第 3 章　课后习题解析及答案

3.1　第 1 章习题解答

1．简要解释下列名词术语

【答】

数字计算机：一种能存储程序，能自动连续地对各种数字化信息进行处理的快速工具。

硬件：是指组成计算机系统的设备实体，如 CPU、存储器、I/O 设备等。

软件：泛指各类程序、文档等。

CPU：即中央处理器，是由运算器和控制器组成的计算机硬件系统的核心部件。

主存储器：位于主机内部，用来存放 CPU 需要使用的程序和数据的部件。

外存储器：位于主机外部，用来存放大量的需要联机保存、但 CPU 暂不使用的程序和数据的部件。

信息的数字化表示：注意，信息的数字化表示包含了两层含义，即：用数字代码表示各种信息，用数字信号（电平、脉冲）表示数字代码。

存储程序工作方式：事先编制程序，事先存储程序，自动、连续地执行程序。

数据通路宽度：是指数据总线一次能并行传送的数据位数。

数据传输率：是指数据总线每秒钟传送的数据量。

字长：一般指参加一次定点运算的操作数的位数。

CPU **外频**：CPU 的外部频率，也是计算机系统的基准频率，是部件工作的时钟信号基础，它由主板上的震荡器产生，此频率一般不会超过 200MHz。

CPU **主频**：CPU 内核的工作频率，也称为 CPU 时钟频率。

CPU **功耗**：CPU 消耗的功率，包括静态功耗和动态功耗。静态功耗主要由电路泄露造成，动态功耗等于 $C \times U^2 \times f$，其中 C 表示等效电容、U 是工作电压、f 是工作频率。

2．数字计算机的主要特点是什么？

【答】

应该从信息的表示方法和计算机的工作方式来说明它的主要特点。

有 5 点：能自动连续地执行程序、运算速度快、运算精度高、存储能力强、通用性好。

3．计算机有哪些主要性能指标？

【答】

主要从计算机的运算能力、存储能力、传送能力、处理能力等几方面考虑。

主要性能指标包括基本字长、运算速度、存储容量（主存容量和外存容量）、数据传输率、外设配置和软件配置等。

4．冯•诺依曼思想包含哪些要点？

【答】

冯•诺依曼思想奠定了现代计算机的基本结构思想，它很好地解决了信息如何表示才能被计算机识别和计算机采用何种工作方式才能自动地对信息进行处理等基本问题。它包含三个要点：

（1）采用二进制代码表示信息，以便计算机识别；

（2）采用存储程序工作方式，才能使计算机自动地对信息进行处理；

（3）由存储器、运算器、控制器、输入/输出设备等功能部件组成计算机硬件系统。

5．信息的数字化表示包含哪两层含义？

【答】

信息的数字化表示不仅要考虑在计算机中如何表示各种原始信息，还要考虑在物理机制上怎样实现。所以，第一层含义：用数字代码表示各种信息；第二层含义：用数字信号表示数字代码。

6．用数字信号表示代码有什么优点？

【答】

每位数字信号只有两种可能的状态，因而可从物理实现、可靠性、数值范围与精度、信息类型、信息处理等方面说明。有以下 5 点：

（1）在物理上容易实现信息的表示与存储；

（2）抗干扰能力强，可靠性高；

（3）数值的表示范围大，表示精度高；

（4）能表示极其广泛的信息类型；

（5）能用数字逻辑技术处理信息。

7．编译方式和解释方式对源程序的处理有什么区别？

【答】

在编译方式中，计算机执行编译程序，将源程序全部转换为目标程序，然后由计算机单独执行目标程序，即先翻译，后执行。

在解释方式中，计算机执行解释程序，将源程序逐段转换为对应的目标程序段，每转换一段便执行该段目标程序，直到整个源程序被解释执行完，即边翻译，边执行。

8．为什么要对计算机系统进行层次划分？

【答】

计算机系统是由硬、软件组成的复杂系统，进行层次划分，有助于根据不同需要，从不同层次去分析、构造、调试、维护和扩充计算机系统。

9．软件系统一般包含哪些部分？试列出你所熟悉的几种系统软件。

【答】

前面几道题都是涉及基本概念的题，从这道题开始，则是与实际应用有关。

软件系统一般包含系统软件和应用软件两部分。所熟悉的系统软件可根据实际情况列出，如操作系统（Windows、Linux、……等等），C 编译程序，数据库管理系统（SQL Server、Sybase、……等等）。

10．以你所熟悉的一种计算机系统为例，列举出该系统所用的 CPU 型号，时钟频率，字长，主存容量，外存容量，所连 I/O 设备的名称等。

【答】

例如使用奔腾芯片的计算机系统，CPU 为 Pentium-200，时钟频率为 200MHz，字长 32 位，主存容量为 256MB，硬盘容量为 40GB，I/O 设备包括键盘、鼠标、显示器、喷墨打印机等。

11．什么是控制流驱动？什么是数据流驱动？

【答】

传统的诺依曼机采用控制流（指令流）驱动方式：按指令序列依次读取指令，根据指令所包含的控制信息对数据进行处理，在程序执行过程中，始终由指令流驱动计算机工作。

数据流驱动方式是对传统诺依曼机工作方式的根本改变：只要数据准备好，有关指令就可并行执行，如数据流计算机。

12．你曾在计算机的机器指令级、操作系统级、汇编语言级或高级语言级上做过工作或练习？或调用过该级的功能？举出所做的工作或所调用的功能名。

【答】

按实际情况回答，比如用汇编语言或高级语言编写过程序等等。

13．试分析微型机、小型和大型计算机的特点。

【答】

微型机采用了集成度很高的电子元件和总线结构，一般采用单 CPU 结构，通用性强，轻便、小巧、价格低，操作使用方便，普及最广，适用于个人电脑；

小型机，软硬件规模大于微机，结构复杂，一般采用多 CPU 架构，具有高可靠性、可用性和高服务性等特征，一般用作中小型高性能服务器；

大型机，一般用作大型的高性能服务器，具有专用的处理器指令集和专用应用软件，运算速度快，存储容量大，通用性强，功能完备，支持大量用户同时使用，具有强大的数据处理能力，价格昂贵。

14．有三款处理器 CPU1 、CPU2 和 CPU3，…

【答】

（1）根据 CPU 执行同一个程序所需的执行时间来判断 CPU 的综合性能，速度最快者

的综合性能更好。因为执行时间：$t= n\times CPI\times(1/f)$, 其中为指令数，则

t_{CPU3} = 2.5n/3GHz;

t_{CPU1} = 1.5n/2GHz;

t_{CPU2} = 1n/1.5GHz;

故 CPU2 的综合性能最好。

（2）时钟周期数 $k=t\times$频率，指令数 n=执行时间×频率÷CPI，所以这三个处理器执行的时钟周期数 k 和指令数 n 分别是：

$k_{CPU1}=10\times f_{CPU1}=20G$，$n_{CPU1}=10\times f_{CPU1}\div 1.5=13.3G$；

$k_{CPU2}=10\times f_{CPU2}=15G$，$n_{CPU2}=10\times f_{CPU2}\div 1.0=15G$；

$k_{CPU3}=10\times f_{CPU3}=30G$，$n_{CPU3}=10\times f_{CPU3}\div 2.5=12G$；

（3）执行时间：$t= n\times CPI\times(1/f)$，假设频率需要提高到 f'，所以：

$(1-30\%)\times t=n\times(1+20\%)\times CPI\times(1/f')$

$\rightarrow 0.7t=n\times 1.2CPI\times(1/f')$

$\rightarrow f'=n\times 1.2CPI\div 0.7t=12f/7=1.7143f$，故频率应提高 1.7143-1≈71.43%。

15．CPU 的浮点运算能力常用 MFLOPS，…

【答】

（1）FLOPS=(指令总数 n×浮点运算指令比例 k) ÷执行总时间 s，所以

$FLOPS_{P1}=(10^6\times 0.4)\div((10^6\times 0.5\times 0.75+10^6\times 0.4\times 1+10^6\times 0.1\times 1.5)\div f)\approx$**1297MFLOPS**;

$FLOPS_{P2}=(3\times 10^6\times 0.4)\div$**(**$(3\times 10^6\times 0.4\times 1.25+3\times 10^6\times 0.4\times 0.7+3\times 10^6\times 0.2\times 1.25)\div f$**)**

≈**405MFLOPS**;

（2）IPS=指令总数 n÷执行总时间 s，所以

$IPS_{P1}=10^6\div$**(**$(10^6\times 0.5\times 0.75+10^6\times 0.4\times 1+10^6\times 0.1\times 1.5)\div f$**)**≈**3243MIPS;**

$IPS_{P2}=(3\times 10^6)\div$**(**$(3\times 10^6\times 0.4\times 1.25+3\times 10^6\times 0.4\times 0.7+3\times 10^6\times 0.2\times 1.25)\div f$**)**≈**2913MIPS;**

（3）执行时间=指令总数×平均 CPI×(1/f)，故

$\mathbf{t}_{P1}=10^6\times(0.5\times 0.75+0.4\times 1+0.1\times 1.5)\div(3\times 10^9)\approx$**3.1×10^{-4}** 秒；

$\mathbf{t}_{P2}=3\times 10^6\times(0.4\times 1.25+0.4\times 0.7+0.2\times 1.25)\div(3\times 10^9)\approx$**1.03×10^{-3}** 秒；

3.2 第 2 章习题解答

1．简要解释下列名词术语

【答】

位权：在 r 进位制的数中，每个数位的数码所表示的数值等于该数码乘以一个与它所在数位相关的常数，这个常数称为该位的位权，简称权。

基数：在进位制中，各数位允许选用的数码个数，称为该进位制的基数，它等于该进位制各数位所允许的最大数码值加 1。

真值：在数的绝对值之前配上正（+；通常可省略）、负（－）符号表示的数称为该数的真值。例如用十进制数表示的真值：159，－132。用二进制数表示的真值；1011、－1011 等。

机器数：在计算机内部使用的，连同数的符号一起数码化的数称为机器数。

原码：让数码序列的最高位为符号位（0 表示正，1 表示负），其余部分为数（真值）的绝对值，这个数码序列称为该数的原码表示。

补码：它是机器数的一种表示方法，如果数为正，则正数的补码与原码形式相同；如果数为负，则负数的补码是将负数原码除符号位不变外，其余各位取反，末位再加 1。

定点数：在计算机中，小数点位置固定不变的数叫做定点数。

浮点数：小数点位置不固定，可随需要浮动的数称为浮点数。

规格化浮点数：是指浮点数的尾数部分用带符号定点小数表示，当 R=2 时，尾数用原码表示时其绝对值满足$1/2 \leq |M| < 1$（补码时是-1≤*M*<-1/2 或 1/2≤*M*<1）的浮点数称为规格化浮点数。

ASCII 码：是美国信息交换标准码的英文全名的简称，它与 ISO646、GB1988 标准兼容；它是 128 个常用字符的数码化表示，如字符 A 的 ASCII 码为 1000001B。

算术移位：算术移位是指对具有数值大小的数，将数码位置左、右移动，使其数值发生变化，但数的符号位不变的一类移位操作。

逻辑移位：逻辑移位是指将（二进制）代码序列视为纯逻辑意义上的代码组合，只是将数码位置循环移动或非循环移动，使数码位置发生变化，但没有正负性质，也没有数值大小变化的问题。

2．将二进制数$(1111010.00111101)_2$转换为八进制与十六进制数。

【答】

将二进制数转换为八进制数或十六进制数，可分别采用二-八缩写形式或二-十六缩写形式将二进制数分段对应转换即可。

对于二-八缩写形式，则是三位二进制数对应一位八进制数，而二-十六缩写形式，则是四位二进制数对应一位十六进制的数。分段时，以小数点为基准，向左每三位（或四位）一组分段，高位不够补 0；向右每三位（或四位）一组分段、低位不够补 0。

根据上面所述，本题答案如下：

$$(1111010.00111101)_2=(172.172)_8$$

$$(1111010.00111101)_2=(7A.3D)_{16}$$

3．将二进制数$(101010.01)_2$转换为十进制数与 BCD 码。

【答】

求二进制数中为 1 的各位的权值之和，则得到该二进制数的相应十进制数：

$$(101010.01)_2=(2^5+2^3+2^1+2^{-2})_{10}=(42.25)_{10}$$

将十进制数的各位均用四位二进制代码表示，即得到 BCD 码：

$$(101010.01)_2=(42.25)_{10}=(01000010.00100101)_{BCD}$$

注意：在 BCD 码中，最高位 0 和最低位的 0（包括与它们相邻的 0）不能省掉。本例最高位的 0 不能省掉。

4．将八进制数$(37.2)_8$转换为十进制数与 BCD 码

【答】

方法一：由按权相加法计算，即将八进制数展开成多项式求和的形式，求得的和为相应的十进制数，然后再写为 BCD 码。

$(37.2)_8=(3\times8^1+7\times8^0+2\times8^{-1})_{10}=(31.25)_{10}$

$(37.2)_8=(31.25)_{10}=(00110001.00100101)_{BCD}$

方法二：将八进制数转换为二进制数（即每位八进制数码用三位二进制数表示或称用二—八缩写形式书写，再将二进制数转换为十进制数与 BCD 码。

$(37.2)_8=(011111.010)_2=(2^4+2^3+2^2+2^1+2^0+2^{-2})_{10}=(31.25)_{10}=(00110001.00100101)_{BCD}$

5．将十六制数$(AC.E)_{16}$转换为十进制数与 BCD 码

【答】

方法一：将十六进制数展开成多项式求和的形式，求得的和即为相应的十进制数，然后再书写成 BCD 码。

$(AC.E)_{16}=10\times(16)^1+12\times(16)^0+14\times(16)^{-1}{}_{10}$

$=(160+12+0.875)_{10}=(172.875)_{10}$

$=(000101110010.100001110101)_{BCD}$

方法二：将十六进制数转换为二进制数（即每位十六进制数码用四位二进制数表示或称为每位用二—十六缩写形式书写），然后把二进制转换为十进制数，进而可书写为 BCD 码。

$(AC.E)_{16}=(10101100.1110)_2=(2^7+2^5+2^3+2^2+2^{-1}+2^{-2}+2^{-3})_{10}$

$=(128+32+8+4+0.5+0.25+0.125)_{10}$

$=(172.875)_{10}$

$=(000101110010.100001110101)_{BCD}$

6．将十进制数$(75.34)_{10}$转换为 8 位二进制数、八进制数及十六进制数。

【答】

方法一：先用减权定位法将$(75.34)_{10}$转换为二进制数，再转换为八进制数与十六进制数。

整数部分减权比较	X_i	位权
75－64=11	1 （高位）	64
11 < 32	0	32
11 < 16	0	16
11－8=3	1	8
3 < 4	0	4
3－2=1	1	2
1－1=0	1（低位）	1

整数部分转换结果：$(75)10=(1001011)_2$

小数部分减权比较	X_i	位权

0.34 < 0.5	0	0.5
0.34－0.25=0.9	1	0.25
⋮	⋮	⋮

所以，(75.34)10=(1001011.01…)2

题目要求转换为 8 位二进制数，故对上述结果采用 0 舍 1 入的舍入法，得到：

(75.34)10≈(1001011.1)2=(113.4)8=(4B.8)16

方法二：按除基取余法将整数 75 转换为二进制整数；按乘基取整法将$(0.35)_{10}$转换为二进制小数。两部分合起来即得到结果，然后再转为八进制数和十六进制数。

整数部分转换：

	余数	二进制数位
2 | 75		
2 | 37	1（低位）	X_0=1
2 | 18	1	X_1=1
2 | 9	0	X_2=0
2 | 4	1	X_3=1
2 | 2	0	X_4=0
2 | 1	0	X_5=0
0	1（高位）	X_6=1

转换结果为(75)10=(1001011)2。

小数部分转换：

小数部分乘以 2	取整数部分
0.34×2=0.68	0
0.68×2=1.36	1
⋮	⋮

所以， $(75.34)10=(1001011.01\ldots)_2$。

题目要求转换为 8 位二进制数，故对上述结果采用 0 舍 1 入的舍入法，得到：

$(75.34)=(1001011.01\ldots)2\approx(1001011.1)2=(113.4)8=(4B.8)_{16}$

7．将十进制数 $\frac{13}{128}$ 转换为二进制数

【答】

将以 2^n 为分母的分数转换为二进制数时，可用简单的方法进行。

$$\left(\frac{13}{128}\right)_{10}=\left(\frac{13}{2^7}\right)_{10}=(13\times 2^{-7})_{10}=(1101\times 0.0000001)_2=(0.0001101)_2$$

8．分别写出下列各二进制数的原码与补码，字长（含一位数符）为 8 位。

（1）+0　　（2）－0　　（3）0.1010

（4）－0.1010　　（5）1010　　（6）－1010

解：题目中给出的数均是真值，其位数都不到 7 位，因此对应的原码和补码均不到 8 位（包括一位符号位）。但题目中要求的字长是 8 位，因此应将对应的原码和补码写成 8 位的机器数，如表 3-2-1 所示。

表 3-2-1 真值、原码、补码的对应关系

机器数 / 真值	原　码	补　码
（1）+0	整数：00000000	整数：00000000
	小数：0.0000000	小数：0.0000000
（2）－0	整数：10000000	*
	小数：1.0000000	*
（3）+0.1010	0.1010000	0.1010000
（4）－0.1010	1.1010000	1.0110000
（5）1010	0 0001010	00001010
（6）－1010	1 0001010	11110110

注：（3）、（4）题的 0.1010、－0.1010 其原码补码均只有 5 位，不够 8 位，在它们尾数末位后补三个“0”，即成为字长 8 位的机器数。

（5）、（6）题的 1010、－1010，其原码、补码均只有 5 位，不够 8 位，在它们尾数前面（即符号位后）补上三位二进制数码：正数补 000；负数补码补 111，负数原码补 000。

9．若 $X_{补}$=0.1010，写出其 $X_{原}$与真值 X

【答】$X_{原}$=0.1010，真值 X=0.1010

10．若 $X_{补}$=1.1010，写出 $X_{原}$与真值 X

【答】$X_{原}$=1.0110，X=－0.0110

11．某定点小数字长 16 位，含一位符号，原码表示，分别写出下列典型值的二进制代码与十进制真值。

（1）非零最小正数　　（2）最大正数

（3）绝对值最小负数　　（4）绝大值最大负数

【答】四种典型值用表 3-2-2 表示。

表 3-2-2 定点小数原码典型值

	原码	二进制真值	十进制真值
(1)非零最小正数	0.000000000000001	0.000000000000001	2^{-15}
(2)最大正数	0.111111111111111	0.111111111111111	$1-2^{-15}$
(3)绝对值最小负数	1.000000000000001	－0.000000000000001	-2^{-15}
(4)绝对值最大负数	1.111111111111111	－0.111111111111111	$-(1-2^{-15})$

12．某定点小数字长 16 位，含一位符号，补码表示，分别写出下列典型值的二进制代码与十进制真值。

（1）非零最小正数　　（2）最大正数

（3）绝对值最小负数　　（4）绝大值最大负数

【答】

四种典型值用表 3-2-3 表示。

表 3-2-3 定点小数补码典型值

	补码	二进制真值	十进制真值
(1)非零最小正数	0.000000000000001	0.000000000000001	2^{-15}
(2)最大正数	0.111111111111111	0.111111111111111	$1-2^{-15}$
(3)绝对值最小负数	1.111111111111111	−0.000000000000001	-2^{-15}
(4)绝对值最大负数	1.000000000000000	−1	-1

