]反=1.011 0011；溢出（丢0）出错

算术左移两位：

[x1]原=0.110 1000；正确

[x2]原=1.010 0000；溢出（丢11）出错

[x3]原=1.110 0100；正确

[y1]补=0.101 0000；溢出（丢10）出错

[y2]补=1.010 0000；正确

[y3]补=1.110 0100；溢出（丢00）出错

[z1]反=1.011 1111；溢出（丢01）出错

[z2]反=1.010 0011；正确

[z3]反=1.110 0111；溢出（丢00）出错

算术右移一位：

[x1]原=0.000 1101；正确

[x2]原=1.011 0100；正确

[x3]原=1.000 1100(1)；丢1，产生误差

[y1]补=0.010 1010；正确

[y2]补=1.111 0100；正确

[y3]补=1.100 1100(1)；丢1，产生误差

[z1]反=1.101 0111；正确

[z2]反=1.111 0100(0)；丢0，产生误差

[z3]反=1.100 1100；正确

算术右移两位：

[x1]原=0.000 0110（10）；产生误差

[x2]原=1.001 1010；正确

[x3]原=1.000 0110（01）；产生误差

[y1]补=0.001 0101；正确

[y2]补=1.111 1010；正确

[y3]补=1.110 0110（01）；产生误差

[z1]反=1.110 1011；正确

[z2]反=1.111 1010（00）；产生误差

[z3]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]补=0.001 0010, [B]补=1.100 1100

[A+B]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]补=0.100 1100, [B]补=1.110 1111 , [-B]补=0.001 0001

[A-B]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]补=1.110 1000, [B]补= 0.010 0100

[A+B]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]补=1 010 1001, [B]补=0 011 0101, [-B]补=1 100 1011

[A-B]补= 1 0101001 + 1 1001011 = 0 1110100 —— 溢出

（5）A=115= 111 0011B, B= -24= -11 000B

    [A]补=0 1110011, [B]补=1，110 1000

[A+B]补= 0 1110011 + 1 1101000 = 0 1011011——无溢出

A+B= 101 1011B = 91

20. 用原码一位乘、两位乘和补码一位乘（Booth算法）、两位乘计算x·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]原=0.110111，[y]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\*×y\*=0.100 111 100 010，z0=x0⊕ y0=0 ⊕1=1，

[x×y]原=1.100 111 100 010，x·y= -0. 100 111 100 010

**原码两位乘：**[-x\*]补=1.001 001，2x\*=1.101 110

|  |  |  |  |
| --- | --- | --- | --- |
| 部分积 | 乘数y\* | Cj | 说明 |
| 000 . 000 000  +001 . 101 110 | 00 101 110 | 0 | 部分积初值为0，Cj=0  根据yn-1ynCj=100，加2x\*，保持Cj=0 |
| 001 . 101 110 |  | 0 |  |
| 000 . 011 011  +111 . 001 001 | 10 001 011  10 001 011 | 0 | 右移2位  根据yn-1ynCj=110，加[-x\*]补，置Cj=1 |
| 111 . 100 100  111 . 111 001  +111 . 001 001 | 00 100 010 | 1 | 右移2位根据yn-1ynCj=101，加[-x\*]补，置Cj=1 |
| 111 . 000 010  111 . 110 000  +000 . 110 111 | 10 001 000 | 1 | 右移2位  根据yn-1ynCj=001，加x\*，保持Cj=0 |
| 000 . 100 111 | 10 001 0 |  |  |

即x\*×y\*=0.100 111 100 010，z0=x0⊕ y0=0 ⊕1=1，

[x×y]原=1.100 111 100 010，x·y= -0. 100 111 100 010

**补码一位乘：**[x]补=0.110111，[-x]补=1.001001，[y]补=1.010010

|  |  |  |  |
| --- | --- | --- | --- |
| 部分积 | 乘数 | Yn+1 | 说明 |
| 00 . 000 000  00 . 000 000  +11 . 001 001 | 1 010 010  0 101 001 | 0  0 | Ynyn+1=00，部分积右移1位  Ynyn+1=10，部分积加[-x]补 |
| 11 . 001 001 |  |  | 右移1位 |
| 11 . 100 100  +00 . 110 111 | 1 010 100 | 1 | Ynyn+1=01，部分积加[x]补 |
| 00 . 011 011 |  |  | 右移1位 |
| 00 . 001 101  00 . 000 110  +11 . 001 001 | 1 101 010  1 110 101 | 0  0 | Ynyn+1=00，部分积右移1位  Ynyn+1=10，部分积加[-x]补 |
| 11 . 001 111 |  |  | 右移1位 |
| 11 . 100 111  +00 . 110 111 | 1 111 010 | 1 | Ynyn+1=01，部分积加[x]补 |
| 00 . 011 110  00 . 001 111  +11 . 001 001 | 0 111 101 | 0 | 右移1位  Ynyn+1=10，部分积加[-x]补 |
| 11 . 011 000 | 0 111 10 |  |  |

