第一章

1.1　解释下列名词

摩尔定律：对集成电路上可容纳的晶体管数目、性能和价格等发展趋势的预测，其主要内容是：成集电路上可容纳的晶体管数量每 18 个月翻一番，性能将提高一倍，而其价格将降低一半。

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。

控制器：计算机的指挥中心，它使计算机各部件自动协调地工作。

时钟周期：时钟周期是时钟频率的倒数，也称为节拍周期或 T 周期，是处理操作最基本的时间单位。

多核处理器：多核处理器是指在一枚处理器中集成两个或多个完整的计算引擎(内核)。

字长：运算器一次运算处理的二进制位数。

存储容量：存储器中可存二进制信息的总量。

CPI：指执行每条指令所需要的平均时钟周期数。

MIPS：用每秒钟执行完成的指令数量作为衡量计算机性能的一个指标，该指标以每秒钟完成的百万指令数作为单位。

CPU 时间：计算某个任务时 CPU 实际消耗的时间，也即 CPU 真正花费在某程序上的时间。

计算机系统的层次结构：计算机系统的层次结构由多级构成，一般分成 5 级，由低到高分别是：微程序设计级，机器语言级，操作系统级，汇编语言级，高级语言级。

基准测试程序：把应用程序中使用频度最高的那那些核心程序作为评价计算机性能的标准程序。

软/硬件功能的等价性：从逻辑功能的角度来看，硬件和软件在完成某项功能上是相同的，称为软/硬件功能是等价的，如浮点运算既可以由软件实现，也可以由专门的硬件实现。

固件：是一种软件的固化，其目的是为了加快软件的执行速度。

可靠性：可靠性是指系统或产品在规定的条件和规定的时间内，完成规定功能的能力。产品可靠性定义的要素是三个“规定”：“规定条件”、“规定时间”和“规定功能”。

MTTF：平均无故障时间，指系统自使用以来到第一次出故障的时间间隔的期望值。

MTTR：系统的平均修复时间。

MTBF：平均故障间隔时间，指相邻两次故障之间的平均工作时间。

可用性：指系统在任意时刻可使用的概率，可根据 MTTF、MTTR 和 MTBF 等指标计算处系统的可用性。

1.3　冯·诺依曼型计算机的基本思想是什么？按此思想设计的计算机硬件系统应由哪些部件组成？各起什么作用？

答：冯诺依曼型计算机的基本思想是存储程序和程序控制，其中的“存储程序”是指将解题的步骤编写成程序，然后把存储存放到计算机的内存中，而“程序控制”是指控制器读出存放在存储器中的程序并根据该程序控制全机协调工作以完成程序的功能。

根据冯诺依曼型计算机的基本思想，计算机的硬件应该由运算器、控制器、存储器、输入/输出设备和总线组成。

各部件的作用：

运算器：对数据进行运算的部件。

存储器：存放程序和数据。

控制器：根据指令的功能控制构成计算机的各大功能部件协调工作，共同完成指令的功

能。

输入设备：将外部信息输送到主机内部的设备。

输出设备：能将计算机内部的信息以不同并且相应的形式反馈给人们的设备。

总线：连接两个或多个设备（部件）的公共信息通路。

1.4　什么是计算机字长？它取决于什么？计算机字长统一了哪些部件的长度？

答：计算机的字长一般指一次参与运算数据的基本长度，用二进制数位的长度来衡量。

它取决于运算器一次运算处理的二进制位数。它是计算机的重要性能指标。常用的计算机字长有 8 位、16 位、32 位及 64 位。

一般与计算机内部寄存器、加法器、数据总线的位数以及存储器字长等长，因此，字长直接影响硬件的代价。

1.5　计算机系统从功能上可划分为哪些层次？各层次在计算机系统中起什么作用？

答：计算机系统分成五级层次结构，第 1 级为微程序设计级、第 2 级为机器语言级、第 3 级为操作系统级、第 4 级为汇编语言级、第 5 级为高级语言级。

各层次的作用：

微程序级：为机器指令级提供机器指令的解释指行功能。

机器指令级：是软件系统和硬件系统的界面，一条机器指令的功能由微程序机器级的一段微型程序的功能实现。

操作系统级：调度计算机中的软件和硬件资源。

汇编语言级：它将用户编写的接近人类语言的程序，翻译成能在机器上运行的目标程序。

高级语言级：完全面向用户，是用户关心的目标，可执行各种用途的程序。

1.6　计算机内部有哪两股信息在流动？它们彼此有什么关系？

答：计算机中有两股信息在流动：一股是控制信息，即操作命令，它分散流向各个部件；一股是数据信息，它受控制信息的控制，从一个部件流向另一个部件，在流动的过程被相应的部件加工处理。

1.9 说明高级语言、汇编语言和机器语言三者之间的差别和联系。

答：机器语言是直接用二进制代码指令表达的计算机语言，是一种面向机器的编程语言，属于低级语言。

汇编语言是用助记符号来表示计算机指令的语言，也是低级的语言。

高级语言是一类接近于人类自然语言和数学语言的程序设计语言的统称，分为面向过程的语言和面向对象的语言。

它们都是计算机的编程语言，并且是计算机编程语言发展的三个阶段。三者各自的特点：

使用机器语言编写的程序，占用内存少、执行效率高。缺点：编程工作量大，容易出错；依赖具体的计算机体系，因而程序的通用性、移植性都很差。

使用汇编语言编写计算机程序，能够根据特定的应用对代码做最佳的优化，提高运行速度；能够最大限度地发挥硬件的功能。但是编写的代码非常难懂，不好维护；开发效率很低，时间长且单调。

高级语言的优点是：编程相对简单、直观、易理解、不容易出错；编写的计算机程序通用性好，具有较好的移植性。

1.10 什么是系统的可靠性?衡量系统可靠性的指标有哪些?如何提高系统的可靠性?

答：系统的可靠性是指系统在规定的条件和规定的时间内，完成规定功能的能力。

衡量系统可靠性的指标有三个：平均无故障时间、平均故障间隔时间和可用性。

提高系统可靠性的常用方法包括避错和容错。前者即避免错误的出现，从而提高系统的平均无故障时间；后者容许错误的出现，但采取有效的方法来防止其造成的不利影响。

1.11 假定某计算机 1 和计算机 2 以不同的方式实现了相同的指令集,该指令集中共有 A、B、C、D 四类指令，它们在程序中所占比例分别为 40%、20%、20%、20%，机器 1 和机器 2 的时钟周期为 600MHZ 和 800MHZ，各类指令在两机器上的 CPI 如表 1.5 所示，求两机器的 MIPS 各为多少？

表 1.5 两台计算机不同指令的 CPI

	A	B	C	D
CPI1	2	3	4	5
CPI2	2	2	3	4

解：CPI1= 2*0.4+ 0.2*(3+4+5)= 3.2　MIPS1= f/(CPI1×10^6) = 600×10^6/(3.2×10^6)=187.5

CPI2= 2*0.4+ 0.2*(2+3+4)= 2.6　MIPS2= f/(CPI1×10^6) = 800×10^6/(2.6×10^6)=307.7

1.12 若某程序编译后生成的目标代码由 A、B、C、D 四类指令组成，它们在程序中所占比例分别为 40%、20%、15%、25%。已知 A、B、C、D 四类指令的 CPI 分别为 1、2、2、2。现需要对程序进行编译优化，优化后的程序中 A 类指令条数减少了一半，而其它指令数量未发生变化。假设运行该程序的计算机 CPU 主频为 500MHZ。完成下列各题：

1)优化前后程序的 CPI 各为多少?

2)优化前后程序的 MIPS 各为多少?

3）通过上面的计算结果你能得出什么结论？

解：1)优化前：CPI= $\sum_{i=1}^{n}(CPI_i \times \frac{IC_i}{IC})$ = 1× 0.4 + 2× 0.2 + 2× 0.15 + 2× 0.25

= 1.6

优化后：A、B、C、D 四类指令在程序中所占比例分别为 1/4、1/4、3/16、5/16，

CPI= $\sum_{i=1}^{n}(CPI_i \times \frac{IC_i}{IC})$ = 1× 1/4 + 2× 1/4 + 2× 3/16 + 2× 5/16

= 1.75

2）根据 公式 MIPS = $\frac{\text{时钟频率}}{\text{CPI} \times 10^6}$ 得

优化前：MIPS = (500×10^6)/(1.6×10^6) = 312.5

优化后：MIPS = (500×10^6)/(1.75×10^6) = 285.7

3）优化后，A 类指令条数减少，造成计算机的 CPI 增加，MIPS 减少。这样的优化虽然减少了 A 类指令条数，却降低了程序的执行速度。

第二章

2.1 解释下列名词

真值：正号和负号分别用“+”和“-”表示，数据位保持二进制值不变的数据表示方法。

数值数据：计算机所支持的一种数据类型，用于科学计算，常见的数值数据类型包括小数、整数、浮点数数等。

非数值数据：计算机所支持的一种数据类型，一般用来表示符号或文字等没有数值值的数据。

机器数：数据在机器中的表示形式，是正负符号数码化后的二进制数据。

变形补码：用两个二进制位来表示数字的符号位，其余与补码相同。即“00”表示正，“11”表示负。

规格化：将非规格化的数处理成规格化数的过程。规格化数规定尾数用纯小数表示，且真值表示时小数点后第一位不为 0（以机器数表示时对小数点后第一位的规定与具体的机器数的形式有关）。

机器零：计算机保存数字的位有限，所能表示最小的数也有范围，其中有一个范围之中的数据无法精确表示，当实际的数据处在这个无法精确表示的数据范围时计算机就将该数作为机器零来处理，因此，计算机中的机器零其实对应的不是一个固定的数，而是一个数据表示范围。

BCD 码：用 4 位二进制数来表示 1 位十进制数中的 0~9 这 10 个数码，即二进制表示的十进制数。

汉字内码：计算机内部存储、处理加工和传输汉字时所用的由 0 和 1 符号组成的代码。

码距：一组编码中对应位上数字位不同的最小个数。

奇偶校验：通过检测校验码中 1 的个数的奇/偶性是否改变来判断数据是否出错的一种数据校验方法。

海明校验：是一种基于多重奇校验且具有检测与纠正错误的校验方法。其基本原理是将有效信息按某种规律分成若干组，每组安排一个校验位进行奇偶测试，就能提供多位检错信息，以指出最大可能是哪位出错，从而将其纠正。

循环冗余校验：是数据通信领域中最常用的一种具有检测与纠正错误能力差错校验码，基利用生成多项式并基于模 2 运算建立编码规则。

检错：检测被传送的信息中是否发生差错。

纠错：纠正信息在传送或存储过程中所发生的错误。

2.2 回答下列问题

1)为什么计算机中采用二进制?

答：因为二进制具有运算简单和表示简单的优点，除此之外还有可靠和容易实现等特点。

具体来说，是因为：

（1）技术实现简单，计算机是由逻辑电路组成，逻辑电话通常只有两个状态，开关的接通与断开，这两种状态正好可以用“1”和“0”表示。

（2）简化运算规则：两个二进制数和、积运算组合各有三种，运算规则简单，有利于简化计算机内部结构，提高运算速度。

（3）适合逻辑运算：逻辑代数是逻辑运算的理论依据，二进制只有两个数码，正好与逻辑代数中的“真”和“假”相吻合。

（4）易于进行转换，二进制与十进制数易于互相转换。

2)为什么计算机中采用补码表示带符号的整数?
答：采用补码运算具有如下两个特征：
（1）因为使用补码可以将符号位和其他位统一处理，同时，减法也可以按加法来处理，即如果是补码表示的数，不管是加减法都直接用加法运算即可实现。
（2）两个用补码表示的数相加时，如果最高位（符号位）有进位，则进位被舍弃。
这样的运算有两个好处：
（a）使符号位能与有效值部分一起参加运算，从而简化运算规则。从而可以简化运算器的结构，提高运算速度；（减法运算可以用加法运算表示出来。）
（b）加法运算比减法运算更易于实现。使减法运算转换为加法运算，进一步简化计算机中运算器的线路设计。

3)浮点数的表示范围和精确度分别由什么决定?字长一定时浮点数的表示范围与精确度之间有和关系?
答：浮点数的表示范围由阶码的位数决定，精确度由尾数的位数决定。
当机器字长一定时，分给阶码的位数越多，尾数占用的位数就越少，则数的表示范围越大。而尾数占用的位数减少，必然会减少数的有效数位，即影响数的精度。

4)汉字输入码、机内码和字型码在汉字处理过程中各有何作用？
答：汉字输入码、机内码和字型码，分别用于汉字的输入、汉字在计算机内的处理以及汉字的显示和打印。
具体来说，计算机要对汉字信息进行处理，首先要将汉字转换成计算机可以识别的二进制形式并输入到计算机，这是由汉字输入码完成的；汉字输入到计算机后，还需要转换成内码才能被计算机处理，显然，汉字内码也应该是二进制形式。如果需要显示和打印汉字，还要将汉字的内码转换成字形码。

5)在机内码中如何区分两个 ASCII 码字符和一个汉字?
答：将一个汉字看成是两个扩展 ASCII 码，使表示 GB2312 汉字的两个字节的最高位都为 1，而每个 ASCII 码字符中每个字节的最高位为 0。这样就能区别一个机内码到底对应一个汉字还是两个西文字符。

6)“8421 码就是二进制数”。这种说法对吗？为什么？
答：这种说法是不对的。8421 码是一种最简单的有权码，它选取 4 位二进制数的前 10 个代码 0000～1001 分别对应表示十进制数的 10 个数码。若按权求和，和数就等于该代码所对应的十进制数。
8421 码是一种编码方式，用于十进位制与二进制数之间的转换。
而二进制数是用 0 和 1 两个数码来表示的数。二者是不同的概念，不能等同。

7)如何识别浮点数的正负？浮点数能表示的数值范围和数值的精确度取决于什么？
答：当采用一般浮点数格式表示浮点数时,阶码和尾数都各包含一位符号位。浮点数的正负由尾数的的符号位决定。当采用 IEEE754 格式时，通过数符就能判断出浮点数的正负。
浮点数能表示的数值范围和数值的精确度，分别取决于阶码的位数和尾数的位数。