13．某定点整数字长 16 位，含一位符号，补码表示，分别写出下列典型值的二进制代码与十进制真值。

（1）非零最小正数　　（2）最大正数

（3）绝对值最小负数　　（4）绝大值最大负数

【答】四种典型值用表 3-2-4 表示。

表 3-2-4 定点整数补码典型值

典型值	补码	二进制真值	十进制真值
(1)非零最小正数	0000000000000001	1	1
(2)最大正数	0111111111111111	111111111111111	$2^{15}-1$
(3)绝对值最小负数	1111111111111111	−1	-1
(4)绝对值最大负数	1000000000000000	−1000000000000000	-2^{15}

14．判别下列各补码表示的尾数是否属于规格化尾数。

【答】(1)是；（2）不是；（3）是；（4）不是；（5）是；（6）不是；

15．某浮点数字长 16 位，其中阶码 6 位，含一位阶符，补码表示，以 2 为底；尾数 10 位，含一位数符，补码表示，规格化。分别写出下列各典型值的二进制代码与十进制真值。

（1）非零最小正数　　（2）最大正数

（3）绝对值最小负数　　（4）绝大值最大负数

【答】四种典型值用表 3-2-5 表示。

表 3-2-5 浮点数典型值

典型值	浮点数代码	真值
(1)非零最小正数	100000，0.10…0	$(2^{-2^5})\times(2^{-1})$
(2)最大正数	011111，0.11…1	$(2^{2^5-1})\times(1-2^{-9})$
(3)绝对值最小负数	10000，1.01…1	$-(2^{-2^5})\times(2^{-1}+2^{-9})$
(4)绝对值最大负数	011111，1.00…0	$(2^{2^5-1})\times(-1)$

16．若采用图 2-4 的浮点数格式，字长 16 位。其中阶码 6 位，含一位阶符，补码表示，

以 2 为底；尾数 10 位，含一位数符，补码表示，规格化；某浮点数代码为(A27F)16，写出其十进制真值。

【答】浮点十六进制代码：A 2 7 F

浮点二进制代码：101000,1001111111

阶码（补码）为：101000

阶码二进制真值：－11000

阶码十进制真值：－24

尾数补码：1.001111111

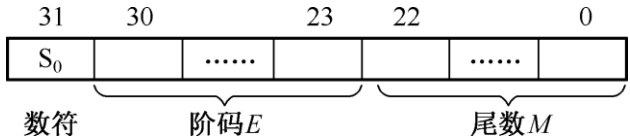
尾数二进制真值：－0.110000001

尾数十进制真值：$-(2^{-1}+2^{-2}+2^{-9})$

浮点十进制数真值：$-(2^{-1}+2^{-2}+2^{-9})2^{-24}=-(2^{-25}+2^{-26}+2^{-33})$

17．若采用教材中图 2-5 所示的 IEEE754 短浮点数格式，请将十进制数 37.25 写成浮点数，并且写出其二进制代码序列。

IEEE754 短浮点数格式为：



【答】将十进制数 37.25 转换为二进制数 100101.01，按 IEEE754 标准的短实数浮点格式要求，将 100101.01 表示为 1.0010101×2^5，故浮点数阶码的真值 e=5。于是，按 IEEE754 标准，得到：

数符 S_0=0，阶码（移码表示）E=（e+127）$_{10}$=（5+127）$_{10}$=（132）$_{10}$=（10000101）$_2$，M=001010100000…00。

最后得到 32 位浮点数的二进制数代码序列为：

01000010100101010000000000000000

18．某一个标准的 IEEE754 格式的短浮点数，表示为十六进制形式为 2AB03700H，请将其转化成对应的十进制数，要求写出主要的转化步骤。

【答】

IEEE754 格式：1 位数符 S+8 位阶码 E+23 位尾数 M，而

2AB03700H=0010 1010 1011 0000 0011 0111 0000 0000

符号位 S=0；阶码 E=010 1010 1=85，e=85-127=-42；M=011 0000 0011 0111 0000 0000

=（$1+2^{-2}+2^{-3}+2^{-10}+2^{-11}+2^{-13}+2^{-14}+2^{-15}$）

所以十进制数= $+(1+2^{-2}+2^{-3}+2^{-10}+2^{-11}+2^{-13}+2^{-14}+2^{-15})\times2^{-42}$

19．用变形补码计算 $X_{补}+Y_{补}$=？并指出是否有溢出。

（1）$X_{补}$=00.110011　　$Y_{补}$=00.101101

（2）$X_{补}$=00.010110　　$Y_{补}$=00.100101

（3）$X_{补}$=11.110011　　$Y_{补}$=11.101101

（4）$X_{补}$=11.001101　　$Y_{补}$=11.010011

【答】作补码加法时，应将两个操作数直接相加。对于变形补码而言，溢出标志=第一符号位⊕第二符号位，因此可以根据运算结果的两个符号位相同或相异，判断是否发生了溢出以及溢出的类型。

（1）
```
  00.110011
+ 00.101101
  01.100000
```
结果的两个符号位相异，表明发生了溢出；又由于第一符号位为 0 表示结果为正，所以发生了正溢出

（2）
```
  00.010110
+ 00.100101
  00.111011
```
结果的两个符号位相同，为 00，表明未发生溢出，结果为正

（3）
```
  11.110011
+ 11.101101
  11.100000
```
结果的两个符号位相同，为 11，表明未发生溢出，结果为负

（4）
```
  11.001101
+ 11.010011
  10.100000
```
结果的两个符号位相异，表明发生了溢出；又由于第一符号位为 1 表示结果为负，所以发生了负溢出

20．用变形补码计算 $X_{补}$-$Y_{补}$=？并指出是否有溢出。

（1）$X_{补}$=00.100011　　$Y_{补}$=00.101101

（2）$X_{补}$=00.110110　　$Y_{补}$=11.010011

（3）$X_{补}$=11.100011　　$Y_{补}$=00.110100

（4）$X_{补}$=11.101101　　$Y_{补}$=11.010011

【答】作补码减法时，将减数 $Y_{补}$变补，再与被减数 $X_{补}$相加。需要注意的是，变补就是将 $Y_{补}$的尾数连同符号位一起变补。

（1）将 $Y_{补}$变补，即$-Y_{补}$=11.010011。
```
  00.100011
+ 11.010011
  11.110110
```
结果的两个符号位相同，为 11，表明未发生溢出，结果为负。

（2）$-Y_{补}$=00.101101
```
  00.110110
+ 00.101101
  01.100011
```
结果的两个符号位相异，表明发生了溢出；由于第一符号位为 0 表示结果为正，所以发生了正溢出。

（3）$-Y_{补}$=11.001100
```
  11.100011
+ 11.001100
  10.101111
```
结果的两个符号位相异，表明发生了溢出；由于第一符号位为 1 表示结果为负，所以发生了负溢出。

（4）$-Y_{补}$=00.101101

11.101101

+ 11.101101

00.011010　　结果的两个符号位相同，为 00，表明未发生溢出，结果为正。

21．请按情况对下列数进行 16→32 的数位扩展。

【答】

（1）补码 D00F=1101000000001111，负数，符号扩展为 **FFFF**D00F；

（2）逻辑数 A12B，零扩展为 **0000**A12B；

（3）补码 7F3A=0111111100111010，正数，符号扩展为 **0000**7F3A；

（4）逻辑数 6B20，零扩展为 **0000**6B20；

（5）原码数 F02A=1111000000101010，负数，符号扩展为 1111F02A；

（6）原码数 5D0C=0101110100001100，正数，符号扩展为 00005D0C；

22．某指令的格式如下：（略）。

【答】

地址单元	1000H	1001H	1002H	1003H	1004H	1005H	1006H
大端	**20**	**34**	01	23	**56**	45	67
小端	**20**	**34**	23	01	**56**	67	45

23．完成下列字长 8 位的定点数和字长 32 位的浮点数运算。

【答】

（1）按原码规则计算-16+24 和-37-15

-16=1001 0000 $_{原}$，24=0001 1000 $_{原}$

1）加法、异号，则数值位求差：001 0000-001 1000=001 0000+110 1000=111 1000

2）无进位，则数值位求差结果再求补：求补（111 1000）=000 1000

3）符号位与被加数符号相反，结果=0000 1000 $_{原}$ = 8

-37=1010 0101 $_{原}$，15=0000 1111 $_{原}$

1）减法、异号，则数值位求和：010 0101+000 1111=011 0100

2）符号位为被减数符号，结果=1011 0100 $_{原}$= -52

2）无进位，则无溢出。

（2）按补码计算-16+24 和-37-15

-16=1111 0000 $_{补}$，24=0001 1000 $_{补}$

1）带符号相加：1111 0000+0001 1000=**1** 0000 1000（最高进位舍去）

2）溢出判断：最高数值位、符号位都有进位，则无溢出

3）运算结果：-16+24=0000 1000 $_{补}$=8

-37=1101 1011 补，15=0000 1111 补

1）带符号相减：1101 1011-0000 1111=1101 1011+求补(0000 1111)=**1** 1100 1100

2）溢出判断：最高数值位、符号位都有进位，则无溢出

3）运算结果：-37-15=1100 1100 补= -52

（3）按标准移码计算-16+24 和-37-15

-16 移=-16+**128**=112=0111 0000 移，24 移=24+**128**=152=1001 1000 移

1）模 2 加：0111 0000+1001 1000=**1** 0000 1000（最高进位舍去）

2）符号位变反：0000 1000→1000 1000

2）溢出判断：被加数、加数、和的符号位（最高位）不完全相同，无溢出

3）运算结果：-16+24=1000 1000 移=8

（4）按 IEEE754 短浮点数规则计算 $1.25\times2^{12}+1.375\times2^{11}$ 和 $1.25\times2^{12}-1.375\times2^{11}$。

$1.25\times2^{12}+1.375\times2^{11}$

1）对阶：$1.375\times2^{11}\rightarrow0.6875\times2^{12}$，对阶后阶码 E=12

2）尾数原码相加：1.25+0.6875→01.0100…00+00.10110…00=01.111100…00

(说明：相加时，数值位同号则请和，符号取被加数符号，最高位无进位则无溢出)

3）尾数规格化：01.111100…00 = +1.9375，符合规格化尾数要求

4）运算结果：0 1000 1011 111100…00 = 45F8 0000**H**

$1.25\times2^{12}-1.375\times2^{11}$

1）对阶：$1.375\times2^{11}\rightarrow0.6875\times2^{\mathbf{12}}$，对阶后阶码 E=12

2）尾数原码相减：1.25-0.6875→01.0100…00-00.10110…00

=01.0100…00+求补(00.10110…00)= 01.0100…00+11.010100…00=**1** 00.100100…00

(最高数值位有进位，相加结果为正)

3）尾数规格化：尾数左移 1 位、E-1，则尾数变为 1.00100…00、阶码变为 11

4）运算结果：0 1000 1010 00100…00 = 4510 0000**H**

24．用流程图描述下列算法流程。

（1）补码一位乘；（2）原码两位乘；（3）原码加减交替除；（4）补码加减交替除；（5）浮点加减运算；（6）浮点乘法运算；（7）浮点除法运算

【答】

可参照原码一位乘法的算法流程进行描述。

（1）补码一位乘法

首先进行初始化，并在乘数末位后设置附加位；然后根据两位判断位比较的结果进行操

作；每作一步操作计数器加 1，作完 n 步操作后若需修正结果，则再增加一步。流程如图 3-2-1 所示。

（2）原码两位乘法

首先，进行初始化，并设置欠账触发器 CJ，与两位乘数一起组成三位判断位；然后，根据三位判断位的状态进行操作；每作一步操作计数器加 1，作完 $n/2$ 步操作后若仍欠账，则再增加一步还账。流程如图3-2-2所示。

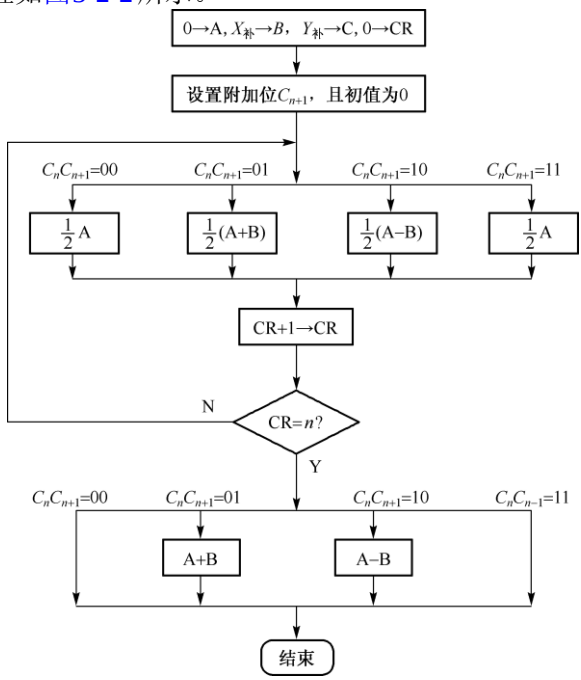


图 3-2-1　补码一位乘法流程

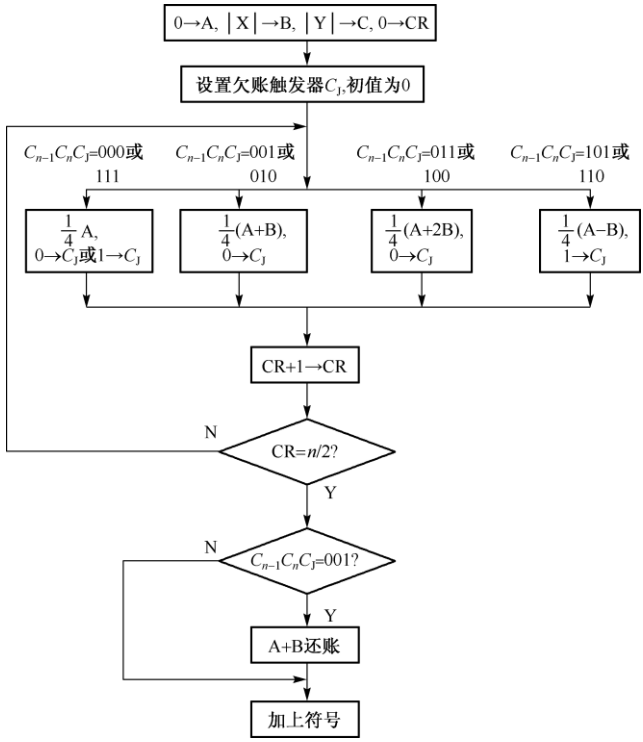


图 3-2-2　原码两位乘法流程

（3）原码加减交替除法

首先，进行初始化，然后将初始余数左移一位减除数，根据余数的正负确定商值，并根据商值决定下一步操作。每做一步操作，计数器加 1，若求 n 位商，则第 n 步操作后如果余数为负，那么再增加一步恢复余数。流程如图3-2-3 所示。

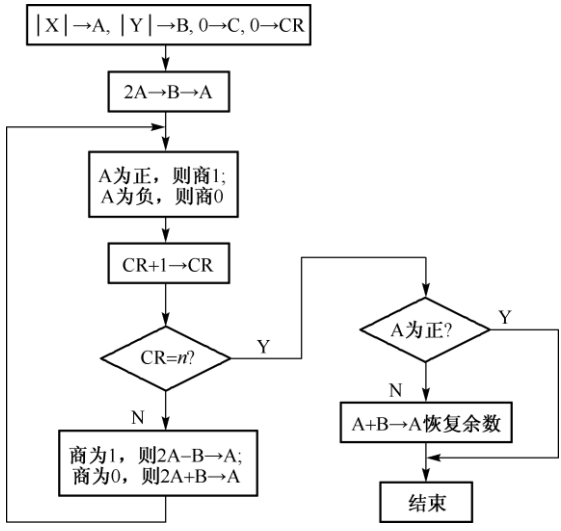


图 3-2-3　原码加减交替除法流程

（4）补码加减交替除法

先进行初始化，然后根据初始余数和除数的符号求商符，并根据商符决定下一步操作。以后每步操作都求商值，并且每做一步操作，计数器加 1。当做完第 n-1 步操作后求得 n-1 位假商，再做一步操作求得第步余数。最后对假商进行修正。流程图如图 3-2-4 所示。

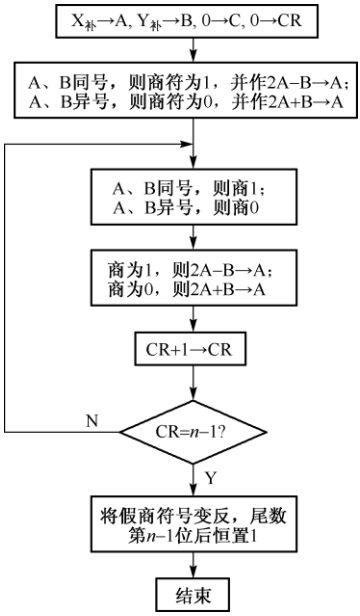


图 3-2-4　补码加减交替除法流程

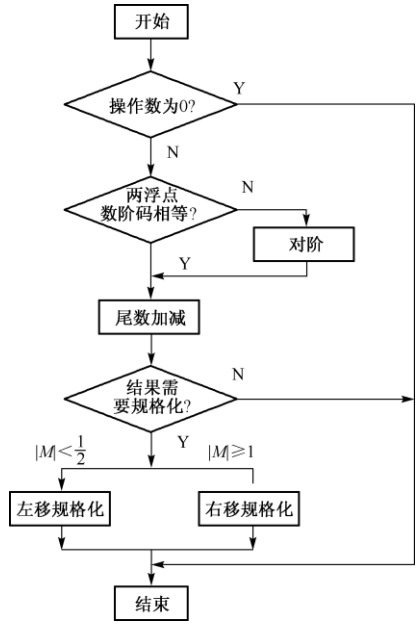


图 3-2-5　浮点加减流程

（5）浮点加减运算

浮点加减运算中，主要的操作是对阶和结果规格化。流程如图3-2-5 所示。

（6）浮点乘法运算

流程图如图 3-2-6 所示。

（7）浮点除法运算

流程图如图3-2-7 所示。

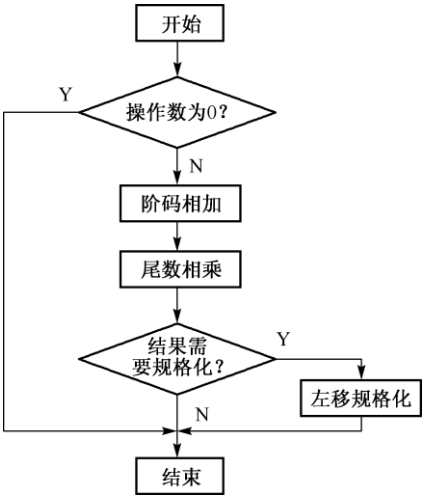


图 3-2-6　浮点乘法流程

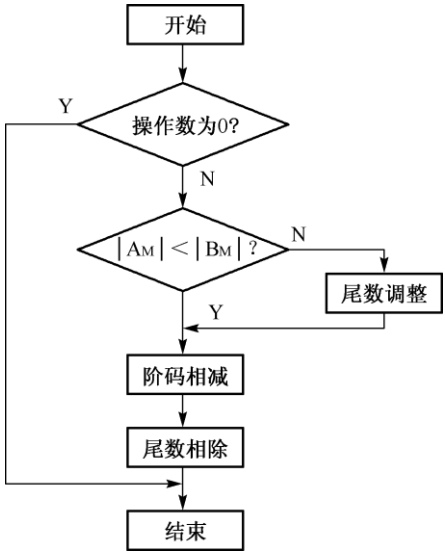


图 3-2-7　浮点除法流程图

操作数为 0 有两种情况：如果被除数为 0，则结果为 0；如果除数为 0，则另行处理。

25．参照教材中图 2-9 的形式，设计下列乘法器和除法器。

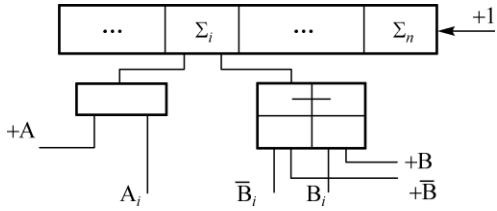
（1）补码一位乘的乘法器；（2）原码两位乘的乘法器

（3）原码加减交替除法器；（4）补码加减交替除法器

【答】

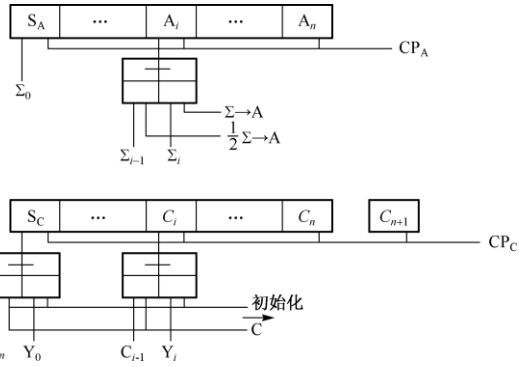
可在教材图 2-9 所示的原码一位乘法器是在加法器的基础上，加入其输入逻辑和输出逻辑而构成。因此要设计其他乘法器、除法器，关键在于根据算法所需的控制命令设计相应的输入逻辑和输出逻辑。