即 [x×y]补=1.011 000 011 110，x·y= -0.100 111 100 010

**补码两位乘：**

**21. 用原码加减交替法和补码加减交替法计算x÷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]原=[x]补=x= 0.100 111  
 y\*=[y]原=[y]补=y= 0.101 011  
 [-y\*]补=[-y]补=1.010 101  
 q0=x0⊕y0=0 ⊕0=0  
 x÷y=x\*÷y\*=[x÷y]原=0.111 010  
 r\*=0.000 010×2-6=0.000 000 000 010  
 计算过程如下：**

**原码加减交替除法：  
 被除数（余数） 商  
 0 . 1 0 0 1 1 1 0 . 0 0 0 0 0 0  
 + 1 . 0 1 0 1 0 1 试减，+[-y\*]补  
 1 . 1 1 1 1 0 0  
1← 1 . 1 1 1 0 0 0 0 .  
 + 0 . 1 0 1 0 1 1 r<0，+y\*  
 0 . 1 0 0 0 1 1   
1← 1 . 0 0 0 1 1 0 0.1  
 + 1 . 0 1 0 1 0 1 r>0， +[-y\*]补  
 0 . 0 1 1 0 1 1  
1← 0 . 1 1 0 1 1 0 0.1 1  
 + 1 . 0 1 0 1 0 1 r>0， +[-y\*]补  
 0 . 0 0 1 0 1 1**

**被除数（余数） 商  
1← 0 . 0 1 0 1 1 0 0 . 1 1 1  
 + 1 . 0 1 0 1 0 1 r>0， +[-y\*]补  
 1 . 1 0 1 0 1 1  
1← 1 . 0 1 0 1 1 0 0.1 1 1 0  
 + 0 . 1 0 1 0 1 1 r<0，+y\*  
 0 . 0 0 0 0 0 1  
1← 0 . 0 0 0 0 1 0 0.1 1 1 0 1  
 + 1 . 0 1 0 1 0 1 r>0， +[-y\*]补  
 1 . 0 1 0 1 1 1 1← 0.1 1 1 0 1 0   
 + 0 . 1 0 1 0 1 1 r<0，+y\*（恢复余数）  
 0 . 0 0 0 0 1 0**

**补码加减交替除法：  
 被除数（余数） 商  
 0 0 . 1 0 0 1 1 1 0 . 0 0 0 0 0 0  
 + 1 1 . 0 1 0 1 0 1 试减，x、y同号，+[-y]补  
 1 1 . 1 1 1 1 0 0  
1← 1 1 . 1 1 1 0 0 0 0 .  
 + 0 0 . 1 0 1 0 1 1 r、y异号，+[y]补  
 0 0 . 1 0 0 0 1 1   
1← 0 1 . 0 0 0 1 1 0 0.1  
 + 1 1 . 0 1 0 1 0 1 r、y同号， +[-y]补  
 0 0 . 0 1 1 0 1 1  
1← 0 0 . 1 1 0 1 1 0 0.1 1  
 + 1 1 . 0 1 0 1 0 1 r、y同号， +[-y]补  
 0 0 . 0 0 1 0 1 1**

**续： 被除数（余数） 商  
1← 0 0 . 0 1 0 1 1 0 0 . 1 1 1  
 + 1 1 . 0 1 0 1 0 1 r、y同号， +[-y]补  
 1 1 . 1 0 1 0 1 1  
1← 1 1 . 0 1 0 1 1 0 0.1 1 1 0  
 + 0 0 . 1 0 1 0 1 1 r、y异号，+[y]补  
 0 0 . 0 0 0 0 0 1  
1← 0 0 . 0 0 0 0 1 0 0.1 1 1 0 1  
 + 1 1 . 0 1 0 1 0 1 r、y同号， +[-y]补  
 1 1 . 0 1 0 1 1 1 1← 0.1 1 1 0 1 1 —— 恒置1  
 + 0 0 . 1 0 1 0 1 1 r、x异号，（恢复余数）  
 0 0 . 0 0 0 0 1 0 且r、y异号， +[y]补  
注：恒置1引入误差。 x÷y=[x÷y]补= 0.111 011   
 [r]补=0.000 010，r=r\*=0.000 000 000 010**