8)简述 CRC 的纠错原理。
答：发送部件将某信息的 CRC 码传送至接收部件，接收部件收到 CRC 码后，仍用约定的生成

多项式 G(x)去除，若余数为 0，表示传送正确；若余数不为 0，表示出错，再由余数的值来确定哪一位出错，从而加以纠正。具体的纠错原理如下：

（1）不论错误出现在哪一位,均要通过将出错位循环左移到最左边的一位上时被纠正；

（2）不为零余数的具有循环特性。即在余数后面补一个零除以生成多项目式,将得到下一个余数,继续在新余数基础上补零除以生成多项式,继续该操作,余数最后能循环到最开始的余数。

（3）CRC 就是利用不为零余数的循环特性,在循环计算余数的同时，将收到的 CRC 编码同步移动，当余数循环到等于最左边位出错对应的余数时，表明已将出错的位移到 CRC 码的最左边,对出错位进行纠错。

（4）继续进行余数的循环计算,并同步移动 CRC 编码,当余数又回到最开始的值时,纠错后的 CRC 码又回到了最开始的位置。至此，完成 CRC 的纠错任务。

2.4 已知数的补码表示形式，求数的真值。

$[x]_{补}$＝0.10010, $[x]_{补}$＝1.10010, $[x]_{补}$＝1.11111,

$[x]_{补}$＝1.00000, $[x]_{补}$＝0.10001, $[x]_{补}$＝1.00001,

解：

$[x]_{补}$＝0.10010,则$[x]_{原}$＝0.10010，x=0.10010；

$[x]_{补}$＝1.10010,则$[x]_{原}$＝1.01101，x=－0.01101；

$[x]_{补}$＝1.11111,则$[x]_{原}$＝1.00000，x=－0；

$[x]_{补}$＝1.00000，则$[x]_{原}$＝1.11111，x=－0.11111；

$[x]_{补}$＝0.10001，则$[x]_{原}$＝0.10001，x=0.10001；

$[x]_{补}$＝1.00001，则$[x]_{原}$＝1.11110，x=－0.11110。

2.5 已知 x＝0.10110,y＝—0.01010,求：

$[x/2]_{补}$, $[x/4]_{补}$, $[y/2]_{补}$, $[2y]_{补}$

解： $[x]_{原}$=0.10110=$[x]_{反}$=$[x]_{补}$，

所以$[x/2]_{补}$=0.010110，$[x/4]_{补}$=0.0010110；

$[y]_{原}$=1.01010，$[y]_{反}$=1.10101，$[y]_{补}$=1.10110，

所以$[y/2]_{补}$=1.110110，$[2y]_{补}$=1.0110。

2.6 C 语言中允许无符号数和有符号整数之间的转换, 下面是一段 C 语言代码:

```
Int x =-1;
Unsigned u=2147483648;
Printf (“x=%u=%d\n”,x,x);
Printf (“u=%u=%d\n”,u,u);
```

给出在 32 位计算机中上述程序段的输出结果并分析原因.

解：x=4294967295=-1；u=2147483648=-2147483648

原因：x 是 int 型，在计算机中以补码形式存在。%u 以无符号输出，%d 输出真值,所以 x=4294967295=-1。

$u=2^{31}$ 是一个无符号数，无溢出，由于首位为 1

%u 符号输出第一位为非符号位，所以是 2147483648

%d 第一位为符号位，所以是负数，取反加 1 还是 2^{31} 所以是-2147483648。

2.10 求与 IEEE754 32 位浮点数 43940000H 对应的十进制数。

解：

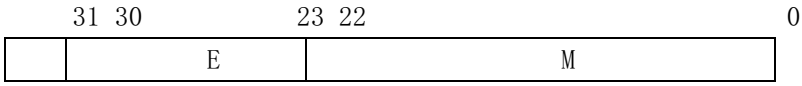
43940000H=（0100 0011 1001 0100 0000 0000 0000 0000）$_2$

S=0，E=（10000111）$_2$-127=8，M=1.00101

所以表示数为 100101000，对应的十进制数为 296。

2.11 求 32 位 IEEE754 浮点数能表示的最大数和最小数。

解：用 IEEE754 格式(E 的取值范围：1~254，留出全 0 和全 1 分别表示 0 和无穷大)



(1) 最大数的二进制表示：

0 11111110 11111111111111111111111 即 $2^{127}(2-2^{-23})$

(2) 最小数的二进制表示：

1 11111110 11111111111111111111111 即 $-2^{127}(2-2^{-23})$

2.15 设有效信息为 01011011，分别写出奇校验码和偶校验码。如果接收方收到的有效信息为 01011010，说明如何发现错误。

解：奇偶校验位分别为：0 和 1，

奇校验码：010110110

偶校验码：010110111

如果采用奇校验，则发送方发出的奇校验码 x=010110110（前 8 位是有效信息位，最后一位是校验位），

如果接收方收到的 x=010110100（只有 1 位出错，最后一个 0 是校验位），

接收方按奇校验方式根据 01011010 计算得到的验位 C′＝1 ，与从信息中读到得校验码的取值不同，表明传送的信息发生了错误。

如果采用偶校验，利用相似的方法可以发现错误。

2.16 由 6 个字符的 7 位 ASCII 编码排列，再加上水平和垂直偶校验位构成如表 2.23 的行列结构（最后一列为水平奇偶校验位，最后一行为垂直奇偶校验位）

表 2.23 ASCII 码交叉校验

字符	7 位 ASCII 码							HP
3	0	X_1	X_2	0	0	1	1	0
Y_1	1	0	0	1	0	0	X_3	1
+	X_4	1	0	1	0	1	1	0
Y_2	0	1	X_5	X_6	1	1	1	1
D	1	0	0	X_7	1	0	X_8	0
=	0	X_9	1	1	1	X_{10}	1	1
VP	0	0	1	1	1	X_{11}	1	X_{12}

则 $X_1X_2X_3X_4$ 处的比特分别为 _1110_； $X_5X_6X_7X_8$ 处的比特分别为 _1000_； $X_9X_{10}X_{11}X_{12}$ 处的比特分别为 _1011_； Y_1 和 Y_2 处的字符分别为 __I__ 和 __7__。

解答思路：利用交叉奇/偶校验原理来确定各个 X 值，再查询 ASCII 码表获知 Y_1 和 Y_2 是什么字符。

2.17 设 8 位有效信息为 01101110，试写出它的海明校验码。给出过程，说明分组检测方式，并给出指误字及其逻辑表达式。如果接收方收到的有效信息变成 01101111，说明如何定位错误并纠正错误。

解：被检验位有 8 位，设检验位有 r 位

因为：$8+r<=2^r-1$

$r=4$;

设四位分别为 P_1，P_2，P_3，P_4

海明码为：$P_1P_20P_3110P_41110$

P1：3，5，7，9，11

P2：3，6，7，10，11

P3：5，6，7，12

P4：9，10，11，12

所以 P1=1，P2=1 P3=0 P4=1

海明码为：110011011110

指错位 G1：1，3，5，7，9，11

G2：2，3，6，7，10，11

G3：4，5，6，7，12

G4：8，9，10，11，12

G1=0，G2=0，G3=0，G4=0

图略。

2.18 设要采用 CRC 编码传送的数据信息 x=1001，当生成多项式为 G（x）＝1101 时，请写出它的循环校验码。若接收方收到的数据信息 x’=1101,说明如何定位错误并纠正错误。

解：作模二除法：$\frac{M(x)\bullet X^3}{G(x)}=\frac{1001000}{1101}=1111+\frac{011}{1101}$

所以循环码为：1001011。

若接收到的数据信息 x’=1101，$\frac{1101011}{G(x)}=1000+\frac{011}{1101}$，所以是第 2 位出错，将第 2 位的 1 改为 0 即可。

第三章

3.1 解释下列名词

变形形补码：即用两个二进制位来表示数据的符号位，其余与补码相同。

溢出：运算结果超出数据类型所能表示数据范围的现象称为溢出。

阵列乘法：采用类似手工乘法运算的方法，用大量与门产生手工乘法中的各乘积项，同时将大量一位全加器按照手工乘法算式中需要进行加运算的各相关项的排列方式组成加法器阵列。

恢复余数除法：比较被除数（余数）与除数的大小是用减法实现的。对原码除法而言，由于操作数以绝对值的形式参与运算，因此，相减结果为正（余数的符号位为 0）说明够减，商上 1；相减结果为负（余数的符号位为 1）说明不够减，商上 0。

由于除法通过减法实现，当商上 1 时，可将比较数据大小时的减法操作与除法操作中的减法操作合并，即商上 1 后继续后面的除法操作。商上 0 时表明不够减，但因比较操作时已经实施了一次减法，因此，需要对执行比较操作后的结果加上除数，既将余数还原成比较操作前的数值，这种方法就称为恢复余数法。

不恢复余数除法：又称加减交替法，是对恢复余数法的改进。不恢复余数法的特点是不够减时不再恢复余数，而根据余数的符号作相应处理就可继续往下运算，因此运算步数固定，控制简单，提高了运算速度。

阵列除法：类似于阵列乘法器的思想，为了加快除法的执行速度，也可以采用阵列除法器来实现除法。为简化运算及阵列除法器的结构，对参加运算的数据进行适当的处理，使其以正数的形式参加运算。

行波进位：多位进位之间存在高位进位的产生依赖低位进位的一种进位方式。

并行进位：高、低进位之间不存在具有依存关系，而是同时计算的进位方式。

算术移位：分为算术左移和算术右移。其中算数左移 n 位相当于乘上 2n,执行方法是把原来的数中每一位都向左移动 n 个位置，左面移出的高位丢弃不要，右面低位空出的位置上全部补 0，当符号位发生改变时表明发生了溢出。算术右移时，符号位保持不变，其余各位依次右移，最右边一位移出，将符号位拷贝到左边空出的位，一次移位相当于除 2。

逻辑移位：逻辑左移 n 位的执行方法，是把原来的数中每一位都向左移动 n 个位置，左面移出的高位丢弃不要，右面低位空出的位置上全部补"0"。 逻辑右移 n 位的执行方法是把原来数中的每一位都向右移动 n 个位置，右面移出的低位丢弃不要，左面高位空出的位置上全部补 0。

对阶：使阶码相等的过程，对阶的时一般采取小的阶码向大阶码看齐的方式。

规格化：就是使浮点数的运算结果中，将尾数从非规格化数变成规格化数的过程。根据尾数形式的不同，规格化可分为左移规格化和右移规格化。

3.2 回答下列问题:

1)为什么采用并行进位能提高加法器的运算速度？

答：由于并行进位电路能很快产生各位的进位信号，使得加法器的速度大大提高。

2)如何判断浮点数运算结果是否发生溢出?

答：由于溢出与数据的表示范围有关，而浮点数的阶码影响到其数据表示的范围，因此，浮点数的溢出是通过接码的是否溢出为判断标志。对于采用双符号位的阶码而言，当双符号位不同时表示浮点数发生溢出，否则则未发生溢出。

3)如何判断浮点数运算结果是否为规格化数?如果不是规格化数,如何进行规格化?

答：当尾数采用补码表示时，若运算结果不是 11. 0××…×或 00. 1××…×的形式时，

结果就不是规格化数。则应进行相应的规格化处理：

•当尾数符号为 01 或 10 时,需要向右规格化，且只需将尾数右移一位，同时将结果的阶码值加 1。

•当尾数运算结果为 11.1××…×或 00.0××…×时需要左移规格化，而且左移次数不固定，与运算结果的形式有关。

左规的方法是尾数连同符号位一起左移位、和的阶码减 1，直到尾数部分出现 11.0 或 00.1 的形式为止。

4)为什么阵列除法器中能用 CAS 的进位/借位控制端作为上商的控制信号?

答：阵列除法器利用不恢复余数的除法，当商上 1 的时候，会产生进位，当商上 0 时，不产生进位，进位信号与上商信号是相同的，所以可以用 CAS 的进位/借位控制作为上商的控制信号。

5)移位运算和乘法及除法运算有何关系?

答：移位运算是乘除法中的基本运算。

3.3 已知x和y，用变形补码计算x＋y，并判断结果是否溢出。

（1） x＝0.11010，y＝0.101110

（2） x＝0.11101，y＝－0.10100

（3） x＝－0.10111，y＝－0.11000

解：（1）$[x+y]_{补}$=01.100010，溢出。

（2）$[x+y]_{补}$=00.01001，未溢出。

（3）$[x+y]_{补}$=10.10001，溢出。

3.7 用补码一位乘法计算 x×y＝?