（1）补码一位乘的乘法器



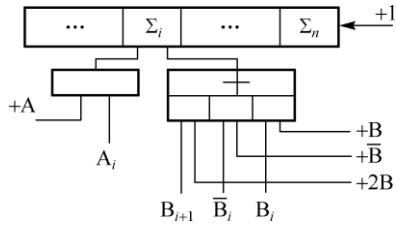
加法器的输入逻辑是 A、B 两个输入门，加法器输入端所需的控制命令有：+A、+B、+$\bar{B}$、+1。由于 A 端只有一个命令+A，故采用与门。B 端有两个命令+B、+$\bar{B}$，采用与或门。命令+1 加在加法器末端，控制末位加 1。

加法器的输出逻辑即是寄存器 A、C 的输入控制门，其所需的控制命令有：Σ→A、Σ/2→A、$\vec{C}$、CP_A、CP_C。寄存器 A 的输入控制门采用与或门，实现将累加和右斜1位送 A 寄存器以及将修正结果送 A 寄存器。寄存器 C 的输入控制门也采用与或门，在初始化时将 $Y_{补}$送 C 寄存器，在运算过程中实现 C 寄存器的右移。

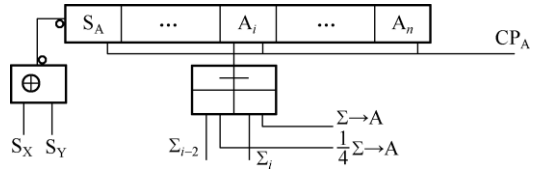


（2）原码两位乘的乘法器

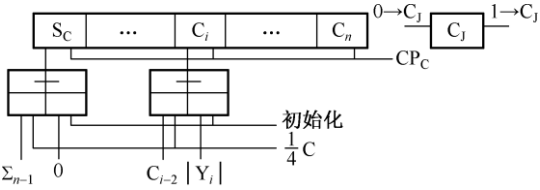
加法器 A 输入端所需的控制命令有+A；B 输入端所需的控制命令有：+B，+2B，$+\overline{B}$，+1。



寄存器 A、C 输入端所需的控制命令有：$\Sigma \to A$，$\frac{1}{4}\Sigma \to A$，$\frac{1}{4}C \to C$，CP_A、CP_C；欠账触发器 C_J 具有置位、复位功能。



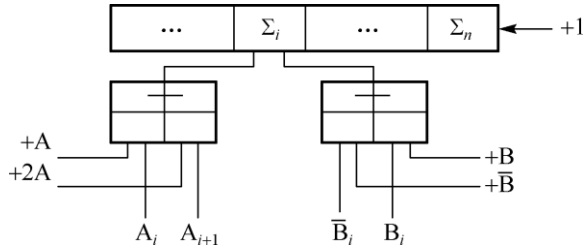
寄存器 A 用来存放乘积的高位部分，其符号位在运算结束后，根据被乘数和乘数的符号进行设置，即同号相乘结果为正、异号相乘结果为负。



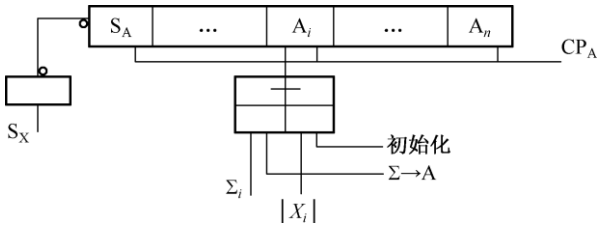
寄存器 C 在初始化时用来存放乘数的绝对值，其符号位设置为 0，以后参加移位，以决定最后是否还账。

（3）原码加减交替除法器

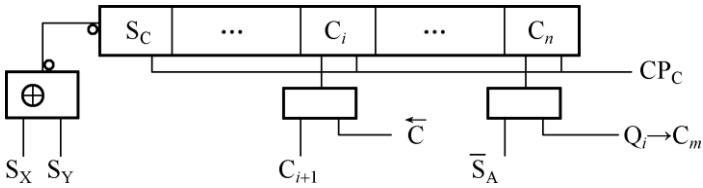
加法器 A 输入端所需的控制命令有：+A、+2A；加法器 B 输入端所需的控制命令有：+B、$+\overline{B}$、+1。



寄存器 A、C 输入端所需的控制命令有：$\Sigma\rightarrow$A、$\overleftarrow{C}$、$Q_i\rightarrow C_n$、CP_A、CP_C。



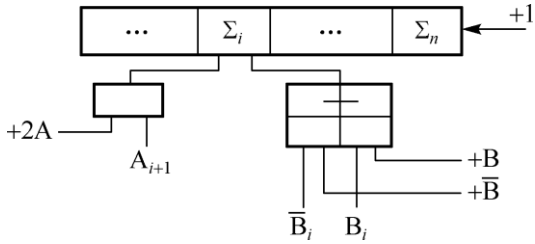
寄存器 A 用来存放余数，运算结束后，其符号位（表明实际余数的符号）由被除数的符号进行设置，即实际余数与被除数同号。



寄存器 C 用来存放商，商值与余数的符号相反，即余数为正商 1，余数为负商 0。每次均在 C_n 位上商，随着 C 左移，将各位商值依次移至 C 中相应位置。商的实际符号由被除数和除数的符号决定，同号相除为正，异号相除为负。

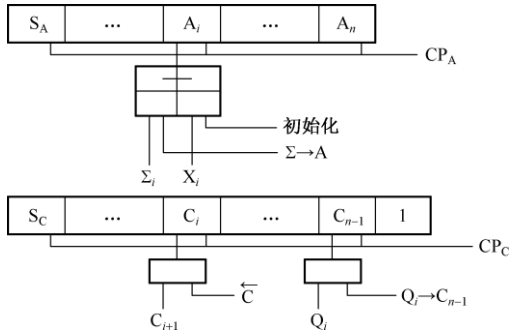
（4）补码加减交替除法器

加法器中的 A 输入端所需的控制命令有：+2A，加法器的 B 输入端所需的控制命令有：+B，+$\overline{B}$，+1。



寄存器 A、C 输入端所需的控制命令有：$\Sigma\rightarrow$A、$\overleftarrow{C}$、$Q_i\rightarrow C_{n-1}$、CP_A、CP_C。

每次均在 C_{n-1} 位上商，商的末位则恒置 1。商值 Q_i 由余数和除数的符号决定，同号商 1，异号商 0。运算结束后，通过 S_C 的置 1、置 0 端将商符变反，实现对商符的校正。



26．某乘法器基本字长 8 位（含 1 位数符），运算时可扩展为双符号位。请参照教材中图 2-9 所示的结构，画出乘法器的完整逻辑图。

【答】假设该乘法器用来实现原码一位乘法。

加法器设置 8 位，其输入端控制命令有+A、+B。

寄存器 A 存放累加和，设置 8 位，含一位符号位 S_A。由于 S_A 可能被累加时产生的进位所占用，因而另设置一位触发器 A_0，其值为 0，右移时移入 S_A，以保证 A 始终以绝对值形式参加运算。A 输入端的控制命令有 $1/2\Sigma \rightarrow A$、CP_A。

寄存器 B 存放被乘数的绝对值，设置 8 位，含一位符号位 S_B。B 输入端的控制命令为 CP_B。寄存器 C 存放乘数的绝对值，设置 8 位，含一位符号位 S_C。在运算过程中没有使用 S_C。C 输入端的控制命令有 $\vec{C}$、CP_C。该乘法器的逻辑图如图 3-2-8 所示。

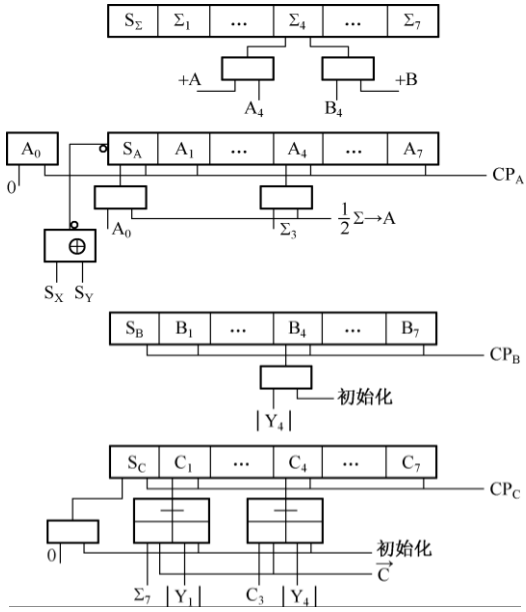


图 3-2-8　乘法器的逻辑图

27．欲写入代码 10011100（原始有效信息）：

（1）采用奇校验，写出配校验位后的校验码；

（2）采用偶校验，写出配校验位后的校验码；

【答】

奇校验的规律为：使整个校验码中“1”的个数为奇数个，偶校验的规律为：使整个校验码中“1”的个数为偶数个。因此配校验位后，分别采用奇、偶校验的校验码为：

	有效信息	校验位
奇校验	10011100	1
偶校验	10011100	0

28．欲写入 8 位有效信息 01101101，试将它编为海明校验码。以表格形式说明其编码方法，并分析所选用的编码方案具有什么样的检错与纠错能力。

【答】

（1）确定信息分为几组，增设几位校验位。有效信息为 $A_1A_2A_3A_4A_5A_6A_7A_8$=01101101，其位数 K=8，设分为 r 组，每组增设一个校验位，因此共有 r 位校验位，校验位与有校信息组成 n 位的海明校验码。

校验时每组产生一位校验信息，组成一个 r 位的指误字，可指出 2^r 种状态，其中全 0 表示无错，余下的组合可分别指明（2^r-1）位中的某一位错误。

因此 r 的值应满足：

$n = K+r \leqslant 2^r - 1$ 的要求，所以 r=4，信息应分为 4 组。

（2）如何分组：等编有效信息为 01101101，增设校验位 P_1、P_2、P_3、P_4，分为 4 组，可产生 4 位指误字 G_1、G_2、G_3、G_4。编码方案为证指误字代码与出错位序号相同，如各组采用偶校验。各位的排列和分组方案如表 3-2-6 所示：

表 3-2-6 海明校验码的分组方案

	1	2	3	4	5	6	7	8	9	10	11	12	指误字
	P_1	P_2	A_1	P_3	A_2	A_3	A_4	P_4	A_5	A_6	A_7	A_8	
第 4 组								√	√	√	√	√	G_4
第 3 组				√	√	√	√					√	G_3
第 2 组		√	√			√	√			√	√		G_2
第 1 组	√		√		√		√		√		√		G_1

（3）对有效信息进行编码

第 1 组	第 2 组	第 3 组	第 4 组
$A_1A_2A_4A_5A_7$=01010	$A_1A_3A_4A_6A_7$=01010	$A_2A_3A_4A_8$=1101	$A_5A_6A_7A_8$=1101
P_1 = 0	P_2= 0	P_3 =1	P_4 = 1

因此最后得到的 12 位海明校验码为：000111011101。

编码方案的检错和纠错模式如表 3-2-7 所示。

表 3-2-7 海明校验码的检错、纠错模式

1	2	3	4	5	6	7	8	9	10	11	12	指误字
P_1	P_2	A_1	P_3	A_2	A_3	A_4	P_4	A_5	A_6	A_7	A_8	G_4 G_3 G_2 G_1
0	0	0	1	1	1	0		1	1	0	1	0 0 0 0

错												0001
	错											0010
		错										0011
			错									0100
				错								0101
					错							0110
						错						0111
							错					1000
								错				1001
									错			1010
										错		1011
											错	1100

29．某海明编码 K=4，r=3，请为此设计其编码、译码、纠错逻辑。

【答】

解题分析：如该海明校验码分组规则如表 3-2-8 所示，编码逻辑如图 3-2-9 所示。

表 3-2-8 海明校验码分组规则

	1	2	3	4	5	6	7	指误字
	P_1	P_2	A_1	P_3	A_2	A_3	A_4	
第 3 组				√	√	√	√	G_3
第 2 组		√	√			√	√	G_2
第 1 组	√		√		√		√	G_1

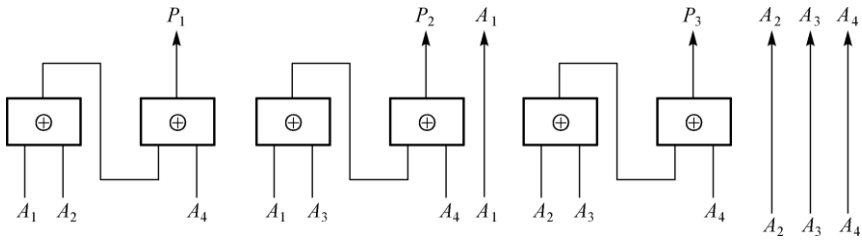


图 3-2-9　海明校验编码逻辑

其译码、纠错逻辑如图 3-2-10 所示，当从外部接收到一组海明码后，送往图 3.4.22 所示的译码电路进行分组奇偶检测，得到一组检错信息，译码后作为控制信号，决定是否要纠正。如 $G_3G_2G_1 = 0$，则最上面的一异或门控制端均为 0，输出 $P_1' \sim A_4' = P_1 \sim A_4$，即不需要纠正。若 $G_3G_2G_1 = 101$，则第 5 位异或门控制端为 1，$A_2' = A_2$，即将出错的第 5 位进行变反纠正，其余未错的各位保持不变。

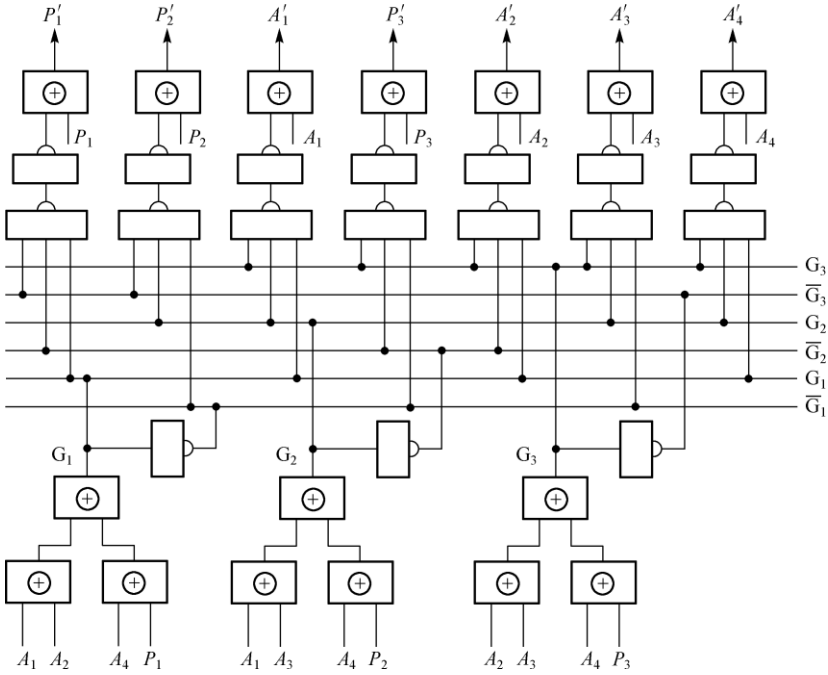


图 3-2-10 海明译码、纠错逻辑

30．某循环校验码，生成多项式 $X_3+X_1+X_0$ 请为此设计一个模 2 除法器。

【答】

模 2 除的分解步骤可归纳为两种类型：如果部分余数（包括被除数）最高位为 0，则将部分余数左移一位； 如果部分余数最高位为 1，则对生成多项式 G(x)作模 2 减，然后余数左移一位。由此，我们可以描述出模 2 除法器逻辑的主要组成，如图 3-2-11 所示。

设置一个移位寄存器，模 2 减的特点是“减 0 不变、1 变反”，可用异或门实现；又因为每做一步除法都要左移一位，所以异或门的输出可直接送往高一位触发器的 D 端，即模 2 减与左移在一拍内完成。用最高位的状态去控制作模 2 减并左移，或单纯左移，控制各位模 2 减的条件就是生成多就是生成多项式 $G(x)$。当 $G(x)$确定后，如本题 $G(x)=1011$，可得到下列几个逻辑式：

$$D_1=(Q_1\cdot 0)\oplus Q_2=Q_2$$
$$D_2=(Q_1\cdot 1)\oplus Q_3=Q_1\oplus Q_3$$
$$D_3=(Q_1\cdot 1)\oplus Q_4=Q_1\oplus Q_4$$

以上逻辑式的含义是：当最高位 $Q_1=0$ 时有 $D_i=Q_{i+1}$ 即单纯左移；当 $Q_1=1$ 时，$Q_2Q_3Q_4$ 分别与 $G(x)$按位异或，再送往高位 D，即做模 2 减后左移一位。设置控制电位 K_3 控制模 2 除。D_4 条件要联系到整个编码、译码电路如何设计，应能串行输入待编信息或待译信息 A，也能在模 2 除的过程中补 0。

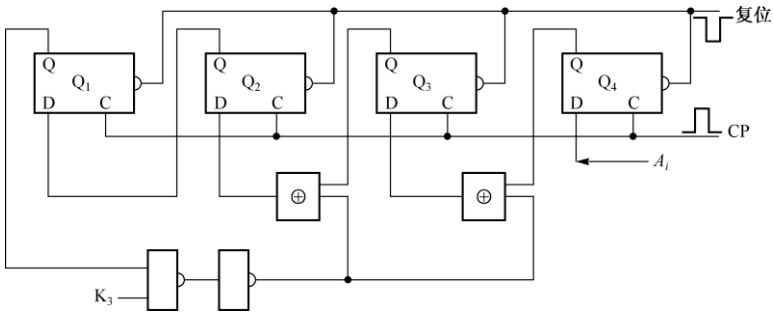


图 3-2-11　模 2 除法器的逻辑结构图

31．设计一套硬件逻辑，以实现循环校验码的编码、译码、校正。画出粗框图。

【答】

假设待编码字为 $M(x)$共 k 位，生成多项式为 $G(x)= x^r +a_{r-1}x^{r-1} + \cdots + a_2x^2 + a_1x^1 + 1$。余数为 $R(x)$，则其编码电路如图3-2-12 所示。