**（2）x= -0.101 01，y=0.110 11  
 [x]原=1.101 01  
 x\*= 0.101 01  
 y\* = [y]原 = [y]补= y = 0.110 11   
 [-y\*]补= [-y]补= 1.001 01  
 [x]补= 1.010 11  
 q0 = x0 ⊕ y0 = 1 ⊕ 0 = 1  
 x\*÷y\*= 0.110 00  
 [x÷y]原=1.110 00   
 x÷y = -0.110 00  
 r\*=0.110 00×2-5  
 =0.000 001 100 0  
 计算过程如下：**

**原码加减交替除法：  
 被除数（余数） 商  
 0 . 1 0 1 0 1 0 . 0 0 0 0 0  
 + 1 . 0 0 1 0 1 试减，+[-y\*]补  
 1 . 1 1 0 1 0  
1← 1 . 1 0 1 0 0 0 .  
 + 0 . 1 1 0 1 1 r<0，+y\*  
 0 . 0 1 1 1 1   
1← 0 . 1 1 1 1 0 0.1  
 + 1 . 0 0 1 0 1 r>0， +[-y\*]补  
 0 . 0 0 0 1 1  
1← 0 . 0 0 1 1 0 0.1 1  
 + 1 . 0 0 1 0 1 r>0， +[-y\*]补  
 1 . 0 1 0 1 1**

**续：   
 被除数（余数） 商  
1← 0 . 1 0 1 1 0 0 . 1 1 0  
 + 0 . 1 1 0 1 1 r<0， +y\*  
 1 . 1 0 0 0 1  
1← 1 . 0 0 0 1 0 0.1 1 0 0  
 + 0 . 1 1 0 1 1 r<0，+y\*  
 1 . 1 1 1 0 1 1← 0.1 1 0 0 0  
 + 0 . 1 1 0 1 1 r<0， +y\*（恢复余数）  
 0 . 1 1 0 0 0**

**补码加减交替除法：  
 被除数（余数） 商  
 1 1 . 0 1 0 1 1 0 . 0 0 0 0 0  
 + 0 0 . 1 1 0 1 1 试减，x、y异号，+[y]补  
 0 0 . 0 0 1 1 0  
1← 0 0 . 0 1 1 0 0 1 .  
 + 1 1 . 0 0 1 0 1 r、y同号，+[-y]补  
 1 1 . 1 0 0 0 1   
1← 1 1 . 0 0 0 1 0 1.0  
 + 0 0 . 1 1 0 1 1 r、y异号， +[y]补  
 1 1 . 1 1 1 0 1  
1← 1 1 . 1 1 0 1 0 1.0 0  
 + 0 0 . 1 1 0 1 1 r、y异号， +[y]补  
 0 0 . 1 0 1 0 1**

**被除数（余数） 商  
1← 0 1 . 0 1 0 1 0 1 . 0 0 1  
 + 1 1 . 0 0 1 0 1 r、y同号， +[-y]补  
 0 0 . 0 1 1 1 1  
1← 0 0 . 1 1 1 1 0 1.0 0 1 1  
 + 1 1 . 0 0 1 0 1 r、y同号，+[-y]补  
 0 0 . 0 0 0 1 1 1← 1.0 0 1 1 1 —— 恒置1   
 + 1 1 . 0 0 1 0 1 r、x异号，（恢复余数）   
 1 1 . 0 1 0 0 0 且r、y同号，+[-y]补  
 注：恒置1引入误差。   
 [r]补=1.010 00， r= -0.000 001 100 0   
 [x÷y]补=1.001 11，x÷y= -0.110 01**

**（3）x= 0.101 00，y= -0.100 01  
 x\*= [x]原= [x]补= x=0.101 00  
 [y]原 = 1.100 01  
 y\* = 0.100 01   
 [-y\*]补=1.011 11  
 [y]补= 1.011 11  
 [-y]补= 0.100 01   
 q0 = x0 ⊕ y0 = 0 ⊕ 1 = 1  
 x\*÷y\*= 1.001 01 —— 溢出  
 [x÷y]原：无定义   
 x÷y = -1.001 01  
 r\*=0.010 11×2-5  
 =0.000 000 101 1  
 计算过程如下：**