（1） x＝0.10110，y＝－0.00011

（2） x＝－0.011010，y＝－0.011101

解：（1）

$[x]_{补}$＝0.10110，$[-x]_{补}$＝1.01010，$[y]_{补}$=1.11101

部分积	乘数	$y_n y_{n+1}$	说明
		↓↓	
00.00000		1.111010	y_{n+1}＝0
＋11.01010			$y_{n+1}y_n$＝01，加$[-x]_{补}$
11.01010			
→ 11.10101	0	1.11101	右移一位，得 P_1
＋00.10110			$y_{n+1}y_n$＝10，加$[x]_{补}$
00.01011			
→00.00101	10	1.1110	右移一位，得 P_2
＋11.01010			$y_{n+1}y_n$＝01，加$[-x]_{补}$
11.01111			
→11.10111	110	1.111	右移一位，得 P_3
＋00.00000			$y_{n+1}y_n$＝11，加 0
11.10111			
→11.11011	1110	1.11	右移一位，得 P_4

＋00.00000			$y_{n+1}y_n$＝11，加 0
11.11011			
→11.11101	11110	1.1	右移一位，得 P_5
＋00.00000			$y_{n+1}y_n$＝11，加 0
11.11101	11110		最后一步数据不移位

所以　　$[x \cdot y]_{补}$＝1.1110111110

(2) $[x]_{补}$＝1.100110，$[-x]_{补}$＝0.011010，$[y]_{补}$=1.100011

部分积	乘数	y_ny_{n+1}	说明
		↓↓	
00.000000		1.1000110	y_{n+1}＝0
＋00.011010			$y_{n+1}y_n$＝01，加$[-x]_{补}$
00.011010			
→00.001101	0	1.100011	右移一位，得 P_1
＋00.000000			$y_{n+1}y_n$＝11，加 0
00.001101			
→00.000110	10	1.10001	右移一位，得 P_2
＋11.100110			$y_{n+1}y_n$＝10，加$[x]_{补}$
11.101100			
→11.110110	010	1.1000	右移一位，得 P_3
＋00.000000			$y_{n+1}y_n$＝00，加 0
11.110110			
→11.111011	0010	1.100	右移一位，得 P_4
＋00.000000			$y_{n+1}y_n$＝00，加 0
11.111011			
→11.111101	10010	1.10	右移一位，得 P_5
＋00.011010			$y_{n+1}y_n$＝01，加$[-x]_{补}$
00.010111			
→00.001011	110010	1.1	右移一位，得 P_6
＋00.000000			$y_{n+1}y_n$＝00，加 0
00.001011	110010		最后一步数据不移位

所以　　$[x \cdot y]_{补}$＝0.001011110010

3.9 设数的阶码为 3 位，尾数为 6 位（不包括符号位）按机器补码浮点运算步骤，完成$[x+y]_{补}$

（1） $x=2^{011}\times 0.100100$，$y=2^{010}\times(-0.011010)$

（2） $x=2^{-101}\times(-0.100010)$，$y=2^{-100}\times(-0.010110)$

解：1、（a）对阶 [x]补=00011 00100100 [y]补=00010 11100110

阶差 △E=Ex-Ey=00001，阶差为1

将[y]补尾数右移一位得到00011 11110011

（b）相加 [x]补+[y]补=00011 00010111，尾数相加为00010111

（c）结果规格化 由于尾数符号位跟最高有效位相同，需要左规：

规格化结果为：[x]补+[y]补=00010 00101110

（d）不需舍入，无溢出

则：[x]补+[y]补=00010 00101110

2、（a）对阶 [x]补=11011 11011101 [y]补=11100 11101010

阶差 △E=Ey-Ex=00001，阶差为1

将[x]补尾数右移一位得到11100 11101111

（b）相加 [x]补+[y]补=11100 11011001，尾数相加为11011001

（c）结果规格化 由于尾数符号位跟最高有效位不相同，不需要左规：

（d）不需舍入，无溢出

3.11 用“与或非”门设计一个 4 位并行进位的加法器。

解：$C_1=Y_1+X_1C_0=\overline{\overline{Y_1}\cdot\overline{X_1C_0}}$

$C_2=Y_2+X_2C_1=Y_2+X_2Y_1+X_2X_1C_0=\overline{\overline{Y_2}\cdot\overline{X_2Y_1}\cdot\overline{X_2X_1C_0}}$

$C_3=Y_3+X_3C_2=Y_3+X_3Y_2+X_3X_2Y_1+X_3X_2X_1C_0=\overline{\overline{Y_3}\cdot\overline{X_3Y_2}\cdot\overline{X_3X_2Y_1}\cdot\overline{X_3X_2X_1C_0}}$

$C_4=Y_4+X_4C_3=Y_4+X_4Y_3+X_4X_3Y_2+X_4X_3X_2Y_1+X_4X_3X_2X_1C_0$

$=\overline{\overline{Y_4}\cdot\overline{X_4Y_3}\cdot\overline{X_4X_3Y_2}\cdot\overline{X_4X_3X_2Y_1}\cdot\overline{X_4X_3X_2X_1C_0}}$

电路图略

第四章

4．1　解释下列名词：

存储单元：保存数据的基本内存单元。根据保存内容的大小，一般可分位存储单元，字存储单元等。存储单元一般应具有存储数据和读写数据的功能，每个单元有一个地址，并按地址访问。

存取时间：又称为存储器的访问时间，是指启动一次存储器的操作(读或写分别对应存与取)到该操作完成所经历的时间。

存取周期：连续启动两次访问操作之间的最短时间间隔。

存储器带宽：单位时间内存储器所能传输的信息量，常用的单位包括位/秒或字节/秒。

静态存储器：存储体以静态 MOS 存储元为基本单元组成的存储器称为静态存储器。

动态存储器：存储体以动态存储元为基本单元组成的存储器称为动态存储器。

刷新：动态存储单元中，为使所存信息能长期保存，在电容电荷泄露完之前定时地补充电荷的过程。

猝发式读：只需给出块的起始地址，然后对固定块长度的数据一个接一个地读出的快速存储器读方式。即一块数据的读出只需要给出一个地址的数据读取方式。

多模块交叉存储器：由多个存储容量相同，读写速度相同或相近的多个存储模块构成容量更大的存储器，其中每个存储模块具有各自独立的地址寄存器、地址译码器、驱动电路和读写控制电路，根据存储模块的组织方式不同，又可分为低位交叉和高位交叉两种组织方式。。

高速缓冲存储器：为缓解快速的 CPU 与慢速主存之间的速度差异，在 CPU 和主存之间插入的一至多级速度较快、容量较小的 SRAM，起到缓冲作用；使 CPU 既可以以较快速度存取 SRAM 中的数据，又不使系统成本上升过高。

双端口存储器：指同一个存储器具有两组相互独立端口的存储器，每个端口有各自独立的数据端口、地址端口以及读/写控制端口、片选端口等，每个端口可独立进行读写操作。

相联存储器：是一种按内容访问的存储器(Content Addressable Memory：CAM)，用于提高查找信息的速度。在计算机系统中，相联存储器主要用于虚拟存储器中存放段表、页表和快表以及高速缓冲存储器中的查找。

时间局部性：指当程序访问一个存储位置时，有很大的可能性程序会在不久的将来再次访问同一位置，程序的循环结构和过程调用就很好地体现了时间局部性。

地址映射：指把主存地址空间映射到 Cache 的地址空间，即把存放在主存中的程序或数据按照某种规则装入到 Cache，并建立两者之间地址的对应关系。

组相联映射：地址映射时，主存数据块只能映射到索引字段所指向的 Cache 特定组（其中的行可任选）；地址变换时，需查找的范围也只是索引字段所指向的特定 Cache 组的所有行。

直接映射：地址映射时，主存数据块只能映射到索引字段所指向的 Cache 行中保存；地址变换时，需查找的范围也只涉及索引字段所指向的特定 Cache 行。

全相联映射：主存地址不划分索引字段，因此地址映射时，主存数据块可以映射到 Cache 的任意行中；地址变换时，需查找所有的 Cache 行。

命中率：指 CPU 访问存储系统时，命中 Cache 的次数占总访问次数的比铝。设 N_C 为某程序运行期间命中 Cache 的次数，N_m 为从主存中访问信息的次数，则

命中率（hit ratio）H 定义为：

$$H=\frac{N_C}{N_C+N_m}$$

地址复用：可以从不同的角度来理解该概念。第一种方式是指 CPU 的地址线在一次存储访问过程中多次使用，每次作为访问地址的不同部分使用；另一种是指地址线在一次存储访问的不同阶段分别作为地址线和数据线使用，即地址总线在存储访问的不同时间段表现出不同的功能。

字扩展：用多位满足一定要求的存储芯片构成容量更大的存储器。

位扩展：用多片存储芯片构成位数更多的存储器。

虚拟存储器：是一种解决主存容量不足的存储管理机制，处于存储系统层次结构中“主存-辅存”存储层次。在这种机制下，通过增加部分软件(如操作系统)和必要的硬件(如地址映射与转换机构、缺页中断结构等)，使辅存和主存构成一个有机的整体，就像一个单一的、可供 CPU 直接访问的大容量主存，程序员使用比主存空间大的逻辑地址空间编程序，作业运行时，主需要将作业当前执行的部分调入内存，而其余部分仍然存放在磁盘中，从而减少对主存的消耗。

页表(慢表)：是一张保存虚拟页号和物理页号(也称实页号)之间对应关系的表格。

页表项：页表的表项，每一个表项由有效位和物理页号两部分构成，用于实现虚拟地址与物理地址之间的转换。

TLB(快表)：又称为转换旁路缓冲器(Translation Look- aside Buffer)，为了降低虚拟存储器地址转换的开销，根据局部性原理，将页表的一部分装入 MMU 或 Cache 中，从而减少虚拟地址与物理地址之间转换时访问内存的次数。

LRU：LRU(Least Recently Used)算法是将近期内长久未被访问过的行换出。

LFU：LFU(Least Frequently Used)算法将一段时间内被访次数最少的那行数据换出。

存储保护：为了保证计算机系统能正确运行，当多个用户共享主存时，应防止由于一个用户程序出错而破坏其他用户的程序和系统软件,以及一个用户程序不合法地访问不是分配给它的主存区域。

Cache 一致性：指保存在 cache 中的数据与保存在主存相关单元的数据相同。

写回法：当 CPU 对 Cache 写命中时，只修改 Cache 的内容不立即写入主存，只当 Cache 行被替换时才将 Cache 中的数据写回主存。

写直达法：也称写贯通法或全写法，其基本思想是当Cache写命中时，同时对Cache和主存中同一数据块进行修改；当cache写未命中时，直接向主存写入新的信息，但此时是否将修改过的主存块调入Cache，写直达法却有两种选择。一种是将数据调入Cache，称为写分配法WA(Write-Allocate)。另一种是不取主存块到Cache，而是直接写主存，称为非写分配法NWA(No-Write-Allocate)。

边界对齐的数据存放：指半字、字、双字都按它们各自地址所指定的空间进行存储，而不是随意存放。

大端：存储器的低字节地址单元中存放的是数据的最高字节，高字节地址单元中存放的是数据的最低字节。

RAID：廉价冗余磁盘阵列 RAID(Redundant Array of Inexpensive Disk) 或独立冗余磁盘阵列 RAID(Redundant Array of Independent Disk)，简称磁盘阵列，它将多块独立的普通磁盘按照一定的方式组织与管理，构成一个大容量、高速度、高容错的存储系统。

寻道时间：将磁头定位到指定磁道上所需的时间。

旋转时间：磁头定位到指定磁道后至指定的记录移到磁头下的时间。

4.2 回答下列问题:

1)计算机系统中采用层次化存储体系结构的目的是什么? 层次化存储体系结构如何构成?

答：采用层次化存储体系的目的包括两方面:其一是解决快速的 CPU 和慢速的主存之间的速度差异;其二是解决主存容量不够大的问题.

存储系统的分级结构由 Cache、主存和辅助存储器三级结构构成。

其理论基础是时间局部性原理和空间局部性原理，Cache—主存存储层次解决了主存速度不快的问题；而主存-辅存存储层次解决了主存容量不足的问题。

2)为什么在存储器芯片中设置片选输入端?

答：由于存储芯片的容量及字长与目标存储器的容量及字长之间可能存在差异，应用存储芯片组织一定容量与字长的存储器时，一般可采用位扩展、字扩展、字位同时扩展等方法来组织。这样就会使用多个存储芯片,从而要设置片选输入端来选择正确的存储芯片来进行操作。

3)动态 MOS 存储器为什么要刷新？如何刷新？

答：动态存储单元中，信息以电荷形式存储在 T_1 或 T_2 管的栅极电容中。由于电容容量小，所存电荷会在一段时间后逐渐泄漏(一般为 ms 级)，为使所存信息能长期保存，需要在电容电荷泄露完之前定时地补充电荷，这一过程称为刷新。

刷新的方法：

①刷新方式：集中刷新、分散刷新和异步刷新。前者存在 CPU 死时间；分散刷新由于刷新次数过多，降低了存储器的速度；异步刷新是前两者的折中。

②刷新按行进行,因此设计刷新电路时需要知道动态存储器的内部行、列结构。

③刷新地址由刷新地址计数提供。

4)试述多体交叉存储器的设计思想和实现方法。

答：多体交叉存储器由多个存储模块构成，这些模块的容量和存取速度相同，具有各自独立的地址寄存器、地址译码器、驱动电路和读写控制电路。根据对多各模块编址方式的不同，又可分为高位多体交叉和低位多体交叉两种方式。

(1)高位交叉：按存储器地址的高位地址划分模块，同一存储体内的地址是连续的。当多个目标同时访问存储器时(如 CPU 和 DMA 设备同时访问存储器),如果访问的地方范围处于不同的存储芯片，则提供并行访问。

(2)低位交叉：按存储器地址的低位地址划分模块，同一存储体内的地址不相邻，相邻地址处在不同存储体中。CPU 可同时启动多个存储体，并进行并行访问。

5)为什么说 Cache 对程序员是透明的?

答:因为在程序员看来,数据是在内存和辅存之间进行交换的,程序员感觉不到中间层 cache 的存在。

6)直接映射方式下为什么不需要使用替换算法?

答：因为在直接映射方式中，一个内存块只能固定的映射到 cache 中的特定行，当有新的主存块调入时， cache 特定行中的内容必须调出,因此不需要替换算法去选择替换掉哪一块。

7)为什么要考虑 Cache 的一致性?

答：正常情况下,cache 中的数据是主存数据的副本,当两者不一致时可能导致程序结果不正确,因此,必须考虑并设法保证 Cache 的一致性。

8)替换算法有哪几种?各有何优缺点?