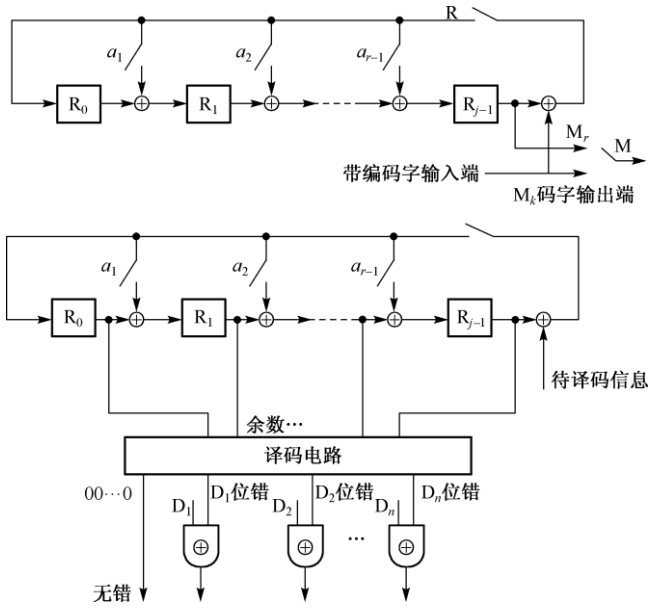


图 3-2-12　循环校验码的编码、译码、校正电路

当 a_i 为 1 时，开关 a_i 闭合，当 a_i 为 0 时，开关 a_i 断开。开始编码时，开关 M 置于 M_k 位置，同时开关 R 闭合，待编信息位由信息输入端从高位到低位串行送入，一方面输入信息从码字输出端输出，另一方面输入信息进入该编码电路做循环运算。当信息送入完毕时，寄存器 R_0、R_1、…、R_{j-1} 中的内容就是余数的值，这时断开开关 R，将开关 M 的位置设置至 M_r 处，则余数就拼接在待编信息位之后，实现编码过程。

32．设计一套子程序，以实现循环校验码的编码、译码、校正。画出流程图。

【答】

流程图如图 3-2-13 所示。

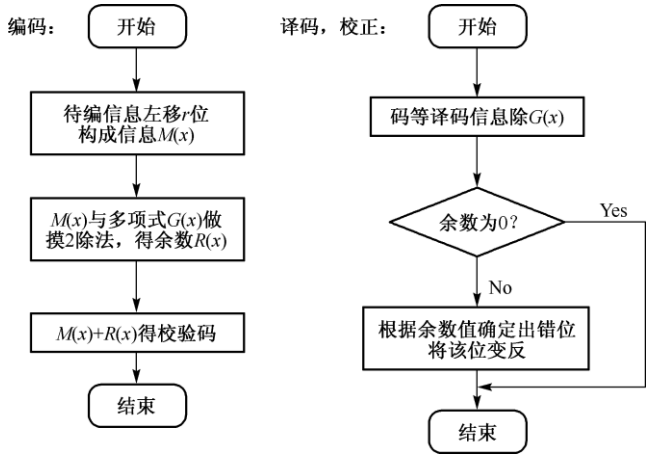


图 3-2-13　循环校验码的编码、译码、校正子程序流程

33．将 4 位有效信息 1001 编成循环校验码，选择生成多项式 $X^3+X^1+X^0$。写出编码过程。

【答】

（1）编码方法

$M(x) = x^3+x^0$，　即 1001　　$(k = 4)$

$M(x)\cdot x^r = x^6+x^3$，　即 1001000　　$(r=3)$

$G(x)=x^3+x^1+x^0$，　即 1011　　$(r+1=4)$

$$\frac{M(x)}{G(x)}=\frac{1001000}{1011}=1010+\frac{110}{1011}$$　（模 2 除）

（2）编码后的校验码为：

$M(x)\cdot x^3+R(x)$=1001000+110=1001110（模 2 加）。

3.3　第 3 章习题解答

1．简要解释下列名词术语

【答】

CPU：是由运算器和控制器组成的计算机硬件系统的核心部件，也称中央处理器。

运算器：在计算机中用来对数据进行加工处理的部件。传统运算器包含输入逻辑、算术逻辑运算部件 ALU、输出逻辑和一部分寄存器。

控制器：在计算机中用来产生各种控制命令（微命令），控制全机操作的部件。传统控制器包含微命令产生部件、时序系统和一部分寄存器。

通用寄存器：可由 CPU 编程访问，能实现多种功能的寄存器。例如可提供操作数，存放运算结果，用作地址指针，作为变址寄存器、基址寄存器、计数器等等。

暂存器：为避免破坏通用寄存器的内容，用来暂时存放某些中间结果的寄存器。

指令寄存器 IR：用来存放现行指令的寄存器。当需要执行某条指令时，先将该指令从存储器取出，并存入 IR 中，然后再对 IR 的内容进行译码。

程序计数器 PC：用来指示指令在存储器中存放位置的寄存器。PC 的内容是指令所在

存储单元的地址，取指后，PC 内容增量计数，指向下一条指令的地址。

程序状态字 PSW：用来记录现行程序的运行状态和指示程序工作方式的寄存器。

时序系统：用来产生时序信号（如周期、节拍、脉冲等）的部件称为时序系统或时序发生器，它由一个振荡器和一组计数分频器组成。

微命令：在计算机中用来控制微操作（如逻辑门的开或关、寄存器的打入或清除等操作）的控制命令称为微命令，也称为微操作控制信号。

组合逻辑控制：简单地讲，由硬连逻辑电路产生微命令的方式称为组合逻辑控制方式。它的基本思想如下：

综合、化简产生微命令的条件，形成相应逻辑式，并用组合逻辑电路实现；执行指令时，由组合逻辑电路(微命令发生器)在相应时间发出所需微命令，控制有关操作。

微程序控制：简单地讲，由微指令译码产生微命令的方式称为微程序控制方式。它的基本思想如下：

将若干微命令编制成一条微指令，控制实现一步操作；将若干微指令组成一段微程序，解释执行一条机器指令；将微程序事先存放在控制存储器中，执行机器指令时再取出。

地址结构：指令的地址结构是指在指令中明确给出的地址。

显地址：在指令代码中明显给出的地址，如在指令中写明操作数的主存单元地址或寄存器号，则这种地址称为显地址。

隐地址：在指令中不明显给出地址码，地址以隐含方式约定，这种隐含约定的地址称为隐地址。

寻址方式：指令中以什么方式提供操作数或操作数地址，称为寻址方式。

立即寻址：由指令直接给出操作数，在取出指令的同时也就取出了可以立即使用的操作数，这种寻址方式称为立即寻址。

直接寻址：由指令直接给出操作数地址，根据该地址可以从主存（或寄存器）中取出操作数，或向主存（或寄存器）写入数据，这种寻址方式称为直接寻址。

寄存器寻址：在指令中给出寄存器号，从该寄存器号所指的寄存器中取出操作数或将数据传送到该寄存器号所指的寄存器中。这种寻址方式实为寄存器直接寻址。

间接寻址：在指令中给出间址单元地址码（即操作数地址的存放单元地址），按照该地址访问主存中该间址单元，从中读取操作数地址，接着按操作数地址再次访问主存，从该单元中读取或向该单元写入操作数。

寄存器间址：由指令给出寄存器号，在该寄存器号所指定的寄存器中存放着操作数地址，按此地址访问主存，读取或写入操作数。

间址单元：在间接寻址方式中，存放操作数地址的主存编址单元称为间址单元。

变址寻址：在指令中的地址部分给出一个形式地址，并且指定一个寄存器作为变址寄存器，将变址寄存器的内容（称为变址量）与形式地址相加，得到操作数地址（称为有效地址）；按有效地址访问主存，从相应的主存单元中读得操作数或向该单元写入数据。

基址寻址：在指令中给出一个形式地址（作为位移量），并且指定一个寄存器作为基址寄存器（该基址寄存器内容作为基准地址）；将基址寄存器内容和形式地址相加，其和作为操作数有效地址；按有效地址访问主存，从该单元读取操作数或向该单元写入数据。

相对寻址：指用程序计数器 PC 的内容作为基准地址，指令中给出的形式地址作为位移量的基址寻址方式。

页面寻址：将程序计数器 PC 的高位段作为操作数有效地址的高位段，指令中给出的形式地址作为操作数有效地址的低位段，将这两部分拼接构成操作数有效地址，这种寻址方式称为页面寻址方式。

堆栈：一种按“后进先出”（或称“先进后出”）存取顺序进行存取的存储结构。

栈顶：堆栈是一个连续的存储区，其一端固定称为栈底，存放最先压入的数；堆栈的另一端是浮动的，称为栈顶，对堆栈的读写都是对栈顶单元进行的；对堆栈的寻址也就是对栈顶单元的寻址，随着堆栈操作的进行，栈顶位置也发生变化。

堆栈指针：指用于指向栈顶位置的寄存器 SP，堆栈指针 SP 的内容是栈顶单元地址。

CISC：具有复杂指令集合的计算机称为 CISC（Complex Instruction Set Computer）。

RISC：采用精简指令系统的计算机称为 RISC（Reduced Instruction Set Computer）。

全加器：含有三个输入量（两个操作数、一个来自低位的进位信号）的二进制加法单元称为全加器。

并行加法器：用多位全加器实现多位数同时相加的加法器称为并行加法器。

进位链：提供进位信号传递通路的硬连逻辑电路称为进位链。

串行进位：进位信号逐级产生，低位进位向高位传递。

并行进位：各个进位信号同时产生，高位进位不依赖于低位进位。

分组进位：也称分级同时进位，即将多位全加器分成若干组，组内采用并行进位，组间也采用并行进位。

指令周期：一条指令从取指到执行完，所用的全部时间称为指令周期。

工作周期：一个指令周期中，完成某一阶段操作所需的时间称为工作周期。如取指周期、源周期、目的周期、执行周期等。

时钟周期：CPU 执行一步操作所需的时间称为一个时钟周期。时钟周期作为时序基准，在一个计算机中其长度是固定不变的。

微指令周期：读取并执行一条微指令所用的时间称为微指令周期。

总线周期：通过总线传送一次数据所用的时间称为总线周期。在同步方式下，一个总线周期可能包含若干个时钟周期。

主存读/写周期：指主存进行连续读/写所允许的最小时间间隔，即两次读/写操作之间的最小间隔。

微指令：将一步操作所需的微命令编写在一串代码中，这串代码称为微指令。它由微命令字段和微地址字段组成。

微程序：由若干条微指令组成一段微程序，用来解释执行一条机器指令。

控制存储器：用来存放各微程序段的专用存储器。它属于 CPU 范畴而不属于主存范畴。

增量方式：这是一种产生微地址的方式，即以顺序执行为主，后续微地址在现行微地址的基础上增量产生，并配合多种常规转移方式。

断定方式：这也是一种产生微地址的方式，通过直接给定和测试断定相结合来产生后续微地址。即后续微地址的一部分由现行微指令给定，另一部分则由测试判断来确定。

SMT：同步多线程，是一种在一个时钟周期内 CPU 能够执行分别来自多个线程的指令的硬件多线程技术。

超线程：利用特俗的硬件指令，把两个逻辑内核模拟成两个物理芯片，让单个处理器能进行线程级的并行计算，由 Intel 在 2002 年开发成功。

多核：在一个处理器芯片上集成多个完整内核，不同内核之间可以并行执行指令，可以实现多个低频内核产生单个高频内核的处理效能。

2．简化地址结构的基本途径是什么？

【答】

在指令中减少显地址的数量，即使用隐地址方式给出地址，指令中的地址（段）个数就可减少。

3．减少指令中一个地址信息的位数的方法是什么？

【答】

采用寄存器寻址、寄存器间址等以寄存器为基础的寻址方式可以大大减少指令中一个地址的信息位数。

4．某主存储器部分单元的地址码与存储器内容对应关系如下：

地址码	存储内容
1000H	A307H
1001H	0B3FH
1002H	1200H
1003H	F03CH
1004H	D024H

（1）若采用寄存器间址方式读取操作数，指定寄存器 R_0 的内容为 1002H，则操作数是多少？

（2）若采用自增型寄存器间址方式(R_0)+读取操作数，R_0 内容为 1000H，则操作数是多少？指令执行后 R_0 的内容是多少？

（3）若采用自减型寄存器间址方式–(R_1)读取操作数，R_1 内容为 1003H，则操作数是多少？指令执行后 R_1 的内容是多少？

（4）若采用变址寻址方式 $X(R_2)$读取操作数，指令中给出形式地址 d=3H，变址寄存器 R_2 内容为 1000H，则操作数是多少？

【答】

（1）操作数是 1200H。

（2）操作数是 A307H，指令执行后 R0 的内容变为 1001H。

（3）操作数是 1200H，指令执行后 R1 的内容为 1002H。

（4）操作数为 F03CH。

5．对 I/O 设备的编址方法有哪几种？请简要解释。

【答】

对 I/O 设备的编址方法实际上就是对 I/O 接口中有关寄存器及相应部件的编址方法，有两大类型。

① 外围设备单独编址。

早期是为每台 I/O 设备分别分配一个设备码，每个设备下属 n 个接口寄存器，在 I/O 指令中给出设备码，并指明接口的哪个寄存器，从而实现 CPU 对外设的访问。现在普遍采用为各 I/O 接口的每个有关寄存器分别分配一种 I/O 端口地址，指令中给出端口地址，也就知道 CPU 访问哪一台设备及其接口寄存器。

② 外围设备与主存统一编址。

将各 I/O 接口中的有关寄存器与主存的各编址单元统一编址，为它们分配统一的总线地址。在传送指令中给出这类总线地址，CPU 就可以访问相应的 I/O 设备及其接口寄存器。

6．I/O 指令的设置方法有哪几种？请简要解释。

【答】

通常有三类常见的 I/O 指令设置方法。

① 在指令系统中设置专门的 I/O 指令，可对外围设备单独分配设备码，或给 I/O 接口的有关寄存器分配专门的端口地址，这种 I/O 指令称为显式 I/O 指令。

② 采用通用的数据传送指令实现 I/O 操作，相应地将外围设备接口的有关寄存器与主存统一编址。这种 I/O 指令是隐含在传送指令中，所以又称为隐式 I/O 指令。

③ 通过 I/O 处理器（或 I/O 处理机）控制 I/O 操作。这种方式下，I/O 指令可分为两级：CPU 调用 IOP 的指令和 IOP 本身的指令。

7．用 74181 和 74182 芯片构成一个 64 位 ALU，采用分级分组并行进位链结构。画出逻辑图，并注明输入、输出信号。

【答】

ALU 逻辑图如图3-3-1所示。

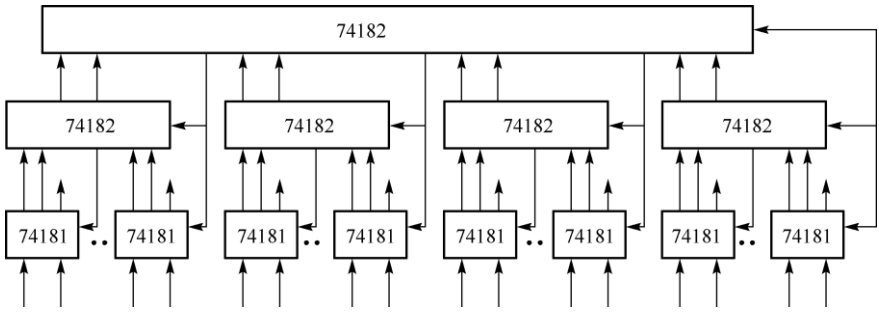


图 3-3-1　ALU 逻辑图

首先，计算 74181 芯片的数量。由于 1 片 74181 实现 4 位数运算，所以 64 位数需要 16 片 74181。

其次，计算 74182 芯片的数量。1 片 74182 只能接受 4 个小组的进位辅助函数，并同时输出 4 个组间进位信号，因此将 16 片 74181 分为 4 组，每组 4 片，组内共用 1 片 74182，同时产生组内各 74181 的初始进位。4 片 74182 再共用另 1 片 74182，同时产生 4 个组间进位信号。所以，一共需要 5 片 74182 芯片。

8．试比较组合逻辑控制和微程序控制的优缺点及应用场合。

【答】

组合逻辑控制和微程序控制是用来控制如何产生微命令的两种方式。

组合逻辑控制方式是直接通过逻辑门电路产生微命令的，因而它的主要优点是产生微命令的速度很快。其主要缺点有两点：其一是设计不规整，因而难于实现设计自动化；其二是采用硬连逻辑，不易修改和扩展指令系统的功能。组合逻辑控制方式主要用于高速计算机或小规模计算机中。

微程序控制方式是通过执行微指令来产生微命令的，主要优点如下：

① 设计规整，设计效率高；

② 易于修改、扩展指令系统功能；

③ 结构规整、简洁，可靠性高；

④ 性价比高。

其主要缺点是：产生微命令的速度慢，因为要多次访问控存读取微指令，访存速度限制了产生微命令的速度；另外，机器的执行效率不高，因为微指令格式较简单，没有充分发挥数据通路本身所具有的并行能力。微程序控制方式主要用于速度要求不高、功能较复杂的机器中，特别适合于系列机。

9．根据模型机数据通路结构，拟定 MOV 指令流程在 ST_2、ST_3、ST_4 中的操作时间表。

【答】

按教材中图 3-45 的 MOV 指令流程图，ST_2 完成一次内部数据通路操作，根据 ALU 输入选择、ALU 功能选择、输出移位选择、结果分配等四段操作设置相应的微命令。ST_2 结束后，若寻址方式为自增型寄存器间址，则进入目的周期 DT 或执行周期 ET；若寻址方式为自增型双间址或变址，则继续 ST，进入 ST_3，完成一次内部数据通路操作，然后进入 ST_4，完成一次从存储器读出，并经内部数据通路传送的操作。

ST_2、ST_3、ST_4 中的操作时间表如表 3-3-1 所示。

表 3-3-1　MOV 指令源周期操作时间表

电位型微命令		脉冲型微命令
ST_2:	R_i→A	CP R_i
	SP→A	CP SP
	PC→A	CP PC
	A+1	CP T　($\overline{P}$)
	DM	CP FT　($\overline{P}$)
	1→ST [@(R)+ ∨ X(R)]	CP ST　($\overline{P}$)

	T+1 [@(R)+ ∨ X(R)] 1→DT [(R)+] $\overline{DR}$ 1→ET [(R)+] DR	CP DT ($\overline{P}$) CP ET ($\overline{P}$)
ST_3:	C→A R_i→B PC→B 输出 A [@(R)+] A+B [X(R)] DM T+1	CP MAR CP T ($\overline{P}$)
ST_4:	EMAR R SMDR MDR→B 输出 B DM 1→DT 或 1→ET	CP C CP T ($\overline{P}$) CP FT ($\overline{P}$) CP ST ($\overline{P}$) CP DT ($\overline{P}$) CP ET ($\overline{P}$)

注：上述操作时间表中基本上未考虑逻辑条件。

10. 拟出下述指令流程及操作时间表。

(1) MOV (R_0), (SP) +;
(2) MOV (R_1) +, X (R_0);
(3) MOV R_2, (PC) +;
(4) MOV - (SP), (R_3);
(5) ADD R_1, X (R_0);
(6) SUB (R_1) +, (R_2);
(7) AND - (R_0), R_1;
(8) OR R_2, (R_0) +;
(9) EOR (R_0), (R_1);
(10) INC X (PC);
(11) DEC (R_0);
(12) COM (R_1) +
(13) NEG - (R_2);
(14) SL R_0;
(15) SR R_3;
(16) JMP SKP;
(17) JMP R_0;
(18) JMP X (PC);