**被除数（余数） 商  
 0 . 1 0 1 0 0 0 . 0 0 0 0 0  
 + 1 . 0 1 1 1 1 试减，+[-y\*]补  
 0 . 0 0 0 1 1   
1← 0 . 0 0 1 1 0 1 .  
 + 1 . 0 1 1 1 1 r>0， +[-y\*]补   
 1 . 1 0 1 0 1   
1← 1 . 0 1 0 1 0 1.0  
 + 0 . 1 0 0 0 1 r<0， +y\*  
 1 . 1 1 0 1 1  
1← 1 . 1 0 1 1 0 1.0 0  
 + 0 . 1 0 0 0 1 r<0， +y\*  
 0 . 0 0 1 1 1**

**被除数（余数） 商  
1← 0 . 0 1 1 1 0 1 . 0 0 1  
 + 1 . 0 1 1 1 1 r>0， +[-y\*]补   
 1 . 1 1 1 0 1  
1← 1 . 1 1 0 1 0 1.0 0 1 0  
 + 0 . 1 0 0 0 1 r<0，+y\*  
 0 . 0 1 0 1 1 1← 1.0 0 1 0 1  
 r>0， 结束  
 注：当x\*>y\*时产生溢出，这种情况在第一步运算后判断r的正负时就可发现。此时数值位占领小数点左边的1位，原码无定义，但算法本身仍可正常运行。**

**补码加减交替除法：  
 被除数（余数） 商  
 0 0 . 1 0 1 0 0 0 . 0 0 0 0 0  
 + 1 1 . 0 1 1 1 1 试减，x、y异号，+[y]补  
 0 0 . 0 0 0 1 1  
1← 0 0 . 0 0 1 1 0 0 .  
 + 1 1 . 0 1 1 1 1 r、y异号，+[y]补  
 1 1 . 1 0 1 0 1   
1← 1 1 . 0 1 0 1 0 0.1  
 + 0 0 . 1 0 0 0 1 r、y同号， +[-y]补  
 1 1 . 1 1 0 1 1  
1← 1 1 . 1 0 1 1 0 0.1 1  
 + 0 0 . 1 0 0 0 1 r、y同号， +[-y]补  
 0 0 . 0 0 1 1 1**

**被除数（余数） 商  
1← 0 0 . 0 1 1 1 0 0 . 1 1 0  
 + 1 1 . 0 1 1 1 1 r、y异号， +[y]补  
 1 1 . 1 1 1 0 1  
1← 1 1 . 1 1 0 1 0 0.1 1 0 1  
 + 0 0 . 1 0 0 0 1 r、y同号，+[-y]补  
 0 0 . 0 1 0 1 1 1← 0.1 1 0 1 1 —— 恒置1   
 r、x同号，结束   
 [r]补=0.010 11，r=r\*=0.000 000 101 1  
 真符位的产生：qf = x0 ⊕ y0 = 0 ⊕ 1 = 1   
 [x÷y]补=10.110 11，x÷y= -1.001 01  
 判溢出：qf ⊕ q0 = 1 ⊕ 0 = 1，溢出**

**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）x=13/32=（0.011 01）2  
 y= -27/32=（-0.110 11）2   
 x\*= [x]原= [x]补= x=0. 011 01  
 [y]原 = 1.110 11  
 y\* = 0.110 11   
 [-y\*]补=1.001 01  
 [y]补= 1.001 01  
 [-y]补= 0.110 11   
 q0 = x0 ⊕ y0 = 0 ⊕ 1 = 1  
 x\*÷y\*= 0.011 11   
 [x÷y]原=1.011 11   
 x÷y =（-0.011 11）2 = -15/32  
 r\*=0.010 11×2-5  
 =0.000 000 101 1**

**原码加减交替除法：  
 被除数（余数） 商  
 0 . 0 1 1 0 1 0 . 0 0 0 0 0  
 + 1 . 0 0 1 0 1 试减，+[-y\*]补  
 1 . 1 0 0 1 0   
1← 1 . 0 0 1 0 0 0 .  
 + 0 . 1 1 0 1 1 r<0， +y\*   
 1 . 1 1 1 1 1   
1← 1 . 1 1 1 1 0 0.0  
 + 0 . 1 1 0 1 1 r<0， +y\*  
 0 . 1 1 0 0 1  
1← 1 . 1 0 0 1 0 0.0 1  
 + 1 . 0 0 1 0 1 r>0， +[-y\*]补  
 0 . 1 0 1 1 1**