答：① 先进先出算法(FIFO)

基本思想：按照数据块进入 Cache 的先后决定替换的顺序，即在需要进行替换时，选择最先被调入 Cache 中的块作为替换块。这种方法要求为每块记录它们进入 Cache 的先后次序。

优点：FIFO 算法系统开销较小。

缺点：是不考虑程序访问的局部性，可能会把一些需要经常使用的块（如循环程序块）也作为最早进入 Cache 的块而替换掉，因此，可能导致 Cache 的命中率不高。

② 近期最少使用(LRU)算法

基本思想：将近期内长久未被访问过的行换出。为此，每行设置一个计数器，cache每命中一次，命中行计数器清零，其它各行计数器增1，因此它是未访问次数计数器。当需要替换时，比较各特定行的计数值，将计数值最大的行换出。

优点：这种算法显然保护了刚调入Cache的新数据，符合cache工作原理，因而使cache有较高的命中率。LRU算法硬件实现简单

③ 最不经常使用(LFU)算法

基本思想：将一段时间内被访次数最少的那行数据换出。为此，每行设置一个计数器，新新调入行的数据从0开始计数，每访问一次被访行的计数器增1。当需要替换时，对这些特定行的计数值进行比较，将计数值最小的行换出。

缺点：一段期间访问情况不能严格反映近期访问情况。例如特定行中的A、B两行，A行在期间的前期多次被访问而后期未被访问，但累积计数值很大，B行是前期不常用而后期却正被频繁访问，但可能由于累积计数小于A行而被LFU算法换出了。

④ 随机替换算法

基本思想：需要进行替换时，从特定的行位置中随机地选取一行进行替换。

优点：硬件实现最容易，而且速度也比前几种策略快。

缺点：随意换出的数据很可能马上又要用，从而降低命中率和cache工作效率。但这个负面效应随着cache容量增大会减少，模拟研究表明随机替换策略的功效只是稍逊于LFU和LRU。

4.3 对于32KB存储器,若按16位字编址,其地址寄存器应是多少位？数据寄存器是多少位？

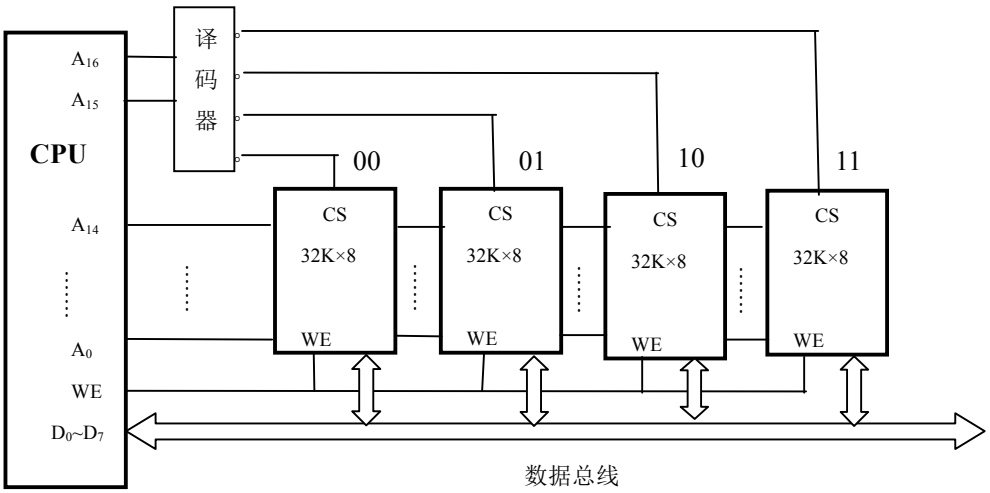
解：该存储器的寻址空间为：32K*8 位/16 位=16K 字

地址寄存器的位数为：14 位

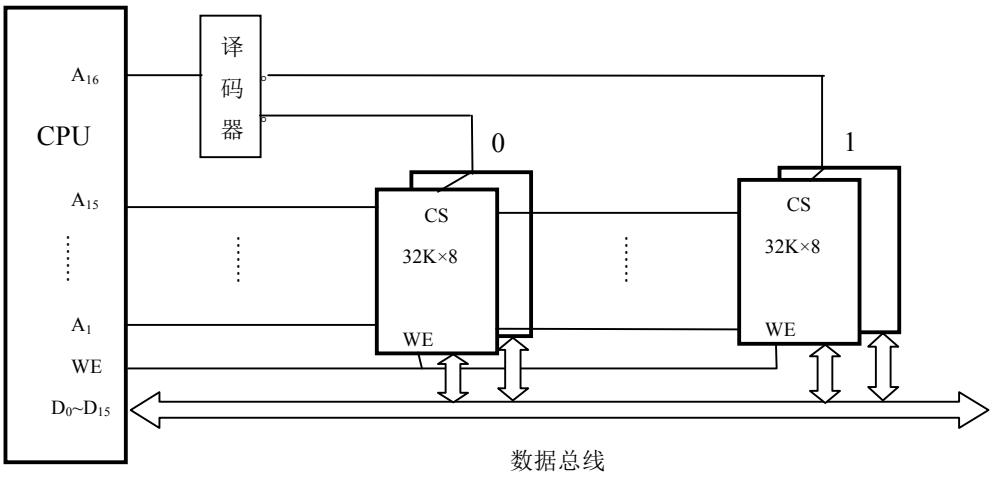
数据寄存器的位数和字长相等为 16 位

4.4 用 4 片 32K×8 位 SRAM 存储芯片可设计哪几种不同容量和字长的存储器？画出相应设计图并完成与 CPU 连接。

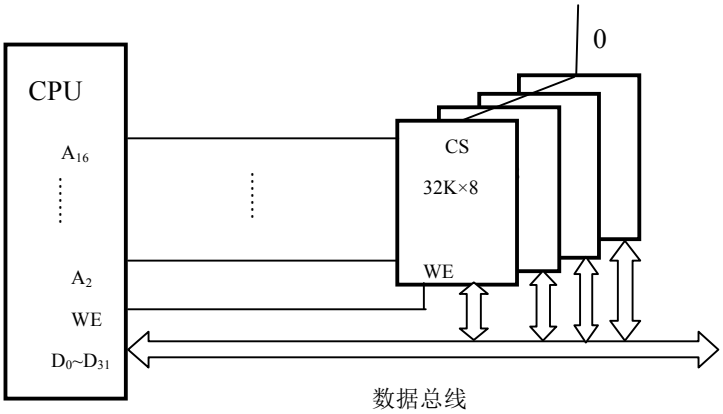
解：可设计字长为 8 位，容量为 128K 的存储器：



可设计字长为 16 位，容量为 64K 的存储器：



可设计字长为 32 位，容量为 32K 的存储器：



4.5 用 32K×8 位 RAM 芯片和 64K×4 位 ROM 芯片，设计 256K×8 位存储器。其中，从 30000H 到 3FFFFH 地址空间为只读存储区，其它为可读、可写存储区。完成存储器与 CPU 连接。

解：只读区域的地址空间为:30000H-3FFFFH,为 64K,需要 64K×4 位 ROM 芯片 2 片，需要 32K×8 位 RAM 芯片的片数为：256K-64K/32K=6 片

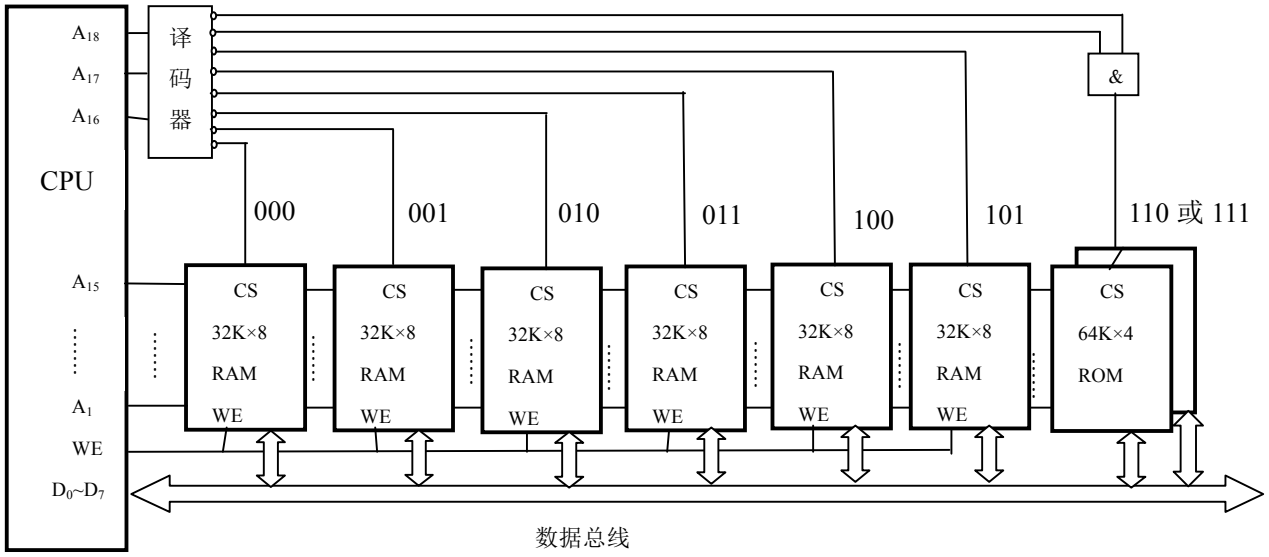
设计如下：存储器的 0000H-2FFFFH 存储空间为 RAM 芯片，也就是 32K×8 位 RAM 芯片 6 片，采用字扩展连接。存储空间 30000H-3FFFFH 使用 64K×4 位 ROM 芯片 2 片，采用位

扩展方式连接。

数据线条数为 8 条：D_0-D_7。

地址线的条数为 18 条：A_1-A_{18}，其中 A_{18}-A_{16} 为片选信号的输入端。

设计图如下:



4.8 用 64K×1 位的 DRAM 芯片构成 1M×8 位的存储器，若采用异步刷新，若每行刷新间隔不超过 2ms，则产生刷新信号的间隔是多少时间？若采用集中刷新方式，则存储器刷新一遍最少用多少个读写周期？CPU 的死时间为多少？（假定存储器的读写时间为 0.5μs）

解：64K×1 位的 DRAM 芯片的排列方式为 256 行*256 列，该存储器中有 64K×1 位的 DRAM 芯片 128 片刷新信号的产生间隔为 2ms.

将 2ms 分成 256 个小段(因为 DRAM 按行刷新)，每个时间段为：7.8125μs,将其中最后 0.5μs 用于刷新 DRAM 的一行，即产生刷新信号的时间间隔为 7.8125μs.

若采用集中刷新,存储器刷新一遍至少需要 256 个读写周期,CPU 的死时间是 256×0.5μs=128μs

4.9 某动态 RAM 芯片，容量为 64K×1 位，除电源线、接地线和刷新线外，该芯片的最小引脚数量是多少？

解：该芯片 1 位数据线，行选通和列选通各一位，64K 的存储器对应 16 根地址线，在 DRAM 中，行和列复用，即地址线为 8 根，故在不考虑电源线的情况下，该 DRAM 芯片的最小引脚数为 1+1+1+8 = 11 个。

4.12 设 Cache 的容量为 2^{14} 块,每块是一个 32 位字,主存容量是 Cache 容量的 256 倍,其中有如表 4.11 所示数据(地址和数据均采用 16 进制表示).

表 4.11 主存数据分布情况

地址	数据
000000	87568536
000008	87792301
010004	9ABEFCD0
01FFFC	4FFFFC68

FFFFF8	01BF2460

将主存中这些数据装入到 Cache 后，Cache 各块中的数据内容及相应的标志是什么?

(1)全相联映射

(2)直接相联映射

(3)组相联映射

解：(1)全相联映射

全相联映射方式下，主存的一个数据块可映射到 Cache 的任意行，表中共有 5 个地址，数据从主存映射到 Cache 后占用其中的 5 行，假设就是 Cache 的前 5 行，具体分布如下表所示。

Cache 行	标志	数据
0	00,0000,0000,0000,0000,0000 (000000H)	87568536
1	00,0000,0000,0000,0000,0010(000002H)	87792301
2	00,0000,0100,0000,0000,0001(004001H)	9ABEFCD0
3	00,0000,0111,1111,1111,1111(007FFFH)	4FFFFC68
4	11,1111,1111,1111,1111,1110(03FFFFEH)	01BF2460

(2)直接相联映射

该方式下，主存的一个数据块只能射到 Cache 的特定行，表中共有 5 个地址，数据从主存映射到 Cache 后占用其中的特定的 5 行。由于一个数据块为 32 位，即 4B，所以，只需要用主存地址的最后 2 位表示块内偏移地,由于 Cache 有 16K 行,所有,去掉最后 2 位地址后的连续 14 位就表示主存数据块映射到的 Cache 行,剩余的 8 位即为对应的标志.具体分布如下表所示。

Cache 行	标志	数据
00 0000 0000 0000(0000 行)	0000 0000	87568536
00 0000 0000 0010(0002 行)	0000 0000	87792301
00 0000 0000 0001(0001 行)	0000 0001	9ABEFCD0
11 1111 1111 1111(03FFF 行)	0000 0001	4FFFFC68
11 1111 1111 1110(03FFE 行)	1111 1111	01BF2460

(3)组相联映射：(假设采用的是四路组相联 1)

该方式下，主存的一个数据块只能射到 Cache 的特定组中的任意行，假定 Cache 采用四路组相联，则 Cache 共分为 4K 组。由于一个数据块为 32 位，即 4B，所以，只需要用主存地址的最后 2 位表示块内偏移地，由于 Cache 有 4K 组，因此，去掉最后 2 位地址后的连续 12 位就表示主存数据块映射到的 Cache 组,剩余的 10 位即为对应的标志.具体分布如下表所示。

Cache 组	标志	数据
0000 0000 0000(000 组任意行)	00 0000 0000	87568536
0000 0000 0010(002 组任意行)	00 0000 0000	87792301
0000 0000 0001(001 组任意行)	00 0000 0100	9ABEFCD0
1111 1111 1111(0FFF 组任意行)	00 0000 0111	4FFFFC68
1111 1111 1110(0FFE 组任意行)	11 1111 1111	01BF2460

4.13 某计算机 Cache 由 64 个存储块构成,采用四路组相联映射方式.主存包含 4096 个存储块,每块由 128 个字组成.访问地址为字地址.

(1)求主存地址和 Cache 地址各有多少位?

(2)按照题目条件中的映射方式,列出主存地址的划分情况,并标出各部分的位数.

解：(1)主存容量为：4096*128=512KW

故主存地址位数为：19 位

Cache 容量为：64*128=8KW

故 Cache 地址位数为：13 位

(2)每个组中包含的存储块的个数为：4 块,Cache 组数 16

故索引字段(Index)位数为：4 位

由于每块由 128 个字组成.访问地址为字地址，故块内地址的位数为：7 位

故标记部分(Tag)的位数为：8 位

则主存地址划分情况如下：

4.14 某计算机中主存容量为 4MB,Cache 容量为 16KB,每块包含 8 个字,每字 32 位,映射方式采用 4 路组相联.设 Cache 的初始状态为空,CPU 依次从主存第 0,1,2,…,99 号单元读出 100 个字(每次读一个字),并重复此操作 10 次.替换算法采用 LRU.

(1)求 Cache 的命中率

(2)若 Cache 比主存块 10 倍,分析采用 Cache 后存储访问速度提高了多少?