（19）RST（SP）+;

（20）JSR（R1);

【答】

拟定指令流程关键是要了解模型机各类指令功能，掌握模型机各种寻址方式，熟悉模型机数据通路结构。此外，还需清楚指令中源地址和目的地址的位置，教材中约定，源在后，目的在前。

（1）MOV　(R_0), (SP)+;

该指令完成出栈操作。需要注意的是，在操作时间表中，当进行周期转换时，5 个打入命令都发，以便简化逻辑条件。另外，为简单起见，没有分开列出电位型微命令和脉冲型微命令。

1）指令流程	2）操作时间表
FT：M→IR	FT：EMAR、R、SIR、
PC+1→PC	PC→A、A+1、DM、CPPC、1→ST、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)；
ST：SP→MAR	ST：SP→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)，
M→MDR→C	EMAR、R、SMDR、MDR→B、输出 B、DM、CPC、T+1、CPT($\overline{P}$)，
SP+1→SP	SP→A、A+1、DM、CPSP、1→DT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)；
DT：R_0→MAR	DT：R_0→A、输出 A、DM、CPMAR、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)；
ET：C→MDR	ET：C→A、输出 A、DM、CPMDR、T+1、CPT($\overline{P}$)，
MDR→M	EMAR、W、T+1、CPT($\overline{P}$)，
PC→MAR	PC→A、输出 A、DM、CPMAR、1→FT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

（2）MOV　(R_1)+, X(R_0);

该指令将采用变址方式获得的源操作数送入目的单元。由于模型机指令字长有限，变址所需的形式地址放在现行指令所在单元的下一个单元中，取指后由 PC 指示。

1）指令流程	2）操作时间表
FT：M→IR	FT：微命令同前；
PC+1→PC	
ST：PC→MAR	ST：PC→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)
M→MDR→C	EMAR、R、SMDR、MDR→B、输出 B、DM、CPC、T+1、CPT($\overline{P}$)
C+ R_0→MAR	C→A、R_0→B、A+B、DM、CPMAR、T+1、CPT($\overline{P}$)，
M→MDR→C	EMAR、R、SMDR、MDR→B、输出 B、DM、CPC、T+1、CPT($\overline{P}$)
PC+1→PC	PC→A、A+1、DM、CPPC、1→DT、CPT($\overline{P}$)、CPFT($\overline{P}$)、

CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

DT：R_1→MAR　　DT：R_1→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)

R_1+1→R_1　　R_1→A、A+1、DM、CP R_1、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)；

ET：C→MDR　　ET：微命令同前

MDR→M　　微命令同前

PC→MAR　　微命令同前

（3）MOV　R_2, (PC)+;

该条指令的源采用立即寻址，指令的功能是将立即数送入 R_2。与变址中的形式地址一样，16 位立即数存放在现行指令所在单元的下一个单元中，取指后由 PC 指示。

1）指令流程　　2）操作时间表

FT：M→IR　　FT：微命令同前；

PC+1→PC

ST：PC→MAR　　ST：PC→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)

M→MDR→C　　EMAR、R、SMDR、MDR→B、输出 B、DM、CPC、T+1、CPT($\overline{P}$)

PC+1→PC　　PC→A、A+1、DM、CPPC、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

ET：C→R_2　　ET：C→A、输出 A、DM、CPR_2、T+1、CPT（P）

PC→MAR　　PC→A、输出 A、DM、CPMAR、1→FT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

（4）MOV　-(SP), (R_3);

该指令执行入栈操作。

1）指令流程　　2）操作时间表

FT：M→IR　　FT：微命令同前；

PC+1→PC

ST：R_3→MAR　　ST：R_3→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)

M→MDR→C　　EMAR、R、SMDR、MDR→B、输出 B、DM、CPC、1→DT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

DT：SP-1→SP、MAR　DT：SP→A、A-1、DM、CPSP、CPMAR、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

ET：C→MDR　　ET：微命令同前

MDR→M　　微命令同前

PC→MAR　　微命令同前

（5）ADD　R_1, X(R_0);

该条指令的功能是将变址获得的源操作数与目的操作数相加，结果存放在目的地址 R_1 中。

1）指令流程　　2）操作时间表

FT：M→IR　　FT：微命令同前
　　PC+1→PC
ST：PC→MAR　　ST：微命令同前
　　M→MDR→C　　微命令同前
　　C+ R_0→MAR　　微命令同前
　　M→MDR→C　　微命令同前
　　PC+1→PC　　PC→A、A+1、DM、CPPC、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
ET：C+ R_1→R_1　　ET：C→A、R_1→B、A+B、DM、CPR_1、T+1、CPT($\overline{P}$)
　　PC→MAR　　PC→A、输出 A、DM、CPMAR、1→FT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

（6）SUB　(R_1)+, (R_2);

该指令的功能是用源操作数减去目的操作数，结果存放在目的单元中。

1）指令流程　　2）操作时间表

FT：M→IR　　FT：微命令同前
　　PC+1→PC
ST：R_2→MAR　　ST：R_2→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)
　　M→MDR→C　　EMAR、R、SMDR、MDR→B、输出 B、DM、CPC、1→DT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
DT：R_1→MAR　　DT：R_1→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)
　　M→MDR→D　　EMAR、R、SMDR、MDR→B、输出 B、DM、CPD、T+1、CPT($\overline{P}$)
　　R_1+1→R_1　　R_1→A、A+1、DM、CPR_1、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
ET：C-D→MDR　　ET：C→A、D→B、A-B、DM、CPMDR、T+1、CPT($\overline{P}$)
　　MDR→M　　EMAR、W、T+1、CPT($\overline{P}$)
　　PC→MAR　　PC→A、输出 A、DM、CPMAR、1→FT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

（7）AND　-(R_0), R_1;

该指令将两数相与，由于源采用寄存器寻址，取指后进入目的周期。

1）指令流程　　2）操作时间表

FT：M→IR　　FT：只需将 1→ST 改为 1→DT，其余微命令同前
　　PC+1→PC
DT：R_0-1→R_0、MAR　　DT：R_0→A、A-1、DM、CPR_0、CPMAR、T+1、CPT($\overline{P}$)
　　M→MDR→D　　EMAR、R、SMDR、MDR→B、输出 B、DM、CPD、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
ET：R_1∧D→MDR　　ET：R_1→A、D→B、A∧B、DM、CPMDR、T+1、CPT($\overline{P}$)
　　MDR→M　　EMAR、W、T+1、CPT($\overline{P}$)
　　PC→MAR　　微命令同前

（8）OR　R_2, (R_0)+;

1）指令流程	2）操作时间表
FT：M→IR	FT：微命令同前
PC+1→PC	
ST：R_0→MAR	ST：R_0→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)
M→MDR→C	EMAR、R、SMDR、MDR→B、输出 B、DM、CPC、T+1、CPT($\overline{P}$)
R_0+1→R_0	R_0→A、A+1、DM、CPR$_0$、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
ET：C∨R_2→R_2	C→A、R_2→B、A∨B、DM、CPR$_2$、T+1、CPT($\overline{P}$)
PC→MAR	微命令同前

（9）EOR　(R_0), (R_1);

1）指令流程	2）操作时间表
FT：M→IR	FT：微命令同前
PC+1→PC	
ST：R_1→MAR	ST：R_1→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)
M→MDR→C	EMAR、R、SMDR、MDR→B、输出 B、DM、CPC、1→DT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
DT：R_0→MAR	DT：R_0→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)
M→MDR→D	EMAR、R、SMDR、MDR→B、输出 B、DM、CPD、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
ET：C EOR D→MDR	ET：C→A、D→B、A EOR B、DM、CPMDR、T+1、CPT($\overline{P}$)
MDR→M	微命令同前
PC→MAR	微命令同前

（10）INC　X(PC);

该加 1 指令采用相对寻址，位移量放在现行指令所在单元的下一单元中，取指后由 PC 指示。

1）指令流程	2）操作时间表
FT：M→IR	FT：EMAR、R、SIR、PC→A、A+1、DM、CPPC、1→DT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
PC+1→PC	
DT：PC→MAR	DT：PC→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)
M→MDR→D	EMAR、R、SMDR、MDR→B、输出 B、DM、CPD、T+1、CPT($\overline{P}$)
PC+D→MAR	PC→A、D→B、A+B、DM、CPMAR、T+1、CPT($\overline{P}$)
M→MDR→D	EMAR、R、SMDR、MDR→B、输出 B、DM、CPD、T+1、CPT($\overline{P}$)
PC+1→PC	PC→A、A+1、DM、CPPC、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
ET：D+1→MDR	ET：D→A、A+1、DM、CPMDR、T+1、CPT($\overline{P}$)
MDR→M	微命令同前
PC→MAR	微命令同前

在模型机的相对寻址中，基准地址 PC 是指向位移量，而不是指向下条指令，因此在 DT 中，PC+1→PC 的操作应在变址计算 PC+D→MAR 的操作之后进行。

（11）DEC　(R_0);

1）指令流程	2）操作时间表
FT：M→IR PC+1→PC	FT：微命令同前
DT：R_0→MAR M→MDR→D	DT：R_0→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$) EMAR、R、SMDR、MDR→B、输出 B、DM、CPD、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
ET：D-1→MDR MDR→M PC→MAR	ET：D→A、A-1、DM、CPMDR、T+1、CPT($\overline{P}$) 微命令同前 微命令同前

（12）COM（R_1）+

1）指令流程	2）操作时间表
FT：M→IR PC+1→PC	FT：微命令同前
DT：R_1→MAR M→MDR→D R_1+1→R_1	DT：R_1→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$) EMAR、R、SMDR、MDR→B、输出 B、DM、CPD、T+1、CPT($\overline{P}$) R_1→A、A+1、DM、CPR_1、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
ET：D→MDR MDR→M PC→MAR	ET：D→A、输出 A、DM、CPMDR、T+1、CPT($\overline{P}$) 微命令同前 微命令同前

（13）NEG　-(R_2);

1）指令流程	2）操作时间表
FT：M→IR PC+1→PC	FT：微命令同前
DT：R_2-1→R_2、MAR M→MDR→D	DT：R_2→A、A-1、DM、CPR_2、CPMAR、T+1、CPT($\overline{P}$) EMAR、R、SMDR、MDR→B、输出 B、DM、CPD、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)
ET：D+1→MDR MDR→M PC→MAR	ET：D→A、A+1、DM、CPMDR、T+1、CPT($\overline{P}$) 微命令同前 微命令同前

（14）SL　R_0：

1）指令流程	2）操作时间表
FT：M→IR PC+1→PC	FT：将 1→DT 改为 1→ET，其余微命令同前；

ET：$\bar{R}_0 \rightarrow R_0$　　ET：$\bar{R}_0 \rightarrow$A、输出 A、DM、CPR_0、T+1、CPT($\bar{P}$)，

PC→MAR　　微命令同前。

（15）SR　R_3；

1）指令流程　　2）操作时间表

FT：M→IR　　FT：微命令同前；

PC+1→PC

ET：$\vec{R}_3 \rightarrow R_3$　　ET：$\vec{R}_3 \rightarrow$A、输出 A、DM、CPR_3、T+1、CPT($\bar{P}$)，

PC→MAR　　微命令同前。

（16）JMP　SKP；

1）指令流程　　2）操作时间表

FT：M→IR　　FT：微命令同前

PC+1→PC

ET：PC+1→PC、MAR　ET：PC→A、A+1、DM、CPPC、CPMAR、1→FT、CPT($\bar{P}$)、CPFT($\bar{P}$)、CPST($\bar{P}$)、CPDT($\bar{P}$)、CPET($\bar{P}$)

（17）JMP　R_0；

1）指令流程　　2）操作时间表

FT：M→IR　　FT：微命令同前

PC+1→PC

ET：R_0→PC、MAR　ET：R_0→A、输出 A、DM、CPPC、CPMAR、1→FT、CPT($\bar{P}$)、CPFT($\bar{P}$)、CPST($\bar{P}$)、CPDT($\bar{P}$)、CPET($\bar{P}$)

（18）JMP　X（PC）；

1）指令流程　　2）操作时间表

FT：M→IR　　FT：微命令同前

PC+1→PC

ET：PC→MAR　　ET：PC→A、输出 A、DM、CPMAR、T+1、CPT($\bar{P}$)

M→MDR→C　　EMAR、R、SMDR、MDR→B、输出 B、DM、CPC、T+1、CPT($\bar{P}$)

PC+C→PC、MAR　　PC→A、C→B、A+B、DM、CPPC、CPMAR、1→FT、CPT($\bar{P}$)、CPFT($\bar{P}$)、CPST($\bar{P}$)、CPDT($\bar{P}$)、CPET($\bar{P}$)

转移指令采用相对寻址时，在进行了变址计算后，不再做 PC+1→PC 的操作。这是因为转移指令执行后要转到转移地址去执行目标指令，而不是顺序执行紧跟转移指令的下条指令。

（19）RST　(SP)+;

1）指令流程　　2）操作时间表

FT：M→IR　　FT：微命令同前

PC+1→PC

ET：SP→MAR　　ET：SP→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)

SP+1→SP　　SP→A、A+1、DM、CPSP、T+1、CPT($\overline{P}$)

M→MDR→PC、MAR　　EMAR、R、SMDR、MDR→B、输出 B、DM、CPPC、CPMAR、1→FT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

（20）JSR　(R_1);

转子指令在取指后进入源周期获取子程序入口，再进入执行周期保存断点，并转到子程序入口。

1）指令流程　　2）操作时间表

FT：M→IR　　FT：将 1→ET 改为 1→ST，其余微命令同前

PC+1→PC

ST：R_1→MAR　　ST：R_1→A、输出 A、DM、CPMAR、T+1、CPT($\overline{P}$)

M→MDR→C　　EMAR、R、SMDR、MDR→B、输出 B、DM、CPC、1→ET、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

ET：SP-1→SP、MAR　　ET：SP→A、A-1、DM、CPSP、CPMAR、T+1、CPT($\overline{P}$)

PC→MDR　　PC→A、输出 A、DM、CPMDR、T+1、CPT($\overline{P}$)

MDR→M　　EMAR、W、T+1、CPT($\overline{P}$)

C→PC、MAR　　C→A、输出 A、DM、CPPC、CPMAR、1→FT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

该题也可以采用另一种作法，即取指后进入执行周期，先保存断点，再获取子程序入口并转该入口。相应的指令流程如下：

FT：M→IR

PC+1→PC

ET：SP-1→SP、MAR

PC→MDR

MDR→M

R_1→MAR

M→MDR→PC、MAR

11．拟出中断周期 IT 中各拍的操作时间表

【答】

假设采用模型机中断机制。当 CPU 响应中断后，向中断控制器发出中断响应信号 INTA，并进入中断周期。中断控制器收到 INTA 后，将被批准的中断源的中断号送往数据总线，以便 CPU 接收。在中断周期中，CPU 关中断，保存断点，将中断号乘以 2，作为向量地址访问向量表，获得中断服务程序入口地址后，转入服务程序。

中断周期流程和操作时间表如下：

1）IT 流程　　2）操作时间表

IT：关中断　　IT：0→I

SP-1→SP、MAR　　SP→A、A-1、DM、CPSP、CPMAR、T+1、CPT($\overline{P}$)

PC→MDR　　PC→A、输出 A、DM、CPMDR、T+1、CPT($\overline{P}$)

MDR→M　　EMAR、W、T+1、CPT($\overline{P}$)

8259→MDR→MAR　　IOR、SMDR、MDR→B、输出 B、左移、CPMAR、T+1、CPT($\overline{P}$)

M→MDR→PC、MAR　　EMAR、R、SMDR、MDR→B、输出 B、DM、CPPC、CPMAR、1→FT、CPT($\overline{P}$)、CPFT($\overline{P}$)、CPST($\overline{P}$)、CPDT($\overline{P}$)、CPET($\overline{P}$)

在关中断操作中，发 0→I 命令，将模型机程序状态字 PSW 中的允许中断位 I 清零。在 8259→MDR→MAR 操作中，发 IOR 命令，从 8259 读出中断号，经 MDR、ALU 传送，并将 ALU 输出的结果左移一位，即获得向量地址（中断号乘以 2）。

12．编写取目的地址微子程序（从 60H 单元开始）。

【答】

取目的地址微子程序与取源操作数微子程序类似，所不同的是，前者只取出目的地址，而不取目的操作数。微子程序如表 3-3-2 所示。

表 3-3-2　取目的地址微子程序

	微地址	操作	微命令
取目的地址	60		按目的寻址方式分支，SC=0110
R	61		返回，SC=1000
(R)	62	R_j→MAR	R_j→A，输出 A，DM，CPMAR，返回，SC=1000
–(R)	63	R_j-1→R_j	R_j→A，A-1，DM，CPR$_j$，SC=0000
	64		转 62，SC=0001
(R)+	65	R_j→MAR	R_j→A，输出 A，DM，CPMAR，SC=0000
	66	R_j+1→R_j	R_j→A，A+1，DM，CPR$_j$，返回，SC=1000
@(R)+	67	R_j→MAR	R_j→A，输出 A，DM，CPMAR，SC=0000
	68	M→MDR→MAR	EMAR，R，SMDR，MDR→B，输出 B，DM，CPMAR，SC=0000
X(R)	69		转 66，SC=0001
	6A	PC→MAR	PC→A，输出 A，DM，CPMAR，SC=0000
	6B	PC+1→PC	PC→A，A+1，DM，CPPC，SC=0000
	6C	M→MDR→D	EMAR，R，SMDR，MDR→B，输出 B，DM，CPD，SC=0000
	6D	R_j+D→MAR	R_j→A，D→B，A+B，DM，CPMAR，返回，SC=1000

13、根据教材中表 3-18 微程序，以微地址序列形式（如 00-01-02-0C…），拟出下述指令的读出与执行过程。

（1）MOV（R_0），（SP）+；

（2）MOV（R_1）+，X（R_0）；

（3）ADD　X（R_0），R_1；

（4）SUB（R_1）+，（R_2）；

（5）NEG　–（R_2）；

（6）JMP（R_0）；

（7）JSR　R_1；

【答】

注意 MOV 指令和双操作数指令中源地址与目的地址的位置，源地址在后，目的地址在前。

（1）MOV（R_0），（SP）+；

00-01-02-03-4C-52-53-54-4F-04-60-62-05-06-07-08-09-00

解释微地址序列：先取指（00-01-02），再转 MOV 指令入口（03），然后转“取源操作数”入口（4C），进入自增型寄存器间址（52-53-54-4F），返回（04），再转“取目的地址”入口（60），进入寄存器间址（62），返回（05），最后执行传送操作并转取指入口（06-07-08-09-00）。

（2）MOV（R_1）+，X（R_0）；

00-01-02-03-4C-59-5A-5B-5C-5D-4F-04-60-65-66-05-06-07-08-09-00

解释微地址序列：取指（00-01-02），再转 MOV 指令入口（03），然后转“取源操作数”入口（4C），进入变址（59-5A-5B-5C-5D-4F），返回（04），再转“取目的地址”入口（60），进入自增型寄存器间址（65-66），返回（05），最后执行传送操作并转取指入口（06-07-08-09-00）。