**被除数（余数） 商  
1← 1 . 0 1 1 1 0 0 . 0 1 1  
 + 1 . 0 0 1 0 1 r>0， +[-y\*]补   
 0 . 1 0 0 1 1  
1← 1 . 0 0 1 1 0 0.0 1 1 1  
 + 1 . 0 0 1 0 1 r>0， +[-y\*]补   
 0 . 0 1 0 1 1 1← 0.0 1 1 1 1  
 r>0， 结束**

**补码加减交替除法：  
 被除数（余数） 商  
 0 0 . 0 1 1 0 1 0 . 0 0 0 0 0  
 + 1 1 . 0 0 1 0 1 试减，x、y异号，+[y]补  
 1 1 . 1 0 0 1 0  
1← 1 1 . 0 0 1 0 0 1 .  
 + 0 0 . 1 1 0 1 1 r、y同号，+[-y]补  
 1 1 . 1 1 1 1 1   
1← 1 1 . 1 1 1 1 0 1.1  
 + 0 0 . 1 1 0 1 1 r、y同号，+[-y]补  
 0 0 . 1 1 0 0 1  
1← 0 1 . 1 0 0 1 0 1.1 0  
 + 1 1 . 0 0 1 0 1 r、y异号， +[y]补  
 0 0 . 1 0 1 1 1**

**被除数（余数） 商  
1← 0 1 . 0 1 1 1 0 1 . 1 0 0  
 + 1 1 . 0 0 1 0 1 r、y异号， +[y]补  
 0 0 . 1 0 0 1 1  
1← 0 1 . 0 0 1 1 0 1.1 0 0 0  
 + 1 1 . 0 0 1 0 1 r、y异号，+[y]补  
 0 0 . 0 1 0 1 1 1← 1.1 0 0 0 1 —— 恒置1   
 r、x同号，结束   
 [r]补=0.010 11，r=r\*=0.000 000 101 1  
 [x÷y]补=1.100 01，x÷y=（-0.011 11）2 = -15/32**

https://wenku.baidu.com/view/73881d4a79563c1ec5da7171.html?fr=search-income1&fixfr=wNv%2BikrJq0waCXDEVyg%2BJA%3D%3D

## 26.按机器补码浮点运算步骤，计算[x±y]补.

  （1）x=2-011× 0.101 100，y=2-010×（-0.011 100）；

  （2）x=2-011×（-0.100 010），y=2-010×（-0.011 111）；

  （3）x=2101×（-0.100 101），y=2100×（-0.001 111)。

解：先将x、y转换成机器数形式：

**（1）**x=2-011× 0.101 100，y=2-010×（-0.011 100）

[x]补=1，101；0.101 100, [y]补=1，110；1.100 100

[Ex]补=1,101, [y]补=1,110, [Mx]补=0.101 100, [My]补=1.100 100

    1）对阶：

[ΔE]补=[Ex]补+[-Ey]补 = 11,101+ 00,010=11,111 < 0，

应Ex向Ey对齐，则：[Ex]补+1=11，101+00，001=11，110 = [Ey]补

[x]补=1，110；0.010 110

2）尾数运算：

  [Mx]补+[My]补= 0.010 110 + 11.100 100=11.111010

[Mx]补+[-My]补=0.010 110 + 00.011100= 00.110 010

3）结果规格化：

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

  [x-y]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×（-0.110 000）

    x-y =2-010×0.110 010

**（2）**x=2-011×（-0.100010）,y=2-010×（-0.011111）

    [x]补=1，101；1.011 110, [y]补=1，110；1.100 001

1. 对阶：过程同(1)的1），则

[x]补=1，110；1.101 111

  2）尾数运算：

    [Mx]补+[My]补= 11.101111 + 11. 100001 = 11.010000

    [Mx]补+[-My]补= 11.101111 + 00.011111 = 00.001110

3）结果规格化：

[x+y]补=11，110；11.010 000,已是规格化数

[x-y]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×（-0.110 000）

    x-y =2-100×0.111 000

**（3）**x=2101×（-0.100 101）,y=2100×（-0.001 111）

    [x]补=0，101；1.011 011, [y]补=0，100；1.110 001

1）对阶：

[ΔE]补=00，101+11，100=00，001 >0，应Ey向Ex对齐，则：

[Ey]补+1=00，100+00，001=00，101=[Ex]补

[y]补=0，101；1.111 000（1）

2）尾数运算：

  [Mx]补+[My]补= 11.011011+ 11.111000（1）= 11.010011（1）

  [Mx]补+[-My]补= 11.011011+ 00.000111（1）= 11.100010（1）

3）结果规格化：

  [x+y]补=00，101；11.010 011（1），已是规格化数

  [x-y]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]补=00，101；11.010 011（舍）