解：(1)0,1,2,…,99 号单元共 100 个字，每块 8 个字，故 100 个字被分配在 13 块内。

Cache 中能存放的块数为：16KB/（8*4）=512 块

因为是四路组相联，故 Cache 组数为：512/4=128

由于 Cache 的初始状态为空，根据前面的分析，13 块数据调入 Cache 后不会被调出，所有 10 次访问中，每块第一次访问不命中外，其余访问均可命中，因此 10 次循环访问共访问内存 100*10 =1000 次，其中不命中的次数只有 13 次。

则 Cache 的命中率为：(1000-13)/1000=98.7%

(2)设访问 Cache 的访问时间为 T（访问一个数据单元所用的时间），则访问主存的访问时间为 10T,故有：

使用 Cache 后访存所用时间为：T2=13*10T+(1000-13)*T=1117T

不使用 Cache 访问耗时为：T1=10000T

故使用了 Cache 后速度提高了：10000T/1117T=8.95 倍

课堂习题：某计算机字长 32 位，采用直接映射 Cache,主存容量 4MB，Cache 数据存储体容量为 4KB，字块长度为 8 个字。

(1) 画出直接映像方式下主存地址划分情况。

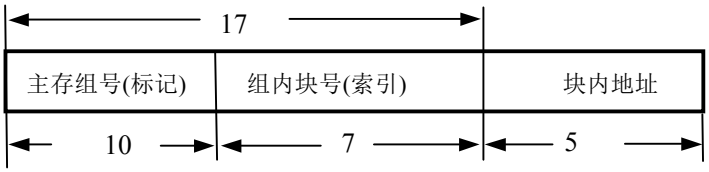
(2) 设 cache 初始状态为空，若 CPU 顺序访问 0-99 号单元，并从中读出 100 个字，假设主存一次读一个字，并重复此顺序 10 次，请计算 cache 命中率。

(3) 如果 cache 的存取时间是 2ns，主存访问时间是 20ns，平均访问时间是多少。

(4) Cache-主存系统访问效率

解：
(1)主存有 4MB/4B=1M 字，地址需要 20 位。Cache 有 4KB/4B=1K 字，1K/8=128 块(行)，需要 7 位。字块大小 8 个字，需要 3 位。
区　　行　　字
++++ ++++ ++== ==== =___
0000 0000　0000 0000 0000
(2)
地址 0：行号：0
0000 0000 0000 0000 0000
地址 100: 行号 12
0000 0000 0000 0110 0100
第一次 100 字
100/8 + 1=13 次载入， 100-13=87 次命中
第二次到第 10 次，全命中
9*100 = 900 次命中。
命中率=（87+900)/1000
(3)
T = 0.987 * 2 + 0.013 * 20 = 2.234ns
(4)
E = 2 / 2.234 = 0.895

4.18 某计算机系统中有一个 TLB 和 L1 级数据 Cache,存储系统按字节编址,虚拟存储容量为 2GB, 主存容量为 4MB,页大小为 128KB,TLB 采用四路组相联方式,共有 16 个页表项.Cache 容量为 16KB,每块包含 8 个字,每字 32 位,采用四路组相联.参照图 4.51 所描述的页式虚拟存储系统中 TLB 和 Cache 命中时详细的存储访问过程,回答下列问题:
(1)虚拟地址中哪几位表示虚页号?哪几位表示页内偏移地址?虚页号中哪几位表示 TLB 标记?哪几位表示 TLB 索引?
(2)物理地址中哪几位表示物理页号?哪几位表示偏移地址?
(3)为实现主存与数据 Cache 之间的组相联映射,对该地址又进行怎样的划分?
解：(1)虚拟存储的页面数为：2GB/128KB=16K，故虚页号的位数为：14 位
由于页大小为 128KB，故页内偏移地址的位数为 17 位
所以虚拟地址的高 14 位表示的是虚页号，低 17 位表示的是页内偏移地址
TLB 采用四路组相联方式,共有 16 个页表项，所以索引需要 2 位，故虚拟页号的高 12 位为 TLB 标记，低 2 位为 TLB 索引
(2)物理内存中包含的页数为：4MB/128KB=32，故物理页号占 5 位，页内偏移地址的位数为 17 位
所以物理地址中高 5 位表示物理页号，低 17 位表示页内偏移地址
(3)主存容量为 4MB,访问该主存需要 22 根地址线,块大小为 32B,故块内字节偏移地址为 5 位, Cache 采用四路组相联，Cache 共分成 16KB/(4*32B) = 128 个组，故索引字段需要 7 位，剩下的 22-5-7 = 10 位为标记。
则主存地址划分如下：


17
主存组号(标记)
组内块号(索引)
块内地址
10
7
5

第五章

5. 1 解释下列名词

指令　指令系统　操作码　地址码　寻址方式　程序计数器 PC　有效地址　地址码扩展　CISC　RISC　存储器堆栈 寄存器堆栈 基址寄存器 变址寄存器

解：(1)指令：控制计算机执行某种操作(如加、减、传送、转移等操作)的命令称为指令。

(2)指令系统：一台计算机中所有指令的集合称为该计算机的指令系统。

(3)操作码：指令中用于控制指令操作性质的字段称为操作码。不同功能的指令其操作码编码不同,如可用 0001 表示加法操作，0010 表示减法操作。

(4)地址码：指令中用于定参与指令操作的操作数的地址或偏移量地址的字段。

(5)寻址方式：寻找指令或操作数有效地址的方法。

(6)程序计数器 PC：程序计数器是用于存放下一条指令所在单元的地址的寄存器。

(7)有效地址：表示操作数所在主存单元的物理地址。

(8)地址码扩展：将指令的操作码字段向不用的地址码字段扩展，从而在指令长度不变的情况下支持更多的指令。

(9)CISC：CISC 是复杂指令系统计算机（ComplexInstructionSetComputer）的简称，这类计算机指令系统复杂，寻址方式种类较多，指令执行效率低。

(10)RISC：RISC 是精简指令集计算机（reduced instruction set computer，）的简称，这类计算机指令系统简单，寻址方式种类少，指令执行效率高。

(11)存储器堆栈：以先进后出的方式存储数据，在内存空间开辟堆栈区，该类堆栈容量大，速度慢，栈顶移动而堆栈中的数据不动。

(12)寄存器堆栈：以先进后出的方式存储数据，利用寄存器开辟的堆栈区，该类堆栈容量小，速度块，栈顶不动，出栈和入栈操作设计栈内所有数据的移动。

(13)基址寄存器：基址寻址方式下用于存放基地址的寄存器。

(14)变址寄存器：变址寻址方式下，用于存放变化的地址的寄存器。

5.2 简要回答下列问题

(1)什么叫指令？什么叫指令系统？

(2)计算机中为什么要设置多种操作数寻址方式?

(3)操作数寻址方式在指令中如何表示?

(4)基址寻址和变址寻址的作用是什么?分析它们的异同点.

(5)RISC 处理器有何特点?

(6)比较定长指令与变长指令的优缺点。

(7)指令的地址码与指令中的地址码含义有何不同?

解:(1) 指令是指控制计算机执行某种操作(如加、减、传送、转移等操作)的命令，而一台计算机中所有指令的集合称为该计算机的指令系统。

(2) 这是为了在效率和方便性以及寻址空间大小保持平衡。

用于快速访问的寻址方式：立即数寻址、寄存器寻址等

扩大寻址范围的寻址方式：间接寻址、寄存器间接寻址、基址寻址等

便于程序设计灵活性的寻址方式：变址寻址、相对寻址、直接寻址等

既扩大寻址范围，又由利于指令执行速度提高的寻址方式：寄存器间接寻址

另外，多种复合寻址寻址方式使得寻址更加灵活。

(3) 由于不同指令可能采用不同的寻址方式获得操作数, 因此, 一般情况下，指令的格式会进一步细分出寻址方式字段。下图所示的为包含寻址方式字段的单地址指令结构。

OP	I	D

其中，OP 为操作码，I 为寻址方式特征码。D 为形式地址，或称偏移量。寻址过程就是把 I 和 D 的不同组合变换成有效地址的过程。I 与操作数寻址方式相关。

(4) 基址寻址面向系统，主要用于程序的重定位和扩展寻址空间。变址寻址是面向用户的，主要解决程序循环问题。

两者相同点：在形式上以及计算操作数的有效地址的方法上，变址寻址和基址寻址中是相似的，都是把个寄存器的内容加上指令字中的形式地址而形成操作数有有效地址。

不同点：两者有着不同的用途。首先，在采用了基址寻址的计算机系统中，基址是不变的，程序中的所有地址都是相对于基地址来变化的。而对于变址寻址来说则相反，指令中的地址字段的形式地址给出的是一个存储器地址基准，变址寄存器 X 中存放的是相对于该基准地址的偏移量。不同的变址寄存器给出的不同的单元。第二，在基址寻址中，偏移量位数较短，而在变址寻址中，偏移量位数足以表示整个存储空间。第三，基址寻址主要是解决程序逻辑空间与存储器物理空间的无关性，而变址寻址主要是为了可以编写出高效访问一片存储空间的程序。

(5) RISC 具有如下特点：使用等长指令、寻址方式少且简单、只有取数和存数指令访问存储器、指令数量和指令格式少于、指令功能简单、CPU 内部设置了大量的寄存器、控制器多采用硬布线方式、大多数指令可在一个时钟周期内完成、支持指令流水并强调指令流水的优化使用。

(6) 定长指令的优点：定长指令具有结构规整，有利于简化硬件，尤其是指令译码部件的设计。

定长指令的缺点：定长指令平均长度长、容易出现冗余码点和指令不易扩展等不足。

变长指令的优点：变字长指令结构灵活，能充分利用指令中的每一位，所以指令码点冗于少，指令的平均长度短，易于扩展。

变长指令的缺点：变长指令的格式不规整，不同指令的取指时间可能不同，导致控制复杂。

(7) 指令的地址码通常指定参与操作的操作数的地址。指令中的地址码字段的作用随指令类型和寻址方式的不同而不同，它可能作为一个操作数、也可能是操作数的地址（包括操作数所在的主存地址、寄存器编号或外部设备端口地址）、也可能是一个用于计算地址的偏移量。

5.5 假设某计算机的指令长度固定为 16 位，具有双操作数，单操作数和无操作数三类指令，每个操作数地址规定用 6 位表示。

(1)现已设计出 m 条双操作数指令，n 条无操作数指令，在此情况下，这台计算机最多可设计出多少条单操作数指令？

(2)当双操作数指令取最大数，且在此基础上，单操作数指令条数也取最大值，试计算这 3 类指令最多可拥有多少条指令？

解：根据题目提交，该指令格式包含两个操作数字段（各 6 位）和一个操作码字段（4 位），单操作数指令通过将指令的 OP 字段向不用的一个地址码字段扩展，零操作数指令在单操作数指令的基础上继续向另一个不用的地址码字段扩展。在扩展的过程中要注意留出扩展标志。

（1）可通过扩展的方式将操作码向地址码扩展。

则计算机最多有 2^4=16 条指令双操作数指令

这里，双操作数指令为 m 条，此时，操作码字段还有(16-m)种扩展标志,用于表示操作码向一个地址字段扩展。

本题目中已知道无操作数指令为 n 条，首先要知道，无操作数指令是在一操作数

指令的基础上，通过将操作码向最后一个地址字段扩展得到的，加上扩展的过程中用到了 y 个扩展标志，则 n = y× 64 (每种扩展标志下，最后地址字段的 6 位全部标志无操作数指令)，由此，可以得到 y = n / 64

所以：单操作数的指有(16-m)*64- n / 64 条。

（2）双操作数指令取最大数：2^4 -1=15 条

单操作数指令取最大数：2^6 -1=63 条

无操作数指令最大有：2^6=64 条

5.6 设相对寻址的转移指令占三个字节，第一个字节是操作码，第二个字节是相对位移量（补码表示）的低 8 位，第三个字节是相对位移量（补码表示）的高 8 位，每当 CPU 从存储器取一个字节时，便自动完成（PC）+1→PC:

(1)若 PC 当前值为 256（十进制），要求转移到 290（十进制），则转移指令第二、三字节的机器代码是什么（十六进制）？

(2)若 PC 当前值为 128（十进制），要求转移到 110（十进制），则转移指令第二、三字节的机器代码又是什么（十六进制）？

解：根据相对寻址有效地址的计算公式 EA=（PC）+ D 可知

D = EA - （PC）

对于相对寻址而言，关键是要求出计算有效地址时 PC 的当前值。

(1) 根据题意，采用相对寻址指令的地址为 256，PC 在取指令完成后修改，则取指令完成后 PC=256+3=259 （因为指令长度为 3 个字节）

所以：D = 290-259 =$(31)_{10}$= 001FH

故：转移转移指令第二字节为：1FH,第三字节为：00H。

(2) 根据题意，采用相对寻址指令的地址为 128，PC 在取指令完成后修改，则取指令完成后 PC=128+3=131

所以：D = 110-131=$(-21)_{10}$=FFEDH(补码)

故：转移转移指令第二字节为：EDH,第三字节为：FFH。

5.8 某计算机的指令格式包括操作码 OP、寻址方式特征位 I 和形式地址 D 等三个字段,其中 OP 字段 6 位,寻址方式特征位字段 I 为 2 位,形式地址字段 D 为 8 位。I 的取值与寻址方式的对应关系为:

I=00:直接寻址

I=01:用变址寄存器 X1 进行变址;

I=10:用变址寄存器 X2 进行变址;

I=11:相对寻址.

设(PC)=1234H,(X1)=0037H , (X2)=1122H,以下四条指令均采用上述格式,请确定这些指令的有效地址:

(1)4420H (2)2244H (3)1322H (4)3521H

解：（1）4420H=（0100010000100000）B

OP	I	D
0 1 0 0 0 1	0 0	0 0 1 0 0 0 0 0

I＝ 0 0 ， 直接寻址方式，EA= 20H

（ 2 ）2244H=0010 0010 0100 0100B

OP	I	D

0 0 1 0 0 0	1 0	0 1 0 0 0 1 0 0

I＝１０：用变址寄存器 X 2 进行变址

Ｅ＝（X 2 ）＋D＝１１６６H

（３）1322H=0001 0011 0010 0010B

OP	I	D
0 0 0 1 0 0	1 1	0 0 1 0 0 0 1 0

I＝１１：相对寻址

Ｅ＝（PC）+2＋D＝１２５8H　（双字节长指令）

（４）3521H=0011 0101 0010 0001B

OP	I	D
0 0 1 1 0 1	0 1	0 0 1 0 0 0 0 1

I＝０１：用变址寄存器 X 1 进行变址

Ｅ＝（X 1 ）＋D＝００５８H

5.9 某计算机 A 有 60 条指令 ,指令的操作码字段固定为 6 位,从 000000-111011,该机器的后续机型 B 中需要增加 32 条指令,并与 A 保持兼容,

(1)试采用操作码扩展方法为计算机 B 设计指令操作码.