（3）ADD　X（R_0），R_1；

00-01-02-0C-4C-4D-0D-60-6A-6B-6C-6D-0E-0F-10-11-07-08-09-00

解释微地址序列：ADD 是双操作数指令，因此取指后进入其入口（0C），转“取源操作数”入口（4C），进入寄存器寻址（4D），返回（0D），再转“取目的地址”入口（60），进入变址（6A-6B-6C-6D），返回（0E-0F），最后执行加法操作并转取指入口（10-11-07-08-09-00）。

（4）SUB（R_1）+，（R_2）；

00-01-02-0C-4C-4E-4F-0D-60-65-66-0E-0F-14-15-07-08-09-00

解释微地址序列：取指（00-01-02），进入双操作数指令入口（0C），转“取源操作数”入口（4C），进入寄存器间址（4E-4F），返回（0D），再转“取目的地址”入口（60），进入自增型寄存器间址（65-66），返回（0E-0F），最后执行减法操作并转取指入口（14-15-07-08-09-00）。

（5）NEG　-（R_2）；

00-01-02-24-60-63-64-62-25-26-2B-2C-07-08-09-00

解释微地址序列：NEG 是单操作数指令，取指后进入其入口（24），转“取目的地址”入口（60），进入自减型寄存器间址（63-64-62），返回（25-26），最后执行求补操作并转取指入口（2B-2C-07-08-09-00）。

（6）JMP（R_0）；

00-01-02-3F-43-4C-4E-4F-44-45-08-09-00

解释微地址序列：JMP 是转移指令，取指后进入其入口（3F-43），转“取源操作数”入口（4C），进入寄存器间址（4E-4F），返回（44-45），将转移地址送入 MAR 并转取指入口（08-09-00）。

（7）JSR　R_1；

00-01-02-3F-46-48-49-4A-4B-47-43-4C-4D-44-45-08-09-00

解释微地址序列：JSR 是转子指令，取指后进入其入口（3F-46），转“压栈”入口，将转子时的返回地址压入堆栈保存（48-49-4A-4B），返回（44-45），将子程序入口地址送入 MAR 并转取指入口（08-09-00）。

14．如果以 R_3 为堆栈指针，软件建立堆栈，试分别编写压栈及弹出操作的子程序。

【答】

假设栈顶单元地址为 1000H，压栈和弹出之前，用立即寻址方式将 1000H 送入堆栈指针 R_3，即执行指令“MOV　R_3, (PC)+”（该指令所在单元的下一单元存放 1000H）。之后，将需要压栈的数据所在单元的地址送入某一寄存器，如 R_0，再调用压栈子程序，该子程序包含一条传送指令：

MOV －(R_3), (R_0);　　将 R_0 所指示的存储单元中的数据压入堆栈。

若需要弹出操作，则调用弹出子程序，该子程序也包含一条传送指令：

MOV　(R_1), (R_3)+;　将数据从堆栈弹出，送入 R_1 所指示的存储单元。

15．如果将 CPU 时钟周期与访存周期分开设置，一个访存周期占用四个时钟周期，试重新设计模型机指令流程。

【答】

一个访存周期占用四个时钟周期，可在第一个时钟周期中传送地址，即 CPU 将地址从 MAR 送往存储器；在第二个时钟周期中向存储器发读令或写令；第三个时钟周期从存储器读出数据或向存储器写入数据；第四个时钟周期结束存储器访问，即 CPU 撤消地址、命令、数据。

下面，以加法指令 ADD（R_1），R_0；为例，重新拟定模型机指令流程。

FT0：MAR→M　－进入 FT，将现行指令地址经地址总线送往存储器。
FT1：读令→M　－将读命令经控制总线送往存储器。
FT2：M→IR　－从存储器读出指令，经数据总线送往 IR。
FT3：PC+1→PC　－撤消总线上的地址、命令、指令，并修改 PC，结束 FT。
DT0：R_1→MAR　－进入 DT，将 R_1 内容送往 MAR。
DT1：MAR→M　－将目的地址经地址总线送往存储器。
DT2：读令→M　－将读命令经控制总线送往存储器。
DT3：M→MDR　－从存储器读出目的操作数，经数据总线送往 MDR。
DT4：MDR→D　－撤消总线上的地址、命令、数据，并将目的操作数从 MDR 送往暂存器 D，结束 DT。
ET0：R_0+D→MDR　－进入 ET，将源操作数和目的操作数相加，结果送往 MDR。
ET1：MAR→M　－将目的地址经地址总线送往存储器。
ET2：写令→M　－将写命令经控制总线送往存储器。
ET3：MDR→M　－将运算结果经数据总线写入存储器。
ET4：PC→MAR　－撤消总线上的地址、命令、数据，并将下条指令的地址送往寄存器 MAR。

16．如果将模型机内部数据通路结构改为教材上图 3-40 的结构，试重新设计模型机。

【答】

通用寄存器组采用存储器结构，输入逻辑采用锁存器，内总线采用双向形式，其他寄存器均挂接在内总线上。

模型机数据通路结构框图如图 3-3-2 所示。

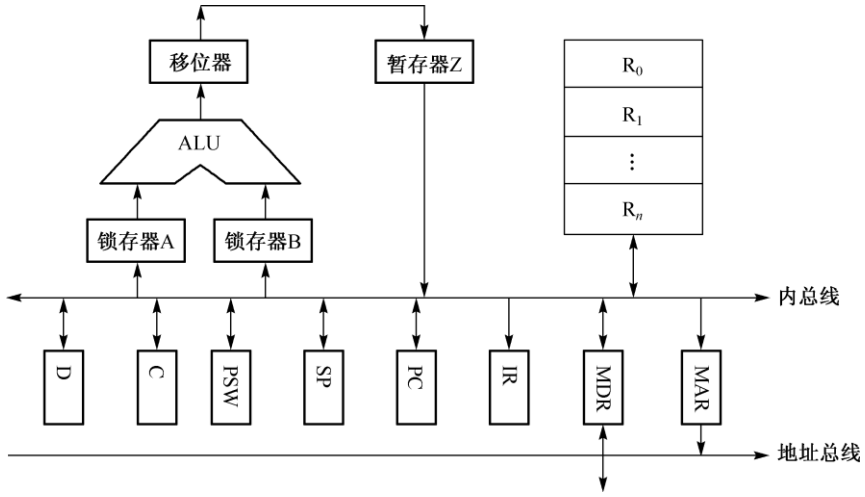


图 3-3-2　新设计的模型机

17．结合 3.5.3 节 MIPS 单周期 CPU，写出下列指令的操作及其对应的微命令。

【答】假设 XOR 运算对应的编码是 1101

（1）XOR rd, rs, rt

$rs xor $rt→$rd：PCSrc=00; regdst=0; aluSrc=0; operation=1101; memR=memW=0; extend=0/1; mem2Reg=0; Regwrite=1;

（2）XORI rt, rs, imm

$rs xor E(imm)→$rd：PCSrc=00; regdst=1; aluSrc=1; operation=1101; memR=memW=0; extend=0; mem2Reg=0; Regwrite=1;

（3）BNE rs, rt, imm

$rs-$rt，若 zero=0 表明两者不相等，则 PC+4+E(imm)<<2→PC；若 zero=1 表明两者相等则 PC+4→PC;

① PCsrc=01(Zero=0 时); regdst=0/1; alusrc=0; operation=0110; memR=memW=0; extend=1; mem2Reg=0/1; Regwrite=0;

② PCsrc=00(Zero=1 时); regdst=0/1; alusrc=0; operation=0110; memR=memW=0; extend=1; mem2Reg=0/1; Regwrite=0;

18．结合 3.5.4 节的 MIPS 多周期 CPU，….

【答】假设 XOR 运算对应的编码是 1101

（1）XOR rd, rs, rt

T0：T0 IR←Mem[PC], PC←PC+4;

//IorD=0, MemRead=1, MemWrite=0, IRWrite=1, ALUSrc_A=0, ALUSrc_B=00, operation=0010, PCSrc=00, PCWrite=1, extend=0, regWrite=0, regdst=0, mem2reg=0;

T1： A←Reg[rs], B←Reg[rt] clock 信号触发

T2： F←A xor B

//IorD=0, MemRead=0, MemWrite=0, IRWrite=0, **ALUSrc_A=1**, **ALUSrc_B=01**, **operation=1101**, PCSrc=00, PCWrite=0, extend=0, regWrite=0, regdst=0, mem2reg=0;

T3 ：Reg[rd]←F

//IorD=0, MemRead=0, MemWrite=0, IRWrite=0, ALUSrc_A=0 ALUSrc_B=00, operation=0000, PCSrc=00, PCWrite=0, extend=0, **regWrite=1**, **regdst=0**, **mem2reg=0;**

（2）XORI rt, rs, imm

T0： IR←Mem[PC], PC←PC+4;

//IorD=0, MemRead=1, MemWrite=0, IRWrite=1, ALUSrc_A=0, ALUSrc_B=00, operation=0010, PCSrc=00, PCWrite=1, extend=0, regWrite=0, regdst=0, mem2reg=0;

T1 A←Reg[rs] clock 信号触发

T2 F←A xor E(imm)

// ALUSrc_A=1, extend=1, ALUSrc_B=10, operation=0010

//IorD=0, MemRead=0, MemWrite=0, IRWrite=0, **ALUSrc_A=1**, **ALUSrc_B=10**, **operation=1101**, PCSrc=00, PCWrite=0, **extend=0**, regWrite=0, regdst=0, mem2reg=0;

T3 Reg[rd]←F

//IorD=0, MemRead=0, MemWrite=0, IRWrite=0, ALUSrc_A=0 ALUSrc_B=00, operation=0000, PCSrc=00, PCWrite=0, extend=0, **regWrite=1**, **regdst=0**, **mem2reg=0;**

（3）BNE rs, rt, imm

T0：IR←Mem[PC], PC←PC+4;

//IorD=0, MemRead=1, MemWrite=0, IRWrite=1, ALUSrc_A=0, ALUSrc_B=00, operation=0010, PCSrc=00, PCWrite=1, extend=0, regWrite=0, regdst=0, mem2reg=0;

T1 A←Reg[rs], B←Reg[rt], F←PC+E(offset)<<2

//ALUSrc_A=0, ALUSrc_B=11, operation=0010, extend=1

//IorD=0, MemRead=0, MemWrite=0, IRWrite=0, **ALUSrc_A=0**, **ALUSrc_B=11**, **operation=0010**, PCSrc=00, PCWrite=0, **extend=1**, regWrite=0, regdst=0, mem2reg=0;

T2 A-B；PC←F

//IorD=0, MemRead=0, MemWrite=0, IRWrite=0, **ALUSrc_A=1**, **ALUSrc_B=01**, **operation=0110**, extend=0, regWrite=0, regdst=0, mem2reg=0;

①PCSrc=00（zero=1）, PCWrite=0

②PCSrc=01（zero=0）, PCWrite=1

19．某 MIPS 架构的多周期 CPU 执行一段程序，指令分布情况如下：…

【答】

（1）平均 CPI

CPI=4T×0.45+5T×0.25+4T×0.15+3T×0.1+2T×0.05=4.05

（2）平均 IPS

IPS=(1÷(100×10－12))÷4.05≈2.47×109

（3） 数据输出通路的带宽。

R=2KB÷(100×4.05×100×10－12)≈50GB/S

20．某 32 位 MIPS 型计算机，其存储器按字编址，存储片段如下：…

【答】

（1）add 指令执行后，PC 寄存器和 rd 寄存器中的内容分别是什么？

PC=00000004H, $rd=30

（2）lw 指令执行后，PC 寄存器和 rt 寄存器中的内容是什么？

PC=00000008H, $rt=000000AH

（3）beq 指令执行后，PC 寄存器的内容是什么？

PC=0000001CH

21．[2014 考研全国统考真题] 某程序中有如下一段循环代码段 P：…

【答】

（1）计算机 M 的存储器编址单位为字节，因为每个 32 位定长指令字占用了 4 个存储单元，故每一个编址单元为一个字节共 8 比特。

（2）因为 sll 指令每次左移 2 位（相当于乘以 4）。R2 的初值为 0，连续两个元素的地址刚好相差 4，而存储器又是以字节遍址的，故每个元素占用 4B。

（3）根据 bne 指令格式知 offset 字段为低 16 位，指令对应的机器代码为 1446FFFA，故 offset 字段的代码为 FFFA=1111 1111 1111 1010，补码表示，故 offset 的真值=-6。

bne 指令地址为 08048114H，bne 指令执行后 PC+4=08048114H+4H=08048118H，分支的目标地址=08048100H，而 08048118H+n×offset=08048100H，

则 n=(08048100H-08048118H=-18H=-24) ÷ (-6)=4，所以 bne 指令转移目标地址=(PC)+4+4×offset

22．简述指令流水线的工作原理。

【答】

为提高处理器执行指令的效率，把一条指令的操作过程分成若干个子过程，且每个子过程都在专门功能电路上完成，这样指令的各子过程就能同时运行，指令的平均执行时间也能大大减少。

23．试比较超标量和超流水的异同点。

【答】

超标量流水线值在一个处理器中针对同一种功能，设置多条并存的流水线。在每个时钟周期中可向流水线发射多条同一类指令，也能从流水线中流出多个处理结果。

超流水线以增加流水线级数的方法来缩短机器周期，使相同时间内超流水线能执行更多的机器指令。

3.4 第 4 章习题解答

1、简要解释下列名词述语

虚拟存储器：依靠操作系统的支持来实现的，为用户提供一个比实际内存大的可访问存储器空间，即在软件编程上可使用的存储器，称为虚拟存储器。

随机存储器 RAM：按给定地址随机地访问任一存储单元，访问时间与单元位置无关。

只读存储器 ROM：在正常工作中只能读出，不能写入的存储器。

存取周期：指存储器做连续访问操作过程中一次完整的存取操作所需的全部时间。

数据传输率：是数据传入或传出存储器的速率。

动态刷新：对动态存储器中原存信息为 1 的电容补充电荷，称为动态刷新。

直接映像 Cache：将主存与 Cache 的存储空间划分为若干大小相同的页，每个主存页只能复制到某一个固定的 Cache 页中。

全相联映像 Cache：将主存与 Cache 的存储空间划分为若干大小相同的页，主存的每一页可以映象到 Cache 的任一页上。

组相联映像 Cache：将主存与 Cache 都分组，主存中一个组内的页数与 Cache 的分组数相同。每一组 Cache 中含有若干页（一般页数较少）；则主存中的各页与 Cache 的固定组号有映象关系，可自由映象到对应的 Cache 组中任一页。

段页式虚拟存储器：将程序按其逻辑结构分段，每段再分为若干大小相同的页，主存空间也划分为若干同样大小的页。相应地建立段表与页表，分两级查表实现虚实地址的转换。以页为单位调进或调出主存，按段共享与保护程序及数据。

相联存储器：是一种按内容寻址的存储器，它是根据所存信息的全部特征或部分特征进行存取的存储器，称为相联存储器。

2．请简计算机系统中的三级存储体系结构模式，并分析这种模式的优点和缺点。

答：三级存储体系包括缓存（cache）、内存和外存，这种模式的优点是层次体系清晰、便于设计实现，也利于系统调度管理，能提高存储系统性能；缺点是结构复杂，管理和控制都比较复杂，硬件成本高。

3．何谓随机存取？何谓顺序存取？何谓直接存取？请各举一例进行说明。

答：随机存取是指按地址码访问存储器，访存时间和存储单元的位置无关，比如内存；顺序存取是指按存储介质的存储顺序移动介质、定位存储区并读写数据，比如磁带；直接存取是指先寻道再定位扇区，然后再读写数据，比如硬盘。

4、某半导体存储器容量 8K×8 位，可选 RAM 芯片容量为 2KX4/片。地址总线 A15～A0（低），双向数据线 D7～D0（低），由 R / $\overline{W}$ 线控制读写。请设计并画出该存储器逻辑图，并注明地址分配与片选逻辑式及片选信号极性。

解题分析：设计存储器需要由总容量确定可选存储器芯片的数量，由于每片芯片容量通常低于总量，要用若干块芯片组成。这样需要在位数和单元数上进行扩展。其次要考虑

如何连接有关存储芯片的地址线、片选信号线、数据线和控制信号线。

存储器由若干存储芯片组成，每个芯片包含若干存储单元。地址分配的原则是将存储器的低位地址分配给存储芯片，以选择芯片内的存储单元；高位地址分配给片选逻辑，译码后产生片选信号，选择某个存储芯片。

解：（1）确定芯片数量：本题存储容量为 8K×8 位，可选的芯片容量为 2K×4/片。在构成存储器时，可选进行位扩展，由两块 2K×4/片的芯片为一组，构成 2K×8 的存储模块。由四组 2K×8 的存储模块构成 8K×8 的存储器。

位扩展：　　2K×4/片 × 2 → 2K×8

单元扩展：　2K×8　　×4 → 8K×8

所以共需芯片数量为：2 × 4 = 8 片

（2）地址分配和片选逻辑设计

本题存储容量为 8K 字节单元，占 16 位地址总线的低 13 位 A_{12}～A_0。用这 13 位地址可寻址整个 8K 存储空间，接下来再对这 13 位地址进行分配。

每块芯片容量为 2K×4/片，需 11 位地址对片内单元寻址。组成 8KB 存储器需 4 组芯片，所以需用 4 个片选信号对它们进行选择。因此将 13 位地址中的低 11 位地址 A_{10}～A_0 分配给各存储芯片，加至各芯片的地址端，剩下的高两位地址 A_{12}、A_{11} 送片选逻辑译码，产生四个片选信号 CS_0、CS_1、CS_2、CS_3，这四个片选信号的逻辑式为：

$$CS_0 = \overline{A}_{12}\overline{A}_{11}$$
$$CS_1 = \overline{A}_{12}A_{11}$$
$$CS_2 = A_{12}\overline{A}_{11}$$
$$CS_3 = A_{12}A_{11}$$

（3）画连接图，需要连接的信号线包括地址线，数据线，控制线和片选信号线，其连接方法为：

芯片地址线连接至：A_{10}～A_0

数据线：一组连接至高四位 D_7～$\overline{D_4}$

　　一组连接至低四位 D_3～$\overline{D_0}$

片选信号分开连，控制信号线 $R/\overline{W}$ 连接至每块芯片，其具体连接如图 3-4-1 示：

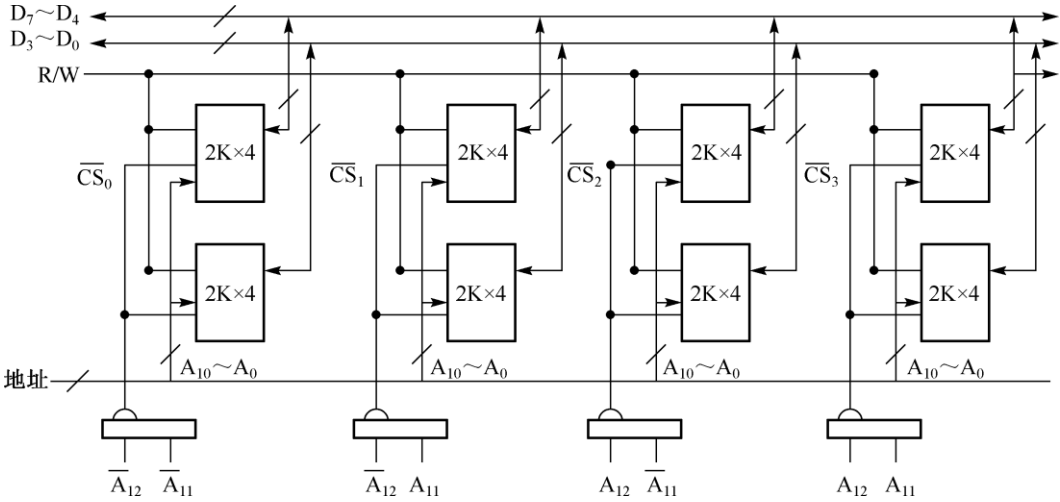


图 3-4-1 存储器逻辑图

5、某半导体存储器容量 7K×8 位。其中固化区 4K×8，可选 EPROM 芯片：2K×8/片。随机读写区 3K×8，可选 SRAM 芯片：2K×4/片、1K×4/片。地址总线 A_{15}～A_0（低），双向数据总线 D_7～D_0（低），R / $\overline{W}$ 控制读写。另有控制信号 $\overline{MREQ}$，低电平时允许存储器工作。请设计并画出该存储器逻辑图，并注明地址分配与片选逻辑式及片选信号极性。