[x-y]补 不变

5）溢出：无

则：x+y=2101×（-0.101 101）

x-y =2100×（-0.111 011）

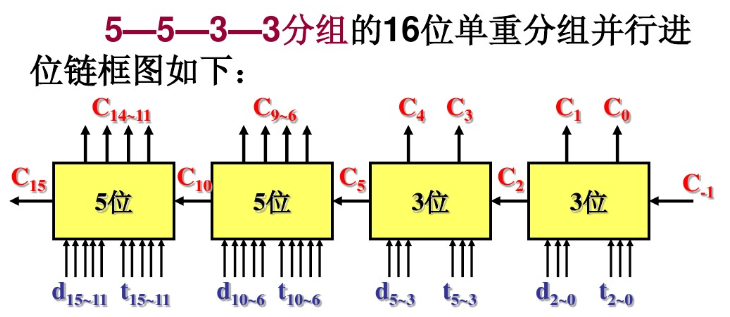
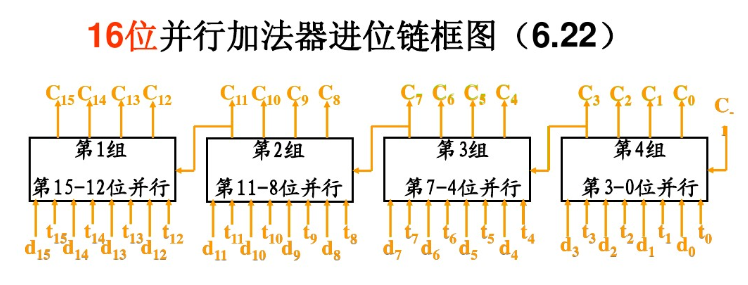
32. 设机器字长为16位，分别按4、4、4、4和5、5、3、3分组后，

  （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）



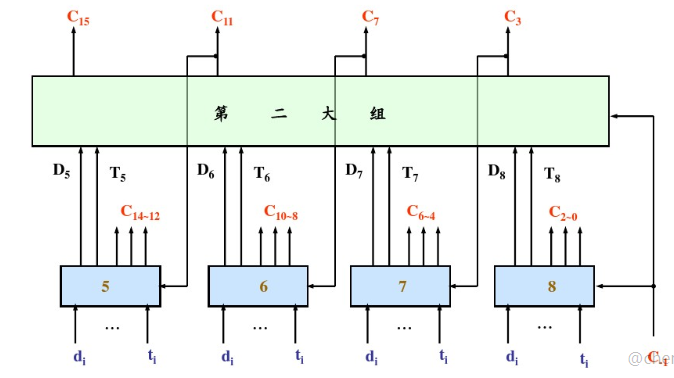
4—4—4—4分组的进位时间=2.5ty\*4=10ty;

5—5—3—3分组的进位时间=2.5ty\*4=10ty;