(2)计算计算机 B 中操作码的平均长度.

解：(1) 因为计算机 B 要与计算机 A 兼容所以计算机 A 中的指令得保留：所以 000000-111011 为 A 的操作码部分。操作码字段的 11100-111111 的取值将作为扩展标识，将操作码扩展到地址字段,只需要占用地址字段 3 位即可表示新的 32 条指令。

(2)由(1)可知，有 60 条指令的操作码为 6 位，32 条指令的操作码为 9 位

所以平均长度为：(60*6+32*9)/92=7.74 位.

第六章

6.1 解释下列名词

指令周期　数据通路　时钟周期　同步控制　异步控制　联合控制 单周期处理器　多周期处理器 微操作　相容性微命令　互斥性微命令　微指令　微程序　微程序控制器　控制存储器　硬布线控制器

6.1. 答：

指令周期：取指令并执行一条指令所需要的时间，一般由若干个机器周期组成，包括从取指令、分析指令到执行完所需的全部时间。

数据通路：数据在功能部件之间传送的路径。

时钟周期：由 CPU 时钟定义的定长时间间隔，是 CPU 工作的最小时间单位，也称节拍脉冲或 T 周期。

同步控制：选取部件中最长的操作时间作为统一的时间间隔标准，使所有部件都在这个时间间隔内启动并完成操作。

异步控制：系统不设立统一的时间间隔标准(基准时钟除外)，各部件按各自的时钟工作，分别实现各自的时序控制，时间衔接通过应答通讯方式(又称握手方式)实现。

联合控制：同步控制与异步控制相结合。对大多数节拍数相近的指令，采用同步控制；而对少数节拍数多不固定的指令，采用异步控制。

单周期处理器：所有指令在一个时钟周期内完成的处理器。

多周期处理器：每条指令的执行分成多个阶段，每个时钟周期完成一个阶段的工作。

微操作：执行部件收到微命令后所进行的操作。

相容性微命令：能同时并行执行的微命令。

互斥性微命令：不能并行执行的微操作。

微指令：由微指令产生的一组实现一定微操作功能的微命令的组合。

微程序：实现一条指令功能的若干条微指令的集合。

微程序控制器：采用微程序设计方法设计的控制器。指令执行过程中所需要的所有控制信号以微指令的方式存在在控制存储器中，指令执行时，逐条读出微指令，以产生执行执行过程中所需要的控制信号。

控制存储器：微程序控制器中用于存放解释所有指令微程序的存储器。

硬布线控制器：又称为组合逻辑控制器，指令执行所需要的控制信号直接由逻辑门电路和触发器等构成的电路产生，与微程序控制器相比，具有结构复杂但速度快的特点。

6.2 回答下列问题

1)中央处理器的基本功能是什么?从实现其功能的角度分析，它应由哪些部件组成?

答：五方面的功能：

指令执行顺序的控制。即控制程序中的指令按事先规定的顺序自动地执行，从而保

证程序执行过程中，指令在逻辑上的相互关系不被改变。

指令的操作控制。即产生指令执行过程中所需要的信号，以控制执行部件按指令规定的操作运行。

时间控制，即对每个控制信号进行定时，以便按规定的时间顺序启动各操作。对于任何一条指令而言，如果操作控制信号的时间不正确，则指令的功能也就不能正确实现。

数据加工处理。即对数据进行算术、逻辑运算，或将数据在相关部件之间传送。

异常和中断处理。如处理运算中的异常及处理外部设备的中断服务请求等。

组成：中央处理器主要由控制器和运算器两部分构成。控制器的主要功能包括：取指令、计算下一条指令的地址、对指令译码、产生相应的操作控制信号、控制指令执行的步骤和数据流动的方向。运算器是执行部件，由算术逻辑单元和各种寄存器组成。

2)CPU 内部有哪些寄存器?它们的功能分别是什么？

答：**(1) 指令寄存器(IR)：**IR 用于保存指令。从主存储器取出的指令存放在 IR 中，直到新的指令从主存中取出为止。

(2) 程序计数器(PC) ：PC 保存将要执行的指令地址，故又称指令地址寄存器。CPU 取指令时，将 PC 的内容送到主存地址寄存器，然后修改 PC 的值形成下一条将要执行的指令地址

(3) 地址寄存器(AR)：AR 用来保存当前 CPU 所要访问的主存单元地址，无论 CPU 是取指令还是存取数据，都必须先将要访问的主存单元地址送 AR，直到读/写操作完成。

(4) 通用寄存器组(GR)：通用的含义是指寄存器的功能有多种用途，GR 可作为 ALU 的累加器、变址寄存器、地址指针、指令计数器、数据缓冲器，用于存放操作数(包括源操作数、目的操作数及中间结果)和各种地址信息等。

(5) 数据缓冲寄存器(DR)

DR 作为 CPU 和主存之间的数据缓冲寄存器用于存放操作数、运算结果或中间结果，以减少访问主存的次数；也可存放从主存中读出的数据，或准备写入主存的数据。

(6) 程序状态字寄存器(PSW)

PSW 用于保存由算术运算指令、逻辑运算指令、测试结果等建立的各种条件标志。常见的状态信息包括进位标志(C)、溢出标志(V)、结果为负数标志(S)及结果为零标志(Z)等。

3)什么是取指周期?取指周期内应完成哪些操作?

答：取指周期就是从开始取指令到取指令完成所需要的时间。取指周期要完成两方面的操作，一是将 PC 的值送存储器地址寄存器 MAR，并完成储单元去取指令；二是如何形成后续指令地址：

•顺序执行指令时，将 PC 内容加当前指令所占用的主存单元数(以字节为单位)；

•当出现转移时，根据寻址方式、转移条件、转移的目标地址等内容计算得到。

4)指令有几种执行方式?说明各自的特点。

答：指令的执行方式有顺序执行方式、重叠执行方式和流水执行三种方式。

顺序执行方式：是一种串行执行方式，取出一条指令的操作全部结束后才能开始下一条指令的指令周期，这种方式控制简单，程序的执行速度慢。

重叠执行方式：是一种局部并行方式，通常将当前指令的执行阶段与下一条指令的取指阶段重叠进行，这种方式控制较复杂，但可以提高程序的执行速度；

流水执行方式：是一种并行执行方式，它将指令的执行分多个阶段（每个阶段的任务由特定的功能部件完成），一般而言，在该执行方式下，指令间的并行程度比重叠执行方式

要高，控制更为复杂，可以更快地提高程序的执行速度。

5)计算机为什么要设置时序系统?说明指令周期、机器周期、和时钟周期的含义。

答：指令执行过程中的所有操作必须按照一定的次序完成，而且这些操作持续的时间也有严格的限制，因此，在计算机系统中需要设置时序系统，对指令执行过程中的所有控制信号进行时间控制，以保证指令功能的正确实现。

通常将一条指令从取出到执行完成所需要的时间称为**指令周期，**包括取指周期和执行周期，指令周期通过右若干和机器周期组成，所包含的机器周期的数量随指令功能和寻址方式的不同而不同。

机器周期分成若干个节拍电位时间段，通常以 CPU 完成一次微操作所需要的时间为基础来定义节拍电位的时间；由 CPU 时钟定义的定长时间间隔，是 CPU 工作的最小时间单位，也称**节拍脉冲或 T 周期**。

6)组合逻辑控制器与微程序控制器各有什么特点?

答：硬布线控制器又称为组合逻辑控制器，这种控制器中的控制信号直接由各种类型的逻辑门电路和触发器等构成，与微程序控制器相比，具有结构复杂但速度快的特点。

微程序控制器的设计采用了存储技术和程序设计技术，使复杂的控制逻辑得到简化。通过过读出存放在微程序控制器中微指令产生指令执行过程中所需要的控制信号，所以，与硬布线控制器相比，微程序控制器的速度较慢。

7)说明程序与微程序，指令与微指令的异同

答：微程序是多条微指令系列的集合,用于实现指令的功能,属于机器指令级别，对用于的透明性不强，存放在 CPU 内的控制存储器中；程序则是为了完成某一应用功能所编写的指令（包括机器语言指令或高级语言指令）集合，属于高级语言级别，对用户的透明性好，运行时存放在计算机的主存中。

指令是指挥计算机执行某种功能的命令，是构成程序的基本单位，由操作码和地址字段构成；而微指令则用于微程序控制器中产生指令执行过程中所需要的微命令，是构成微程序的基本单位，由操作控制字段、判别测试字段和下地址字段等组成。

8)微命令有哪几种编码方法?它们是如何实现的?

答：微指令的微命令有三种编码方法，分别是直接表示方法、字段直接译码法和混合控制法。

直接表示法的基本思想是：将微指令操作控制字段的每个二进制位定义为一个微命令，用“1”或“0"表示相应的微命令的“有”或“无”。

字段直接译码法的基本思想是：将微指令格式中的操作控制字段分成若干组，每组中包含若干个互斥性微命令，将相容性的微命令安排在不同组。

混合控制法：将直接表示法与字段直接译码法混合使用，以便在微指令字长、并行性及执行速度和灵活性等方面进行折衷，发挥它们的共同优点。

9)简述微程序控制器和硬布线控制器的设计方法?

答： **微程序控制器设计方法：**

1)分析指令执行的数据通路，列出每条指令在所有寻址方式下的执行操作流程和每一步所需要的控制信号；

2)对指令的操作流程进行细化，将每条指令的每个微操作分配到具体的机器周期的各个时间节拍信号上；

(3)设计微指令格式、微命令编码方法和程序组织方式；

(4)编制每条指令的微程序；并按照所设计的微程序组织方式存放到控存中；

(5)对微命令进行同步控制，并送数据通路的相关控制点。

硬布线控制器设计方法：

1)分析指令执行的数据通路，列出每条指令在所有寻址方式下的执行操作流程和每一步所需要的控制信号；

2)对指令的操作流程进行细化，将每条指令的每个微操作分配到具体的机器周期的各个时间节拍信号上，即对操作控制信号进行同步控制。

3)对每一个控制信号进行逻辑综合，得到每个控制信号的逻辑表达式。

4)最后采用逻辑门或 PLA 或 ROM 实现逻辑表达式的功能，各控制信号送数据通路的相关控制点。

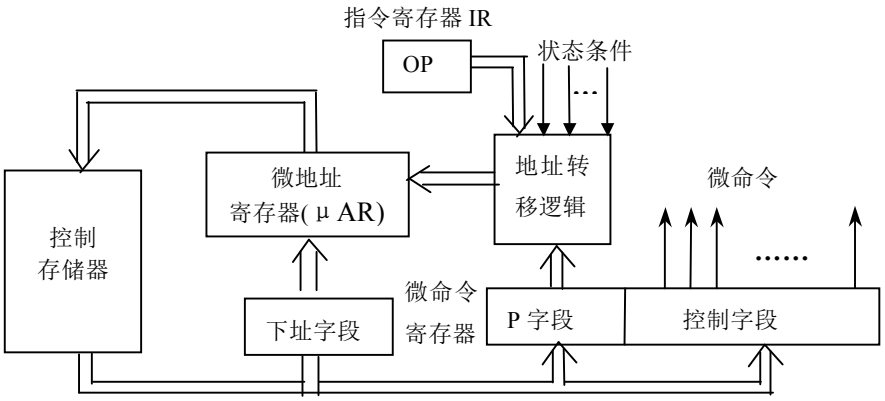
6.4 已知某机采用微程序控制方式，控存容量为 128*32 位。微程序可在整个控存中实现转移，控制微程序转移的条件共 3 个，微指令采用水平型格式，后继微指令地址采用断定方式。请问；

(1) 微指令的三个字段分别应为多少位?

(2) 画出对应这种微指令格式的微程序控制器逻辑框图。

解答：（1）分别是：控制字段 23 位；测试字段 2 位；下址字段 7 位；

（2）



6.5 某微程序包含 5 条微指令,每条微指令发出的操作控制信号如表 6.25 所示，试对这些微指令进行编码，要求微指令的控制字段最短且能保持微指令应有的并行性。

表 6.25 微指令及其对应的微操作控制信号

微指令	微操作控制信号
μI_1	*a,c,e,g*
μI_2	*a,d,f,h,j*
μI_3	*a,d,e*
μI_4	*a,b,i*
μI_5	*a,d,f,j*

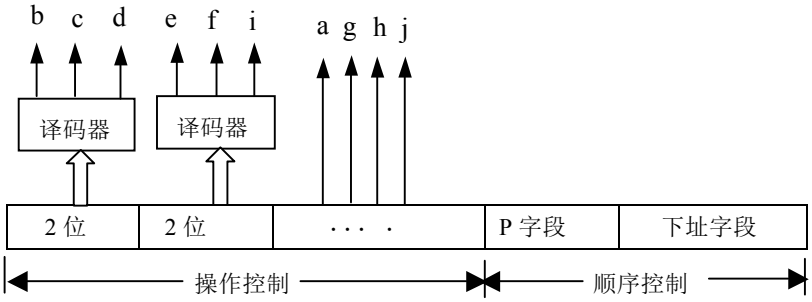
答：由题可得下表：

微 指令	微操作控制信号									
	a	b	c	d	e	f	g	h	i	j
μI_1	√		√		√		√			

μI_2	√			√		√		√		√
μI_3	√			√	√					
μI_4	√	√							√	
μI_5	√			√		√				√

由微命令的编码方法可知，采用字段直接译码法可以有效缩短微指令的长度，为此，先找出互斥性的微命令。

从表中可以发现两个互斥组(b,c,d) , (e,f,i)(或其它可能存在的互斥组)，这两个互斥组采用字段直接译码法，其余的 a,g,h,j 等四个微命令采用直接表示方。



6.6 某 CPU 的结构如图 6.37 所示，其中 AC 为累加器，条件状态寄存器保存指令执行过程中的状态。a, b, c, d 为四个寄存器。图中箭头表示信息传送的方向。完成下列个题：