解题分析：

（1）芯片选择

需两块 2K×8/片的 EPROM 芯片进行单元扩展，构成 4K×8 的固化区。2 块 2K×4/片及 2 块 1K×4/片的 SRAM 芯片进行位扩展和单元拼接构成 3K×8 的随机读写区。

（2）地址分配及片选逻辑设计：7K 字节单元占 16 位地址线的低 13 位 A_{12}～A_0。

连接至各芯片的地址分别为：

2K×8 的 EPROM 占 11 位地址线：A_{10}～A_0

2K×4 的 SRAM 占 11 位地址线：A_{10}～A_0

1K×4 的 SRAM 占 10 位址线：A_9～A_0

组成 7 K 的存储器共有 4 组芯片，2 组 2 K×8 的 EPROM、1 组 2 块 2 K×4 的 SRAM、1 组 2 块 1 K×4 的 SRAM 芯片，2 K 单元的芯片用剩下的高两位地址 A12、A11 送片选逻辑译码，产生片选信号 CS0、CS1、CS2，1 K 单元的芯片用剩下的高三位地址 A12、A11、A10 送片选逻辑译码，产生片选信号 CS3，这 4 个片选信号的逻辑式为：

$$CS_0 = \overline{A}_{12}\overline{A}_{11}$$

$$CS_1 = \overline{A}_{12}A_{11}$$

$$CS_2 = A_{12}\overline{A}_{11}$$

$$CS_3 = A_{12}A_{11}\overline{A}_{10}$$

该存储器逻辑图见图 3-4-2：

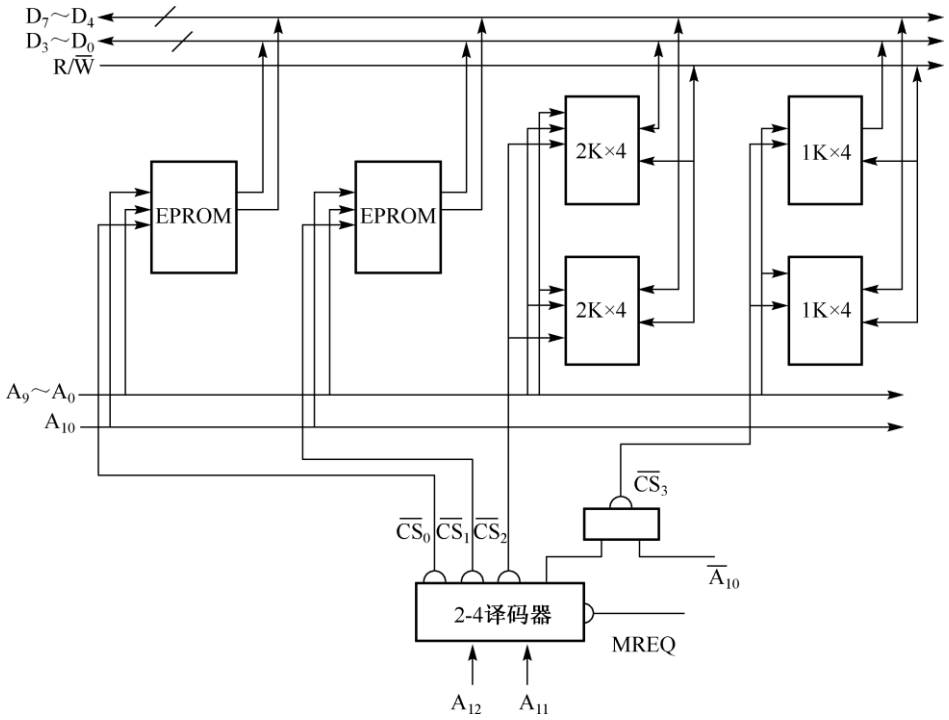


图 3-4-2　存储器逻辑图

6．某机地址总线 16 位 A_{15}～A_0（低），访存空间为 64 KB。外围设备与主存统一编址，即将外围设备接口中有关寄存器与主存单元统一编址，I/O 空间占用 FC00H～FFFFH。现用 2164 构成主存储器，请设计并画出该存储器逻辑图，并画明芯片地址线与总线的连接逻辑，以及行选信号与列选信号的逻辑式，使访问外设时不访问主存。

【答】

由 I/O 所占用的地址空间为 FC00～FFFF，即

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	…	A_0
1	1	1	1	1	1	0	…	0
							⋮	
1	1	1	1	1	1	1	…	1

可知，I/O 占用了 10 位地址空间，所以 I/O 的单元数为 1KB，因此主存的地址空间为 63 KB。2164 为 64 K×1 位容量的芯片，因此需 8 块 2164 芯片构成主存储器。

每片 2164 提供一位数据送数据总线，构成 8 位数据，64K 单元需要 16 位地址线寻址，由于 2164 只有 8 位地址线，所以采用地址的分时复用技术。16 位地址分两次送入芯片，先送入行地址，再送入列地址，相应地需要行选择信号 $\overline{RAS}$ 和列选择信号 $\overline{CAS}$ 来标明当前送入的是行地址还是列地址。

由于 I/O 与主存空间统一编址，占用 64 K 地址空间的最后 1 K 地址空间，因此当高 6 位地址为全 1 时，应选中 I/O 地址空间，否则访问主存空间，而行选信号和列选信号的逻辑为 $RAS=CAS=A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}$。

该存储器逻辑图见图3-4-3：

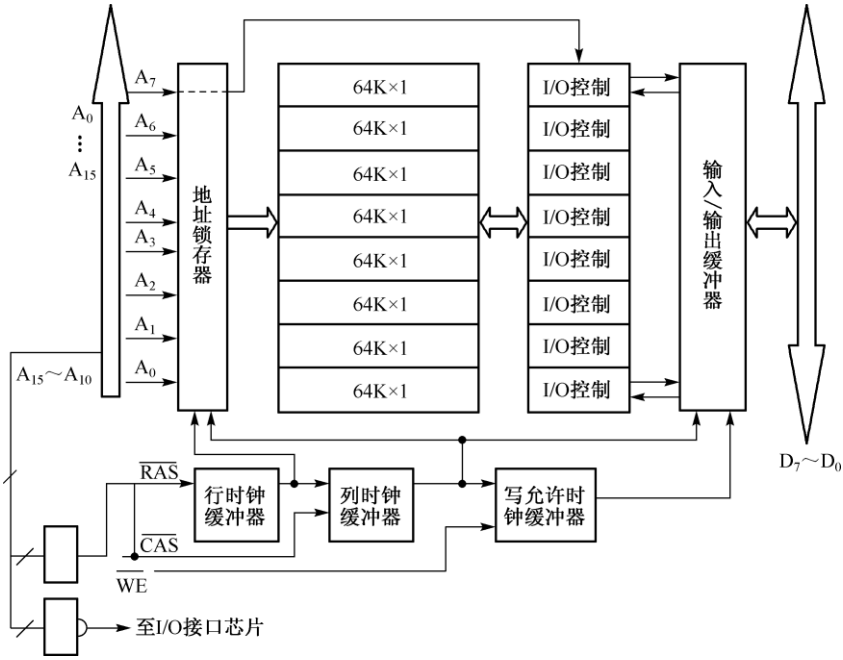


图 3-4-3　存储器逻辑图

7．某半导体存储器容量为 14 KB，其中 0000H～1FFFFH 为 ROM 区，2000H～37FFH 为 RAM 区，地址总线 A_{15}～A_0（低），双向数据总线 D_7～D_0（低），读写控制线 $R/\overline{W}$。可选用的存储芯片有 EPROM（4 KB /片）和 RAM（2 K×4 /片）。

（1）计算所需各类芯片的数量。

（2）说明加到各芯片的地址范围值和地址线。

（3）写出各片选信号的逻辑式。

（4）画出该存储芯片级逻辑图，包括地址总线、数据线、片选信号线（低电平有效）及读写信号线的连接。

【答】

（1）芯片数量：

ROM 区容量=（1FFFH-0000H+1）÷2^{10}=8K，故 EPROM 要 2 片；

RAM 区容量=（37FFH -2000H +1）÷2^{10}=6K，故 RAM 要 6 片；

（2）各组芯片的地址范围和地址线分布情况如表 3-4-1 所示：

表 3-4-1　各组芯片的地址分布范围

	$A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9\ A_8A_7\ldots A_0$
EPROM　4K×8	0 0 0 0 0 0 0 0 0…0
	0 0 0 0 1 1 1 1 1…1
EPROM　4K×8	0 0 0 1 0 0 0 0 0…0

		0	0	0	1	1	1	1	1	1…1
RAM 2K×4	RAM 2K×4	0	0	1	0	0	0	0	0	0…0
		0	0	1	0	0	1	1	1	1…1
RAM 2K×4	RAM 2K×4	0	0	1	0	1	0	0	0	0…0
		0	0	1	0	1	1	1	1	1…1
RAM 2K×4	RAM 2K×4	0	0	1	1	0	0	0	0	0…0
		0	0	1	1	0	1	1	1	1…1

第 0 组，4K 地址空间，故片内地址线 12 根：A_{11}-A_0

第 1 组，4K 地址空间，故片内地址线 12 根：A_{11}-A_0

第 2 组，2K 地址空间，故片内地址线 11 根：A_{10}-A_0

第 3 组，2K 地址空间，故片内地址线 11 根：A_{10}-A_0

第 4 组，2K 地址空间，故片内地址线 11 根：A_{10}-A_0

（3）各组芯片的片选逻辑表达式

$\overline{CS_0} = \overline{A_{15}}\overline{A_{14}}\overline{A_{13}}\overline{A_{12}}$，

$\overline{CS_1} = \overline{A_{15}}\overline{A_{14}}\overline{A_{13}}A_{12}$，

$\overline{CS_2} = \overline{A_{15}}\overline{A_{14}}A_{13}\overline{A_{12}}\overline{A_{11}}$，

$\overline{CS_3} = \overline{A_{15}}\overline{A_{14}}A_{13}\overline{A_{12}}A_{11}$，

$\overline{CS_4} = \overline{A_{15}}\overline{A_{14}}A_{13}A_{12}\overline{A_{11}}$

因为明确规定了 16 位的地址码，因此片选信号线中应接入 A_{15} 和 A_{14}。

（4）逻辑结构图如图 3-4-4 所示。

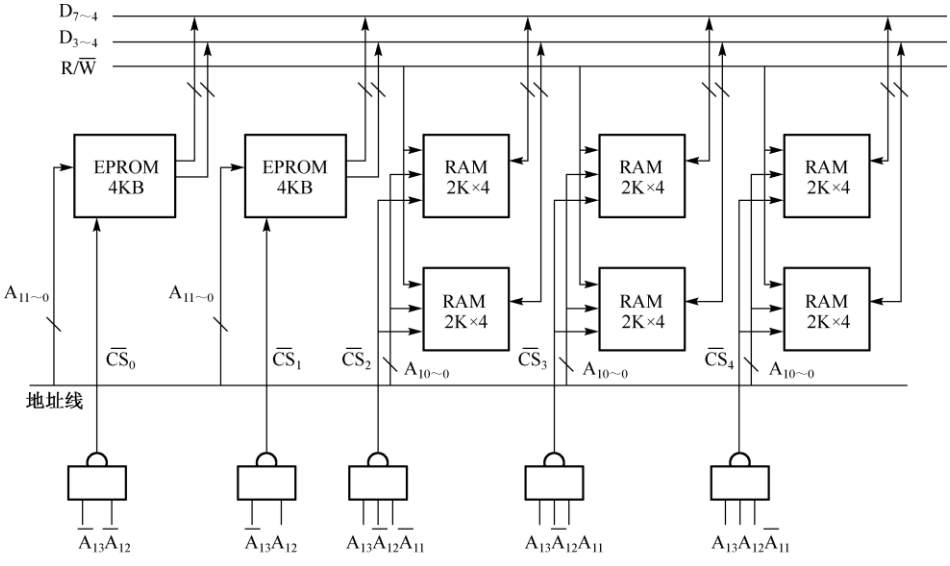


图 3-4-4　逻辑结构图

8．SRAM 和 DRAM 分别依靠什么原理存储信息？需要刷新吗？

【答】

SRAM 即静态半导体存储器，静态半导体存储器中的一位存储单元，实际上是一个双稳态触发器，靠交叉反馈存储信息。

DRAM 即动态半导体存储器则是靠电容电荷存储信息，如电容上有电荷代表存放的信号 1，电容无电荷为存放的信号 0。

SRAM 不需要刷新，而 DRAM 则需要刷新。

9. 设某机主存 1MB，用 1MB/片的 DRAM 芯片构成，芯片最大刷新周期 2ms，问在 2ms 之内至少应该安排几个刷新周期？

【答】

由于动态存储器的刷新是按行进行的，每一行中的位单元同时进行刷新，所以刷新不是按单元地址进行的，而是按行地址进行，存储芯片阵列的行数是多少就需要在最大刷新周期中安排多少个刷新周期。

1MB/片的 DRAM 芯片存储器若为 1024×1024 矩阵，那么在 2ms 之内至少应该安排 1024 个刷新周期。

10．某机主存容量 64KB，用 2164DRAM 芯片构成。地址线 A15～A0（低），双向数据线 D7～D0（低），R/W 控制读写操作。请设计并画出该存储器逻辑图。对地址的行、列转换与数据线连接，应有明确描述。

【解题分析】

2164 为 64K×1 位的存储芯片，要构成 64KB 容量的存储器，需要用 8 片 2164 做位拼接，每片提供一位数据，分别连接至双向数据线 D_7 ～D_0 上。

【答】

该存储器逻辑图见图 3-4-5。

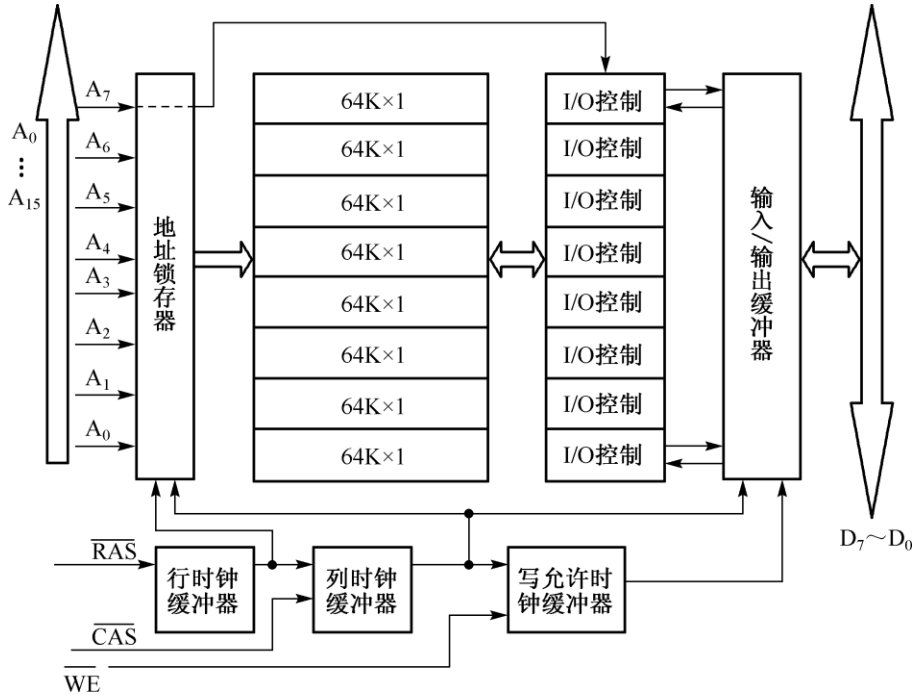


图 3-4-5　存储器连接逻辑图

11．某机主存容量 64KB，用 2164DRAM 芯片构成。请为此设计一种动态刷新逻辑。

【解题分析】

2164 为 64 K×1 位的芯片，芯片内部构成 4 个 128×128 的矩阵，由动态存储器刷新按行安排可知，该动态存储器在 2 ms 的时间间隔内需安排 128 个刷新周期。如以异步方式刷新，则每个刷新周期的时间间隔为：2 ms/128 = 15.625 μs。

异步刷新的逻辑如图所示，刷新周期由一个单稳和一个非门组成的振荡器产生。该振荡器每隔 15.625 μ S 产生一个脉冲信号，定时触发刷新请求触发器，于是每隔 15.625μs 产生一个刷新请求信号，其电路如图 3-4-6 所示。该刷新请求信号触发器行计数器输出行地址，将该行所有位单元刷新。

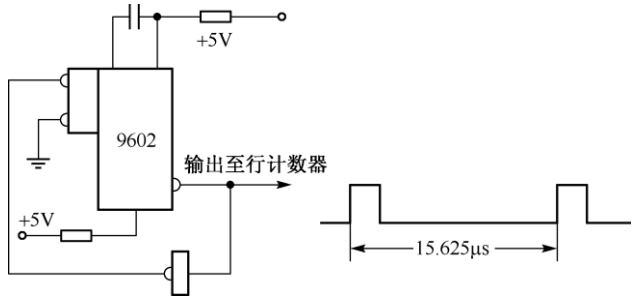


图 3-4-6　动态刷新逻辑

12．动态刷新周期的安排方式有哪几种？简述它们的安排方法，并指出在下列情况中可选取哪一种或几种刷新方式。

a）教学用单板计算机。

b）常用个人计算机。

c）带多个分时终端的超小型计算机。

d）如果主存的存取周期 200 ns，CPU 平均访存时间约占主存工作时间的 90%。

e）主存的存取周期 200 ns，CPU 平均访存时间约占主存工作时间的 40%。

【答】

动态刷新周期的安排方式有以下 3 种：

① 集中刷新方式：在要求的时间间隔内集中安排所有刷新周期，其余时间用于正常访存工作。

② 分散刷新方式：将各刷新周期分散的安排在各存取周期之后，即将每个存取周期分为两部分，前半期用于正常读/写或保持，后半期用于刷新。

③ 异步刷新方式：按芯片阵列的行数决定所需要的刷新周期数，并将各刷新周期分散安排在刷新要求的时间间隔周期中，每隔固定时间提出一次刷新请求，安排一个刷新周期。

题目中所述各种情况可选择的刷新方式如下。

① 教学用单板计算机：由于该计算机是为了教学用，主要让学生理解刷新的基本原理和实现方法，所以该类计算机可选择三种刷新方式中的任何一种。

② 常用个人计算机：采用异步刷新，由于集中刷新会有明显的死区，而分散刷新只适用于低速的系统，都不适合于个人计算机，异步刷新是大多数计算机系统采用的方式。

③ 带多个分时终端的超小型计算机：可采用分散刷新方式，可将刷新安排在各分时终端的时间片中。

④ 如果主存的存取周期 200 ns，CPU 平均访存时间约占主存工作时间的 90%：采用异步刷新。由于 CPU 平均访存时间要占主存工作时间的 90%，集中刷新死区过长，而分散刷新的次数太多，占用主存的时间太长，只有异步刷新方式对主存工作的时间占用最少。