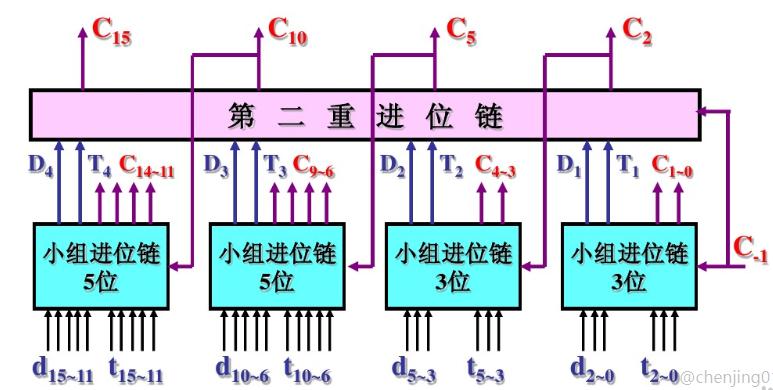
两种分组方案最长加法时间相同。

结论:单重分组并行进位的最长进位时间只与组数有关，与组内位数无关。

（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：



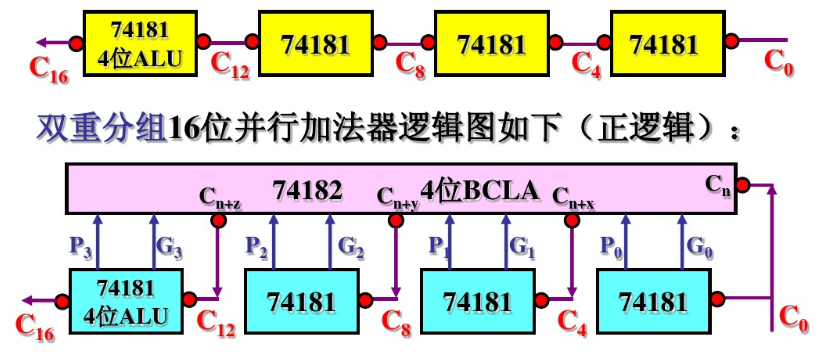
5—5—3—3分组的进位时间=2.5ty×3=7.5ty；

4—4—4—4分组的进位时间=2.5ty×3=7.5ty；

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

## 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与硬件结构之间存在着密切的关系？

机器指令: 是CPU能直接识别并执行的指令，它的表现形式是二进制编码。机器指令通常由操作码和操作数两部分组成。

指令系统: 计算机所能执行的全部指令的集合，它描述了计算机内全部的控制信息和“逻辑判断”能力。

指令系统是计算机硬件和软件的接口部分，是全部机器指令的集合。

## 答：参考P300。什么叫寻址方式？为什么要学习寻址方式？

寻址方式: 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，它与硬件结构紧密相关，而且直接影响指令格式和指令功能。

学习寻址方式，是为了找到指令中参与操作的数据，然后根据指令，得出结果。

## 答：参看P310。什么是指令字长、机器字长和存储字长？

指令字长: 是指机器指令中二进制代码的总位数。指令字长取决于从操作码的长度、操作数地址的长度和操作数地址的个数。不同的指令的字长是不同的。

机器字长: 是指计算机进行一次整数运算所能处理的二进制数据的位数（整数运算即定点整数运算)。机器字长也就是运算器进行定点数运算的字长，通常也是CPU内部数据通路的宽度。即字长越长，数的表示范围也越大，精度也越高。机器的字长也会影响机器的运算速度。

存储字长: 一个存储单元存储一串二进制代码（存储字)，这串二进制代码的位数称为存储字长，存储字长可以是8位、16位、32位等。

## 答：略。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。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=24-M-N

当M=1（最小值），N=1（最小值）时，二地址指令最多有：Kmax=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= 24 -（N/26 + M/212 ）；

当（N/26 + M/212 ）≤1时（N/26 + M/212 向上取整），K最大，则二地址指令最多有：

Kmax=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： 寄存器间接寻址：  
1.指令中的地址码是寄存器编号  
2.寄存器中存储的是真实地址(EA)  
 间接寻址：  
1。指令中地址码是某一主存地址（不妨设为：A）  
2。A中是真实地址(EA)  
另外，寄存器间接寻址和间接寻址均包括多次间址，即：需要寻址多次才能找到真实地址(EA)，如三次间址（注：多次间址往往需要在寄存器或主存存储单元中留出低一位表明是第几次间址）

**10. 试比较基址寻址和变址寻址。**

解:

1）都可以有效的扩大指令寻址范围。

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→(IX)+1

2）先间址再变址寻址过程简单示意如下：EA=(IX)+(A)，IX→ (IX)+116. 某机主存容量为4M×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26=64。

（3）由于存储字长为16位，故一次间址的寻址范围为216；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215。

（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：232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。

解:CPU实质包括运算器和控制器两大部分.