(1)根据 CPU 的功能和结构标明图中四个寄存器的名称；

(2)简述指令 LDA X 的数据通路，其中 X 为主存地址，指令的功能是将主存 X 单元的内容送入 AC 中。

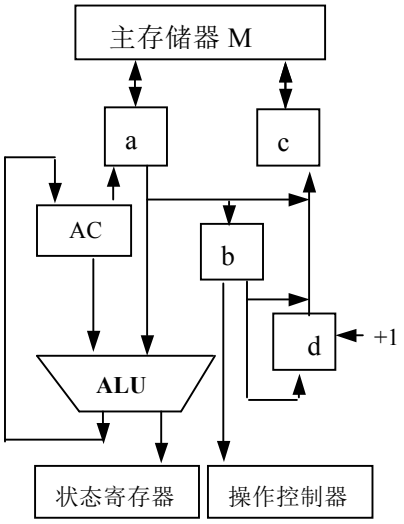


图 6.37　某 CPU 的结构框图

6.6 答：（1）a：数据缓冲寄存器 MDR；

b：指令寄存器 IR；

c：地址寄存器 MAR；

d：程序计数器 PC

（2）首先，取指阶段有条数据通路，1.取指令送到指令寄存器：PC→MAR→主存→MDR→IR；2.修改 PC 为下一条指令做准备：PC→PC+1

其次，执行阶段用到的数据通路是，从主存中取数据并送到寄存器 AC 中：IR(形式地址部分)→MAR→MEM→MDR→AC

6.7 对图 6.4 所示的单总线 CPU，如加法指令中的第二个地址码有寄存器寻址、寄存器间接寻址、存储器间接寻址三种寻址方式，并在指令中用代码表示寻址方式，对应的指令及功能如下：

a) ADD R0, R1 ; R[0]← (R[0]) + (R[1]) ,

即把寄存器 R0 的内容和 R1 的内容相加,结果送送 R0 保存

b)ADD R0, (R1) : R[0]← (R[0]) +(M[R[1]])

即把寄存器 R0 的内容和 R1 的内容所指主存单元的值相加,结果送 R0 保存

c) ADD R0, (addr) : R[0]← (R[0]) +(M[addr])

即把寄存器 R0 的内容和主存单元 addr 的值相加,结果送 R0 保存

分别设计上述三条指令的指令周期流程图，并列出每一步的控制信号。

6.7 答：三种寻址方式下取指周期的操作及其控制信号如表 6.3 的上半部分所示.下面只给出每种情况下指令执行周期的操作及其控制信号.

a)

操作	周期	功能说明	控制信号
X ← (R[0]) Z ← ALU R[0] ← (Z)	执行	寄存器 R0 内容送暂存器 X ALU 执行加法操作，将结果送暂存器 Z 暂存器 Z 的内容送寄存器 R0 中	R0out=Xin=1 R1out=1=ADD=1 Zout = R0in=1

b）

操作	周期	功能说明	控制信号
MAR← (R[1]) MDR←(MEM)	执行	将寄存器 R1 中的内容作为地址送 MAR 将主存对应单元的数据送 MDR	R1out=ARin=1 READ=DREin=1
X ← (R[0]) Z ← ALU R[0] ← (Z)	执行	寄存器 R0 内容送暂存器 X ALU 执行加法操作，将结果送暂存器 Z 暂存器 Z 的内容送寄存器 R0 中	R0out=Xin=1 DRIout=ADD=1 Zout = R0in=1

c）

操作	周期	功能说明	控制信号
MAR← IR(addr) MDR←(MEM)	执行	将立即数 addr 中的内容作为地址送 MAR 将主存对应单元的数据送 MDR	IRout=ARin=1 READ=DREin=1
X ← (R[0]) Z ← ALU R[0] ← (Z)	执行	寄存器 R0 内容送暂存器 X ALU 执行加法操作，将结果送暂存器 Z 暂存器 Z 的内容送寄存器 R0 中	R0out=Xin=1 DRIout=ADD=1 Zout = R0in=1

第八章

8.1 解释下列名词

总线　系统总线　内存总线　I/O 总线　三态门　总线事务　总线复用　总线带宽　突发传输　总线连接　总线仲裁　串行传送　并行传送　数据传输模式　总线标准　PCI 总线　AGP 总线　总线事务分离　波特率　桥　全双工　半双工　主设备　从设备　广播与广集　同步通信　异步通信

参考答案：

1.总线：部件之间信息传输的公共通路，通过总线可实现部件之间数据信息和控制信息的传输；

2.系统总线：连接 CPU、主存、I/O 模块等主要部件之间的信息传输线；

3.内存总线：连接处理器和存储器的总线称为内存总线；

4.I/O 总线：主要用于计算机和 I/O 设备之间的通信连接线；

5.三态门：指输出具有高电平、低电平和高阻状态的逻辑门；

6.总线事务：把总线上一对设备之间的一次信息交换过程称为一个“总线事物”

7.总线复用：线指一组传输线具有多种用途,分时传送不同类型的信息。最常见的是地址总线和数据总线复用, 即将地址总线和数据总线共用一组物理线路，某一时刻该总线传输地址信号，另一时刻传输数据信号。

8.总线带宽：在总线上每秒能传输的最大字节量，一般单位为 B/S

9.突发传输：是一种连续的、成批的数据传送方式。一般在数据开始传送前先给出批量传输数据的起始地址，然后连续传输多个数据，除首地址外，后续数据的地址依次通过前一个数据的地址加 1 的方式获得。由于减少了传送地址的时间，所以突发传输下数据的传输率大大提高。

10.总线连接：研究构成计算机的各大组成部件如何与总线相连并构成一个有机的整体，这与系统总线的结构紧密相关

11.总线仲裁：也称总线的控制。因为总线为多个部件共享，为防止同时有多个部件同时使用总线导致的数据冲突，需要有一个总线控制机构来解决总线使用权的仲裁，以某种方式选择其中一个设备作为主设备。

12.串行传送：指一个单位信息按从低位到高的顺序逐位以脉冲方式传送，它们共享一条传输线，一次只能传送一位。

13. 波特率:串行传送方式下,每秒钟传送的二进制位数称为波特率。

14.并行传送：指一个信息的所有位同时传送，每位都有各自的传输线，互不干扰，一次传送多位信息。

15.数据传输模式：当前的总线标准大都能支持以下四类数据传送模式，读、写操作，块传送操作，写后读、读修改写操作，广播、广集操作

16.总线标准：为实现系统与各模块、模块与模块之间的一个互连而指定的总线规范。

17.PCI 总线：PCI 总线是一个与处理器无关的高速外围总线，从结构上看，PCI 是在 CPU 和原来的系统总线之间插入的一级总线，具体由一个桥接电路实现对这一层的管理，并实现上下总线之间的接口以协调数据的传送。它采用同步通信方式和集中式控制策略，并具有自动配置能力。

18. AGP 总线：AGP（Accelerated Graphics Port)局部总线规范是 Intel 于 1996 年 7 月，PCI2.1 版规范基础之上扩充修改而成，专门为显卡量身打造的一种总线标准，以满足随着 3D 游戏的迅速普及,显卡的数据吞吐量越来越大的需求。

19. 总线事务分离：一次总线事务一般包括地址阶段和数据阶段。在地址阶段，获得总线使

用权的设备向被寻址的从设备发出地址信息，从设备确认该地址，并向主设备发回应答信号。在数据阶段，主从设备之间传输数据信息。
20.波特率：在信道上每秒钟传送的码元(波形)个数。
21.桥：是一个总线转换部件，通过它把一条总线的地址空间映射到另一条总线的地址空间上，从而实现不同总线之间的互联互通。
22.全双工：使数据在两个方向上同时进行传送操作。指在发送数据的同时也能够接收数据，两者同步进行。
23.半双工：息在两点之间能够在两个方向上进行发送，但双向传送需要分时进行。
24.主设备：获得总线使用权的设备。
25.从设备：数据传输过程中，被主设备寻址的设备或只能接受从其他设备发出信息的设备。
26.广播与广集：一般而言，数据传送只在一个主方和一个从方之间进行。但有的总线允许一个主方对多个从方进行写操作，这种操作称为广播。与广播相反的操作称为广集，它将选定的多个从方数据在总线上完成响应的操作（比如或操作或与操作）。
27.同步通信：通信双方在统一的时钟控制下进行信息的传输
28. 异步通信：又称应答通信，是指通信联络的控制信号采用异步方式的一种通信方法，即总线上的部件通过总线传送信息时，源部件不只是单向发送信息，它在发出一个信息后，要等待目的部件发回确认信号，再发送下一个信息。

8.2 简要回答下列问题
1)计算机系统为什么采用总线结构？
2)比较单总线、双总线、三总线结构的性能特点。
3)总线的信息传送方式有哪几种？各有什么特点？
4)集中式总线控制方下，确定总线使用权优先级的方法有哪几种？各有什么特点？
5)影响总线性能的因素有哪些?
6)什么是突发传输模式?采用突发传输模式有什么优点?
解：

1. 计算机系统为什么采用总线结构

计算机所有功能的实现过程就是各种信息在计算机内各大功能部件之间进行交换的过程，因此，必须在部件之间构筑信息传输的公共通路，即总线。计算机系统通过总线将 CPU、主存储器及输入输出设备连接起来，并在这个通路上传送地址信息、数据信息及控制信息。

2. 比较单总线、双总线、三总线结构的性能特点。

在单总线结构的计算机中只有一条系统总线，因此构成计算机系统的各部件如 CPU、主存储器及输入输出设备等，都只能连接在这一条总线上并构成一个完整的计算机系统。单总线结构具有如下优点：

总线结构简单，使用灵活，扩充容易。在总线上增加新的外设不涉及到总线的扩展和已经连接到总线上其它设备的变化。

单总线结构的不足主要表现在：

1)主存与外部设备采用统一编址,减少了主存的地址空间；

2)高速设备和低速设备连接在同一组总线上，高速设备的高速特性得不到发挥；

3)总线只能被分时使用，通信速度慢；

4)任何两部件之间的信息传递都共享受一组总线，系统总线负载重,系统性能低。

双总线在单总线结构的基础上，通过在 CPU 和主存储器之间增加一组高速的存储总线(也称主存总线)而得到。这种类型的双总线结构具有如下特点：

(1)仍然保持了单总线系统扩展容易的优点；

(2)存储总线的使用，大大降低了系统总线的负载；

在双总线的基础上，将主存从系统总线上分离出来，并将原来的系统总线分离成主存总线和 I/O 总线的**三总线结构**。

3. 总线的信息传送方式有哪几种？各有什么特点？

串行传送：特点是只需一条传输线，成本低。当远距离传输时，如几百米甚至几公里以上，采用这种方式比较经济。但是，串行传送速度慢。

并行传送：优点是传送速度快。然而，这种方式要求线数多，成本高。因此，在距离不远时可以采用并行传输。

并串行传送：将被传送信息分成若干组，组内采用并行传送，组间采用串行传送。它是对传送速度与传输线数进行折衷的一种传送方式。

分时传送：一是采用总线复用，指的是在某个传输线上既传送地址信息，又传送数据信息，其目的是为了减少线数，为此必须划分时间片，以便在不同的时间间隔中完成传送地址和传送数据的任务。二是指共享总线的部件分时使用总线。总线资源是系统的公共资源，挂在总线上的部件可以有很多，但在一个特定时间片内，总线通常只为一个源件和一个目的部件提供服务，所以多个部件要求使用总线时，只能由总线控制器按时间片分时提供服务。

4. 集中式总线控制方下，确定总线使用权优先级的方法有哪几种？各有什么特点？

串行连接方式：链式查询方式，优点是结构简单、扩充容易。缺点主要表现在优先级固定，对单点故障敏感，当优先级高的部件频繁请求使用总线时，会使优先级较低的部件长期不能使用总线，采用串行查询方式,响应速度慢，所以串行链接方式适合于小系统中使用。

计数器定时查询方式：优点是优先级改变灵活，单点故障不再影响其他部件的正常工作，不足是系统扩展较复杂，计数地址线增加后涉及到与所有部件连接的改变，响应速度仍然较慢。

独立请求方式：特点是响应时间快，不必逐个设备地查询。此外，独立请求方式对优先次序的控制相当灵活，既可采用优先级固定法，也可通过程序改变优先次序，还可通过屏蔽(即禁止)某个请求，以禁止相应的部件使用总线。缺点是增加线数和控制器的复杂度。

5. 影响总线性能的因素有哪些?

总线宽度，波特率，比特率，总线传输周期，总线带宽

6. 什么是突发传输模式?采用突发传输模式有什么优点?

由一个地址阶段和多个数据阶段组成。其中地址阶段发送的是连续数据单元的首地址，在数据阶段传送多个连续单元的数据，因此，突发传送模式也称为成组传送模式，在该传送模式中，每个总线周期仍传送一个字长的信息，但不释放总线，直到这批信息送完后，再释放总线

优点：传输相同的数据量，采用突发传输方式，可减少地址的传输次数和总线的申请次数。

8.3 假设一个同步总线的时钟频率为 100MHZ,总线带宽为 32 位,每个时钟周期传输一个字长的数据,该总线的最大数据传输率为多少?若要将总线带宽提高一倍,有哪几种可行方案?