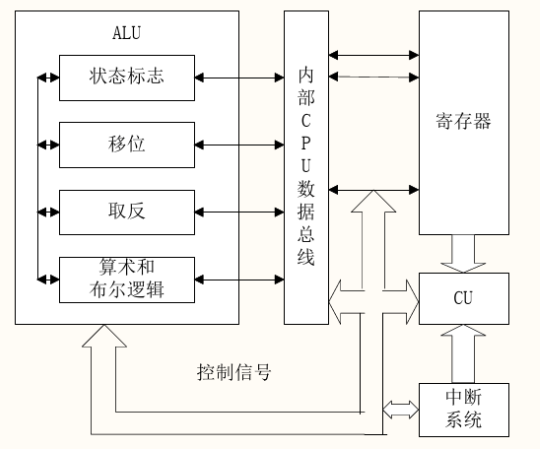
CPU必须具有控制程序的顺序执行(称指令控制)

产生完成每条指令所需的控制命令(称操作控制)

对各种操作加以时间上的控制(称时间控制)

对数据进行算术运算和逻辑运算(数据加工)

以及处理中断等功能。



寄存器：专用于存放当前指令的地址;

CU：能发出各种操作命令序列的控制部件;

ALU：存放操作数的寄存器和实现算逻运算的部件，完成算术运算和逻辑运算;ALU部件实际上只对CPU内部寄存器的数据进行操作

中断系统：处理异常情况和特殊请求。

## 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：



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20µs，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20μ 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

## 1. 设CPU内有这些部件：PC、IR、MAR、MDR、AC、CU。

（1）写出取指周期的全部微操作。

（2）写出减法指令“SUB X”、取数指令“STA X”（X均为主存地址）在执行阶段所需的全部微操作。

（3）当上述指令为间接寻址时，写出执行这些指令所需的全部微操作。

（4）写出无条件转移指令“JMP Y”和结果溢出则转指令“BAO Y”在执行阶段所需的全部微操作。

（1）PC->MAR; 1->R; M(MAR)->MDR; MDR->IR;

OP(IR)->CU; (PC)+1->PC;

（2）SUB X : AD(IR)->MAR; 1->R; M(MAR)->MDR;

(AC)-(MDR)->AC;

STA X : AD(IR)->MAR; 1->W; AC->MDR;

MDR->M(MAR)

（3）间接寻址时，比直接寻址多一个间址周期：

AD(IR)->MAR; 1->R; M(MAR)->MDR;

MDR->AD(IR);

（4）JMP Y : AD(IR)->PC

BAO Y: O·AD(IR)+O·(PC)->PC

## 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机平均指令周期=1/0.4MIPS=2.5µs

A机时钟周期=1/8MHz=125ns

A机机器周期=125ns×4=500ns=0.5µs

A机每个指令周期中含机器周期个数=2.5µs÷0.5µs=5个

B机时钟周期 =1/12MHz≈83ns

B机机器周期 =83ns×4=332ns

设B机每个指令周期也含5个机器周期，则：

B机平均指令周期=332ns×5=1.66µs

B机平均指令执行速度=1/1.66µs=0.6MIPS

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：

时钟周期=1/8MHz=0.125×10-6s

机器周期=0.125×10-6s×2=0.25×10-6s

平均指令周期=0.25×10-6s×4=10-6s

平均指令执行速度=1/10-6s=1MIPS

当参数改变后：机器周期= 0.125×10-6s×4=0.5×10-6s

平均指令周期=0.5×10-6s×4=2×10-6s

平均指令执行速度=1/（2×10-6s） =0.5MIPS

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=1/10MHz=0.1×10-6s

机器周期=0.1×10-6s×4=0.4×10-6s

  平均指令周期=1/1MIPS=10-6s

  每个指令周期所含机器周期个数=10-6s /0.4×10-6s =2.5个

   当芯片改变后：机器周期=0.4µs×4=1.6µs

平均指令周期=1.6µs×2.5=4µs

  平均指令执行速度=1/4µs=0.25MIPS

  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=1/0.8MIPS=1.25×10-6=1.25µs

机器周期=1.25µs÷2.5=0.5µs

时钟周期= 0.5µs÷4=0.125µs

CPU主频=1/0.125µs=8MHz8. 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 =（0.6×0.35+0.8×0.45+10×0.05+1.4×0.15）μs= 1.28μs

      机器平均运行速度 = 1/1.28μs≈0.78MIPS

(2)时钟周期 = 1/6MHz≈0.167μs

指令平均运行周期数 = 1.28μs ÷0.167μs≈7.66CPI

若CPU芯片升级为10MHz，时钟周期 = 1/10MHz=0.1μs

指令平均运行时间 = 0.1μs×7.66=0.766μs

机器平均运行速度 = 1/0.766μs≈1.3MIPS10.

## 9.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Bi为寄存器B的输入控制；Bo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) →B)

（2）SUB A,H ((AC)-(H) →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·→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\*48=29\*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<=23

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。