解：(1) 时钟频率为 100MHZ,则一个时钟周期的时间 T = 1/100MHZ = 0.01μs

1 个时钟周期的时间为 0.01μs

数据传输率 = 4B / 0.01μs = 400MB/s

（2）有下列几种方法可以将总线的带宽提高一倍：

•将总线数据线增加到 64 位；
•将总线的时钟频率增至 200MHZ；
•每个时钟周期传输 2 个数据。

8.4 采用异步通信方式传送 ASCII 码时，若数据位 8 位，校验位 1 位，停止位 1 位，计算当波特率为 4800 时，字符传送的速率是多少？每个数据位的时间长度是多少？数据位的传送速率是多少？

解：（1）字符传送速率为：4800/10=480 字符/秒；
（2）每个数据位的时间为 1/（8*480）=0.26ms
（3）8*480=3840 位/秒

8.5 有 4 个设备 A、B、C、D 的响应优先级从高到低的次序为 D>B>A>C，画出串行链式排队电路。

解：

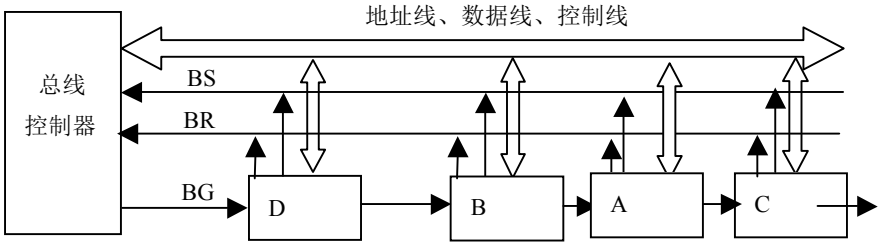
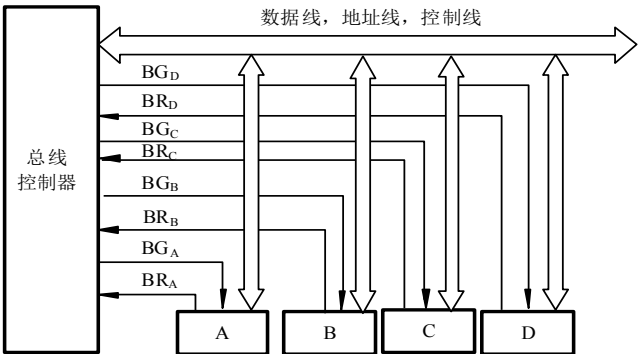


图 8.10　串行链接方式示意图

8.6 有 4 个设备 A、B、C、D 的响应优先权为 A>B>C>D，试画出独立请求方式的排队电路。

解：



其中优先权的设定可通过编程进行修改。

8.7 假定有一个具有以下性能的系统 1)存储器和总线系统支持大小为 4-16 个 32 位字的数据块访问;2)总线的时钟频率为 200MHZ，总线宽度为 64 位，每 64 位数据的传输需要一个时钟周期，向存储器发送一个地址需要一个时钟周期；每个总线操作之间需要 2 个总线周期(设一次存储之前总线总是处于空闲状态);3)对最初的 4 个字的访问时间为 200ns,随后的 4 个字能在 20ns 的时内被读取,假定总线传输数据的操作可以与读下 4 个字的操作重叠进行。读操作中，分别用 4 个字的数据块和 16 个字的数据块传输 256 个数据，计算机两种情况下总线传输的带宽和每秒中总线事务的次数。（说明：一个总线传输操作包含一个地址和紧随其后的数据）

解：用 4 个字的数据块传输 256 个数据，需要传送 64 次数据，根据题目假设条件，传输 256 个数据需要的总时间为：

t= 200ns + 20ns * (64-1) = 1460ns

则传输的带宽为：256*4B/1460ns = 701MB/s

总线事务次数为：64*（1+2）=192 次。

用 16 个字的数据块传输 256 个数据：t=3*16*T+200ns*16+48*20ns

传输带宽为：256*4B/（3*16*T+200ns*16+48*20ns）=233MB/s

总线事务次数为：16*（1+8）=144 次。

第九章

9.1 解释下列名词

接口　中断　中断处理优先级　中断屏蔽　多重中断　中断向量　中断响应优先级　中断隐指令　程序中断 I/O　程序查询 I/O　DMA　周期挪用　通道　选择型通道　通道指令　输入设备　输出设备　显示分辨率　点距　行反转扫描法

解：(1)接口：接口是两个不同部件或系统之间的连接部分，可以是两个硬设备(可以都是计算机，也可以都是外部设备)之间的连接，也可以是软件系统中两个独立程序块之间的连接。

(2)中断：计算机系统运行时，若系统外部、内部或现行程序本身出现某种非预期的事件，CPU 将暂时停下现行程序，转向为该事件服务，待事件处理完毕，再恢复执行原来被终止的程序，这个过程称为中断。

(3) 中断处理优先级:处理优先级是指 CPU 实际完成中断处理程序的先后次序。对单级中断而言,先被 CPU 响应的中断服务程序先完成；对多重中断而言，先被 CPU 响应的中断不一定先完成，这与中断屏蔽密切相关。

(4) 中断屏蔽: 为了便于利用程序控制中断处理的先后顺序,可通过程序有选择地封锁部分中断源发出的中断请求，而允许其余部分中断仍得到响应，这种方式称为中断屏蔽。

(5) 多重中断: 若在中断服务程序执行过程中，如果允许 CPU 响应其它中断请求，则这种中断称为多重中断，也称中断嵌套。

(6) 中断向量: 通常将中断服务程序的入口地址和程序状态字(有的机器不包含此项)称为中断向量。

(7) 中断响应优先级: 响应优先级是指 CPU 对各设备中断请求进行响应的先后次序,它根据中断事件的重要性和迫切性而定。当几个设备同时有中断请求时，优先级高的先响应，优先级低的后响应。

(8) 中断隐指令: CPU 响应中断之后，经过某些操作，转去执行中断服务程序。这些操作是由硬件直接实现的,把它称为中断隐指令。中断隐指令并不是指令系统中的一条真正的指令，它没有操作码，所以中断隐指令是一种不允许、也不可能为用户使用的特殊指令。

(9) 程序中断 I/O: 当主机启动外设后，无需等待查询，而是继续执行原来的程序，外设在做好输入输出准备时，向主机发出中断请求，主机接到请求后就暂时中止原来执行的程序，转去执行中断服务程序对外部请求进行处理，在中断处理完毕后返回原来的程序继续执行。

(10)程序查询 I/O: 程序查询方式是一种程序直接控制方式,这是主机与外设间进行信息交换的最简单的方式,输入和输出完全是通过 CPU 执行程序来完成的。一旦某一外设被选中并启动后，主机将查询这个外设的某些状态位，看其是否准备就绪？若外设未准备就绪，主机将再次查询；若外设已准备就绪，则执行一次 I/O 操作。

(11)DMA: 直接存储器存取控制方式 DMA 方式下外设与主存之间传送数据时，CPU 仍可执行主程序.

(12)周期挪用:周期挪用是指利用 CPU 不访问存储器的那些周期来实现 DMA 操作，此时 DMAC 可以使用总线而不用通知 CPU 也不会妨碍 CPU 的工作。

(13)通道: 通道方式是 DMA 方式的发展，在通道方式下，数据的传送方向、存取数据的内存起始地址及传送的数据块长度等都由独立于 CPU 的通道来进行控制，因此，通道方式可进一步减少 CPU 的干预。

(14)选择型通道: 对于这种高速传输，通道难以同时对多个这样的设备进行操作，只能一次对一个设备进行操作,这种通道称为选择通道。

(15)通道指令: 通道程序是由一系列通道指令组成的,通道指令一般包含被交换数据在内存中应占据的位置、传送方向、数据块长度及被控制的 I/O 设备的地址信息、特征信息（例如

是磁带设备还是磁盘设备）等.
(16)输入设备:向计算机输入数据和信息的设备.
(17)输出设备:是人与计算机交互的一种部件，用于数据的输出。
(18)显示分辨率:显示分辨率是显示器在显示图像时的分辨率，分辨率是用点来衡量的，显示器上这个“点”就是指像素(pixel)。
(19)点距: 点距指屏幕上相邻两个同色像素单元之间的距离，即两个红色（或绿、蓝）像素单元之间的距离。
(20)行反转扫描法: 先对所有行线送"1"，所有列线送“0”，读键盘行扫描值；然后反过先对所有行线送"0"，然后对所有列线送“1”，并读键盘列扫描值。
9.2 简要回答下列问题
1)什么是接口?它有哪些功能?
2)主机与外部设备之间如何连接?
3)主机与外部设备信息交换的控制方式有哪些?各有什么特点?
4)什么是程序程序查询 I/O 方式,简要说明其工作原理.
5)比较单级中断和多重中断处理流程的异同点.
6)中断隐指令完成什么功能?
7)为什么在保护现场和恢复现场的过程中,CPU 必须关中断?
8)CPU 响应中断的条件有哪些?
9)什么是中断向量,简要分析中断向量方式下形成中断向量的基本方法.
10)为什么采用 DMA 方式能提高成组数据传送的速度?
11)什么是中断优先级?它具有哪两层含义?划分优先等级的原则是什么?
12)计算机中断系统中使用屏蔽技术有什么好处?
13)计算机中断响应后，如何调出中断服务程序?
14)DMA 方式传送数据前，主机应向 DMA 接口输送哪些参数?
15)比较中断 I/O 和 DMA 的一统点。
16)比较 DMA 与通道的异同点。
17)中断系统中设计中断允许和中断屏蔽的作用分别是什么?两者是否可以合二为一?
解：(1)接口是两个不同部件或系统之间的连接部分，可以是两个硬设备(可以都是计算机，也可以都是外部设备)之间的连接，也可以是软件系统中两个独立程序块之间的连接。
具有的功能：1)寻址功能。2)数据输入/输出功能。3)匹配主机与外设的速度差距。4)实现数据格式转换或逻辑电平转换。5)传送主机命令。6)反映设备的工作状态。
(2)主机通过接口连接 I/O 设备，接口实现主机与外设的连接和信息的交换。
(3) 主机与外部设备信息交换的控制方式有:程序查询控制方式、程序中断控制方式、直接存储器存取控制方式(DMA)、通道方式、外围处理机方式。
特点：程序查询控制方式接口设计简单，但是 CPU 与外设只能串行工作，由于 CPU 的速度比外设的速度要高得多，所以在信息传送过程中，CPU 的大量时间是花费在查询和等待上，从而使系统效率大大降低。
程序中断控制方式：允许外部设备用“中断”信号中止 CPU 正在执行的程序。具体他说，当接口电路需要与 CPU 进行数据交换（输入、输出等）时，便由接口电路向 CPU 发出一个中断请求信号，CPU 响应这一中断请求，并调用中断服务程序完成一个或多个字节的信息交换。这种方式不需要接口软件主动查询，而是由接口电路主动通知 CPU，即在设备准备数据阶段，CPU 与外设能并行工作，使得接口软件的效率比较高。
直接存储器存取控制方式：数据传输的基本单位是数据块；所传输的数据是从设备直接送入内存的,或者相反；整块数据的传送是在控制器的控制下完成的；

通道方式：CPU 发出启动通道的指令，通道就开始工作。I/O 通道控制 I/O 控制器工作，I/O 控制器又控制 I/O 设备。这样，一个通道可以连接多个 I/O 控制器，而一个 I/O 控制器又可以连接若干台同类型的外部设备。
外围处理机方式：通常用于大、中型计算机系统中。由于 PPU 基本上独立于主机工作，其结构更接近一般处理机，甚至就是一般的通用微小型计算机。它可以完成 IOP 的功能，还可以完成码制变换、格式处理、数据块检错、纠错等操作。

9.4 设某机有 5 级中断；L0，L1，L2，L3，L4，其中断响应优先次序为：L0 最高，L1 次之，L4 最低。现在要求将中断处理次序改为 L1->L3->L0->l4->L2，试问：
(1)表 9.4 所示的中断屏蔽字该如何设置(“0”表示允许中断，“1”表示中断屏蔽)?
(2)若这 5 级中断同时都发出中断请求，按更改后的次序画出进入各级中断处理程序的过程示意图。

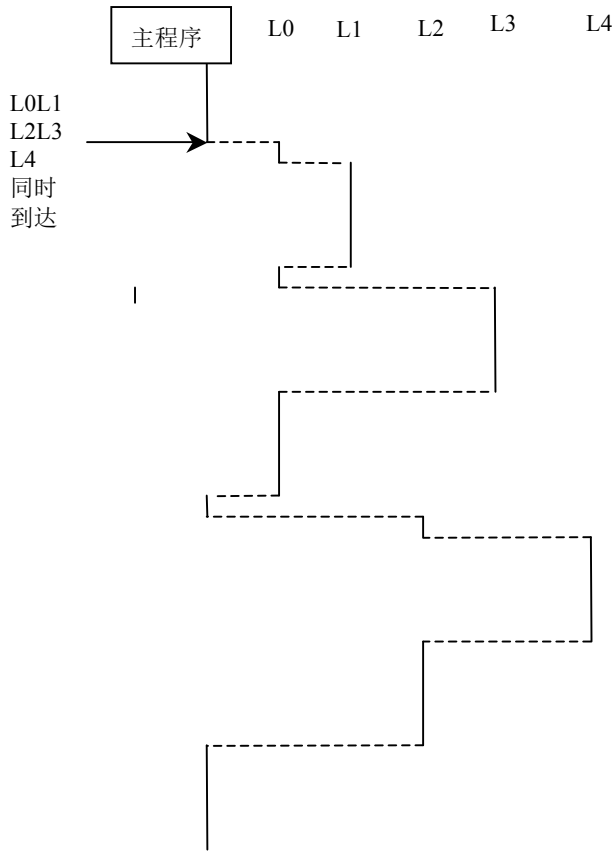
9.4 更新后的中断屏蔽表

中断处理程序	中断处理级屏蔽位				
	L0 级	L1 级	L2 级	L3 级	L4 级
L0 中断处理程序					
L1 中断处理程序					
L2 中断处理程序					
L3 中断处理程序					
L4 中断处理程序					

解：(1)

中断处理程序	中断处理级屏蔽位				
	L0 级	L1 级	L2 级	L3 级	L4 级
L0 中断处理程序	1	0	1	0	1
L1 中断处理程序	1	1	1	1	1
L2 中断处理程序	0	0	1	0	0
L3 中断处理程序	1	0	1	1	1
L4 中断处理程序	0	0	1	0	1

(2)



9.6 某计算机的 CPU 主频为 500MHZ,与之连接的外设的最大数据传输率为 20KBps,外设接口中有一个 16 位的数据缓冲器,相应的中断服务执行时间为 500 个时钟周期,通过计算分析该设备是否可采用中断 I/O 方式?若该设备的最大数据传输率为 2MBps,该设备是否可采用中断 I/O 方式?

解：由题意可知：当外设的最大数据传输率为 20KBps，缓冲区为 2B。则

每秒钟产生的中断数为：20KB/2B=10000 次。

每次的执行为 500 个周期。则中断占 CPU 时间的比率为：

$500*10000/(500*10^6)=1\%$,对 CPU 的影响不大，可以采用中断方式。

当最大数率为 2MBps 时，

每秒钟产生的中断数为：$2MB/2B=10^6$

则中断占 CPU 的时间比率为：

$(500*10^6)/(500*10^6)=100\%$

故不能采用中断方式。