⑤ 主存的存取周期 200 ns，CPU 平均访存时间约占主存工作时间的 40%：可采用集中刷新的方式。由于 CPU 平均访存时间只占主存工作时间的 40%，可在剩余的时间内安排集中刷新，这样可使刷新电路简单易于设计。

13．若对磁表面存储器写入代码 10011，请画出 NRZ-1 制、PE 制、FM 制、M^2F 制等记录方式的写电流波形。

【答】

（1）NRZ-1 制的写入规律为：写 0 时，写入电流维持原方向不变，写 1 时，写入电流方向翻转。其写电流波形如图3-4-7 所示。

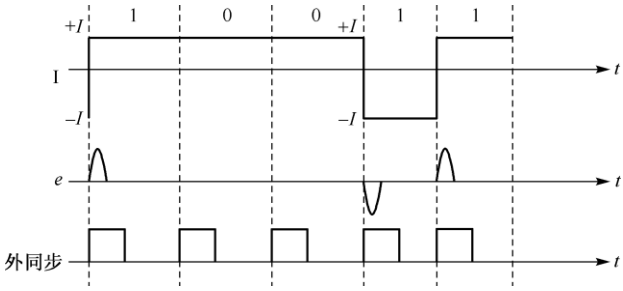


图 3-4-7　NRZ-1 制写电流波形

（2）调相制的写入规律为：写 0，在位单元中间位置让写入电流负跳变，由+*I*→–*I*；写 1，在位单元中间位置让写入电流正跳变，由–*I*→+*I*。其波形如图3-4-8 所示。

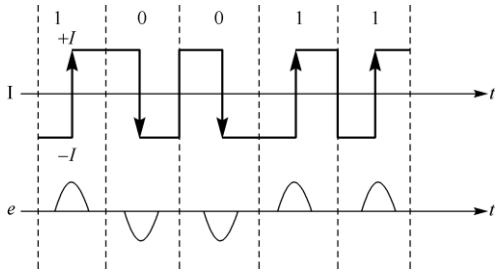


图 3-4-8　调相制写电流波形

（3）FM 制的写入规律为：在位单元中间记录数据信息。如果写入 0，则位单元中间不变。如果写入 1，则写入电流在位单元中间改变一次方向。其波形如图3-4-9 所示。

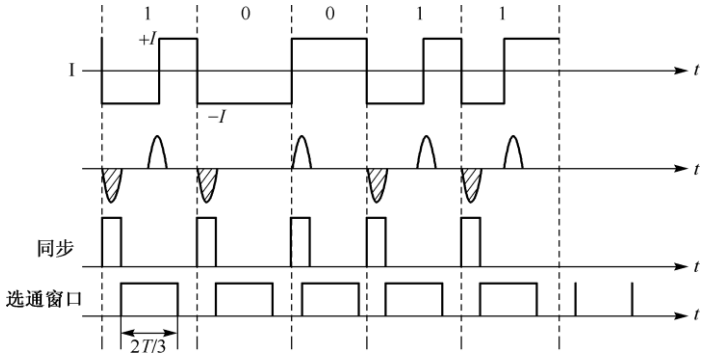


图 3-4-9　FM 制写电流波形

（4）M²F 改进型调频制的写入规律为：写 1 时，在位单元中间改变写入电流方向；写入两个以上 0 时，在它们的交界处改变写入电流方向。其波形如图3-4-10 所示。

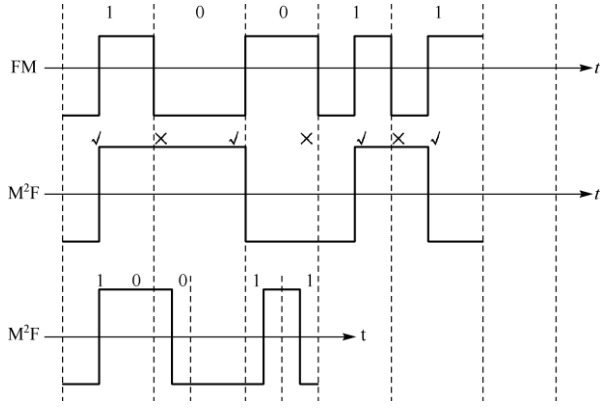


图 3-4-10　M²F 制写电流波形

14．某磁盘有 100 个柱面，每个柱面有 10 个磁道，每个磁道有 128 个扇区，每个扇

区容量为 512 字节。该磁盘的存储容量是多少？

【答】

100 个圆柱面，表明单面有 100 个磁道；

每个柱面有 10 个磁道，表明共有 10 个记录面；

则磁盘容量=100×10×128×512B

=64000KB

15．某计算机字长为 32 位，CPU 主频为 500 MHz，磁盘共有 16 个盘面，512 个柱面，每磁道包含 100 个扇区，每个扇区 512 字节，该磁盘旋转速度为 12000 RPM。

（1）计算该磁盘总容量？

（2）计算该磁盘的数据传输率（bit/s）？

【答】

（1）总容量=16×512×100×512B=400MB；

（2）数据传输率即带宽，表示磁盘单位时间内的数据传输量

磁盘转速=12000/60s

=200RPS（转/秒)

磁盘带宽=(100×512×8) ×200

=80Mb/s

16．磁盘的磁道和光盘依靠什么原理来记录信息，磁道和光道有何不同？

【答】

磁盘依靠磁表面材料的磁化特征来记录信息，光盘依靠涂层材料的形变、相变或者热磁效应来记录信息。磁道是分布均匀的同心圆，光道是类似蚊香的放射状螺旋线形。

17．双端口存储器与两个独立存储器有何不同？

【答】

双端口存储器和两个独立的存储器虽然都具有两个彼此独立的读 / 写口，每个读 / 写口都有一套独立的地址寄存器和译码电路，可以并行地独立工作，并且两个读 / 写口可以按各自接收的地址，同时读出或写入，或一个写入而另一个读出。但双端口存储器与两个独立的存储器不同的是，两套读 / 写口的访存空间相同，可以访问同一区间、同一单元；而两个独立的存储器的访存空间也是彼此独立的。

18．何谓单体多字并行主存系统？何谓多体交叉存取并行主存系统？

【答】

单体多字并行主存系统即是多个并行存储器共用一套地址寄存器，按同一地址码并行地访问各自的对应单元。如假设有 n 个存储器顺序排列的 n 个字，每个字有 w 位，送入这 n 个存储器的地址均为 A，则这 n 个存储器同时访问各自的 A 号单元。我们也可以将这 n 个存储器视作一个大存储器，每个编址对应于 n 字×w 位，因而称为单体多字方式。

多体交叉并行主存系统即使用 n 个容量相同的存储器，或称为 n 个存储体，它们具有

自己的地址寄存器、数据线、时序，可以独立编址地同时工作。各存储体的编址大多采用交叉编址方式，即将一套统一的编址，按序号交叉地分配给各个存储体。一段连续的程序或数据，将交叉地存放在几个存储体中，所以称为多体交叉存取并行主存系统。

19．某机主存按字节编址，而系统总线数据通路宽 32 位。请提出一种连接方案，以粗框描述，并简要说明。

【解题分析】

如果系统总线数据通路宽 32 位，主存是按字节编址，则存储器系统可由四个存储体组成，存储体选择通过选择信号 $\overline{BE_3}$～$\overline{BE_0}$ 实现，该选择信号可以通过用低们地址线译码得到。如果要传送一个 32 位数，那么 4 个存储体都被选中；若要传送一个 16 位数，则有 2 个存储体被选中；若传送的是 8 位数，则只有一个存储体被选中。

【答】

选择存储体与地址译码真的关系如表 3-4-2 所示。存储器与系统总线连接粗框图如图 3-4-11 所示。

表 3-4-2　存储体选择译码表

$\overline{BE_3}$	$\overline{BE_2}$	$\overline{BE_1}$	$\overline{BE_0}$	A_1	A_0	M_1	M_2	M_3	M_4	描述
0	0	0	0	0	0	1	1	1	1	访问 32 位数
1	1	0	0	0	0	1	1	0	0	访问 16 位数
0	0	1	1	0	1	0	0	1	1	访问 16 位数
1	1	1	0	0	0	1	0	0	0	访问 8 位数
1	1	0	1	0	1	0	1	0	0	访问 8 位数
1	0	1	1	1	0	0	0	1	0	访问 8 位数
0	1	1	1	1	1	0	0	0	1	访问 8 位数

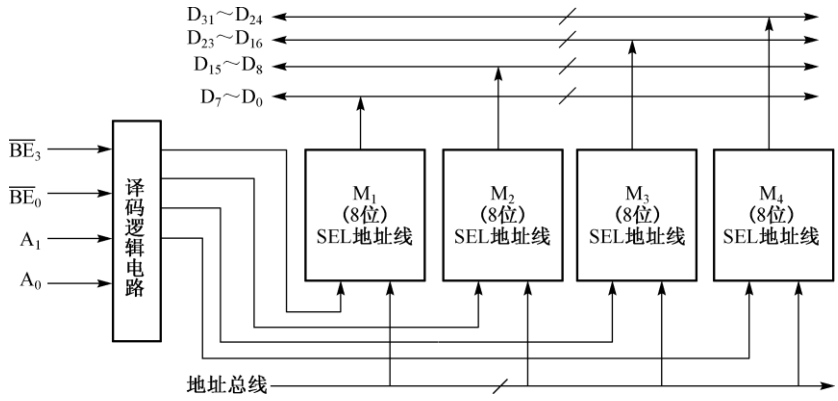


图 3-4-11　存储体连接粗框图

第二种方案则可以采用多体交叉存储器的连接方式，系统应由 4 个独立的存储体组成，各存储体容量相同，以字节为单位编址，每条存储体连接 8 位的数据线，存储体采用交叉编址方式，将主存的整个地址空间，按序号交叉地分配给 4 个存储体。

其连接的粗框图如图 3-4-12 所示。

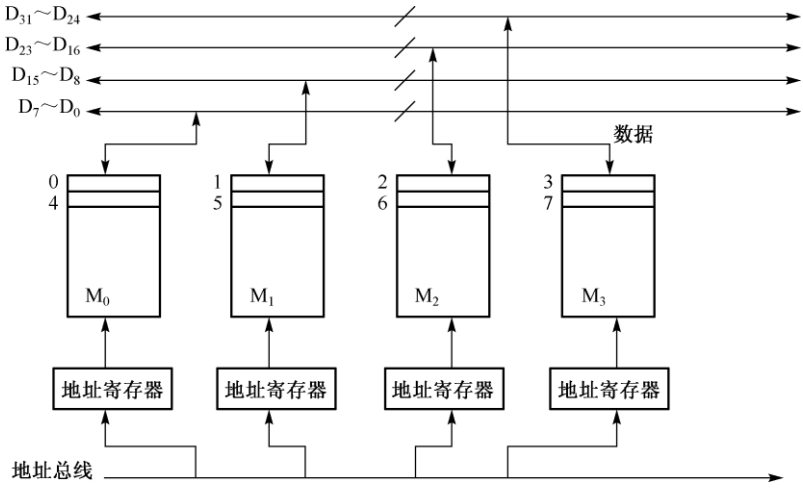


图 3-4-12　多体交叉存储器与系统总线连接方式

当需要传送一个 32 位数时，4 个存储体都被选中，当需要传送一个 16 位数据时，有 2 个存储体被选中；当需要传送 8 位数据时，则只有一个存储体被选中。

20．如果要求半导体存储器在完成读/写后产生信号 READY，并通过系统总线通知其它设备。请设计该信号的产生逻辑。

【解题分析】

如果半导体存储器的读/写时间大于 CPU 所要求的时间，在这种情况下，为了保证 CPU 与存储器的时序的正确配合，就要产生 READY 信号，使 CPU 插入一个等待状态，并通过系统总线通知其它设备。

【答】

能插入一个等待状态的 READY 信号电路逻辑如图 3-4-13 所示。

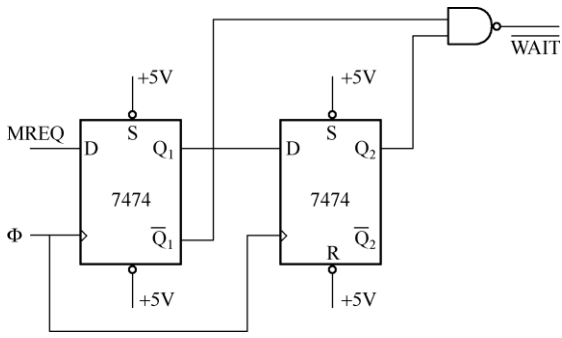


图 3-4-13　产生等待状态 READY 信号的逻辑

21．如果两台 CPU 通过各自的地址总线与数据总线共享一个半导体存储器，请为此设计存储器逻辑，使两组地址/数据线之间能够分离，且能处理访存冲突。

【解题分析】

如果要两台 CPU 通过各自的地址总线与数据总线共享一个半导体存储器，可采用双端口存储器。双端口存储器有两套独立的地址寄存器、地址译码器、数据寄存器和读写电路，可同时接受来自两方面的访存请求，两组地址/数据线之间就能够分离。

【答】

双端口存储器逻辑如图 3-4-14 所示。

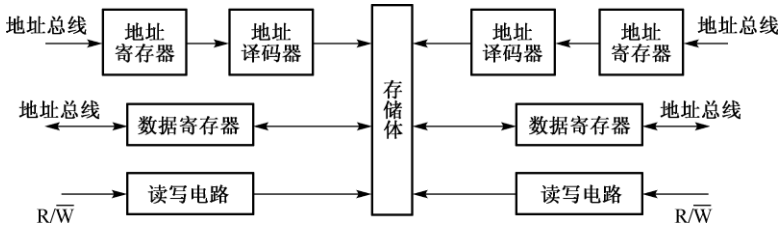


图 3-4-14　双 CPU 共享存储器连接图

以双端口存储器的方式，两个访问端口独立工作互不干扰，只有当两个端口试图在同一时间内访问同一地址单元时，才会发生冲突。这时可由存储器的仲裁逻辑根据两端口访问请求到达存储器的时间差来决定首先为哪一个 CPU 服务。

22．如果需用常规存储芯片构成双端口存储器，允许存取周期延长。请为此设计存储器逻辑。

【答】

如果需用常规存储芯片构成双端口存储器，由于常规存储芯片没有两个独立的端口，所以需要为其设计一套控制逻辑，来实现双端口的功能，在接口中需要实现以下四个方面的逻辑：

a)　两个数据缓冲

b)　地址寄存

c)　存储访问的控制逻辑

d)　共享仲裁控制逻辑

其存储器的逻辑及与系统的连接逻辑如图3-4.15 所示。

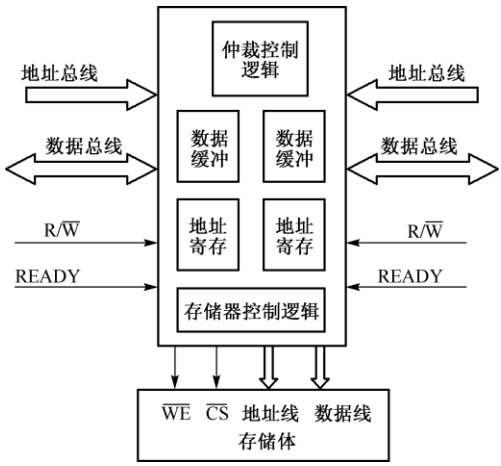


图 3-4-15　以常规存储芯片构成双端口存储器逻辑

如将以上存储系统用于双 CPU 共享的应用中，当两个 CPU 同时对存储器发出访问需求时，控制接口中的共享仲裁控制逻辑就会以访问请求的顺序，在第一个存储器访问周期时允许第一个 CPU 对存储器进行访问，同时向第二个 CPU 发出 READY 等待信号，然后在下一个存储器访问周期撤销该等待信号，允许第 2 个 CPU 对存储器进行访问。

23. 试比较当前光盘与磁盘两者的记录密度、平均存取时间和数据传输率三项性能指标。

【答】

具体见表 3-4-3。

表 3-4-3　光盘与磁盘的性能指标比较

	记录密度	平均存取时间	数据传输率
光盘	可达 300 Mbpi	>100 ms	DVD 可达 21.728 KBps
磁盘	可达 50 Tbpi	<15 ms	SATA 可达 600 MBps

24．主存和 Cache 之间的映射方式有哪几种？请简述每种方式的基本思想，并分析每种方式中如何通过变换内存地址以得到 Cache 的目标地址。

【答】

主要有直接映射、全相联映射和组相联系映射。

直接映射：CACHE 不分组而主存分组，每组包含的数据块数量等于 CACHE 的数据块数。主存映射到 CACHE 时，组内序号为 K 的数据块可映射到 CACHE 中序号为 K 的数据块位。

全相联映射：CACHE 和主存均不分组，主存的第 K 个数据块可自由映射到 CACHE 中的任意数据块位；

组相联映射：CACHE 和主存均分组，且主存各组包含的数据块数等于 CACHE 分组后的组数，主存某组内第 K 个数据块可映射到 CACHE 第 K 组任意一个块位。

25．计算机存储系统按字节编址，其中 Cache 可以分成 16 块，每块的大小固定为 64 字节。在主存和 Cache 之间可以采用三种映射方式，请针对主存的第 268 号单元回答下列关于 Cache 的问题：

【答】

（1）若采用直接映射，那么对应的块号和标记分别是什么？

主存分组，组内块数=16，$268_{10}=100\underline{001100}_2$，每组内的块内字节序号 6 位、组内块序号 4 位，故 268 号字节单元所在的块号是 0100，所在组号全 0；直接映射时标记字段就是组号，因此标记字段也是全 0；

（2）若采用全相联映射，那么对应的块号和标记分别是什么？

主存不分组，缓存的标记字段对应到主存块的块号，$268_{10}=100\underline{001100}_2$，块号=100，标记与块号相同是 100

（3）若采用 2 路组相联映射（即每组 2 块），那么对应的组号和块标记分别是什么？

主存分组，组内的块数等于 CACHE 分组的组数。CACHE 分成 8 组，每组 2 块；

则对应的主存应分成若干组，每组 8 块；$268_{10}=100001100_2$，低 6 位对应块内的字节序号，依次往左 3 位是组内的块序号，最高若干位是组号。

对应在 CACHE 中的组号=主存组内的块序号=100，标记字段=主存的组号为全 0=0；

（4）若采用 4 路组相联映射（即每组 4 块），那么对应的组号和块标记分别是什么？

CACHE 分成 4 组，每组 4 块；主存分成若干组，每组 4 块。

$268_{10}=100001100_2$，低 6 位对应块内的字节序号，依次往左 2 位是组内的块序号，最高位是组号。

对应在 CACHE 中的组号=主存组内的块序号=00，标记字段是 1。

26．什么是 TLB？它和页表之间是什么关系？试分析 TLB 和页表之间的直接映射、全相联映射、组相联映射的异同点。

【答】

TLB 即 Translation Lookaside Buffer，简称快表。TLB 用来专门存储页表中的热点项，也可以认为 TLB 的内容是页表中一部分内容的子集副本。

TLB 和页表之间的直接映射：页表分组、每组的页表行=TLB 行数，映射时页表各组内某行只能映射到 TLB 中对应行序号位置；

全相联映射：页表不分组，页表项可以映射到 TLB 的任何一行；

组相联映射：TLB 分组，页表分组、每组页表行=TLB 分组数，映射时页表各组内第 n 行可以映射到 TLB 中第 n 组内的任何一行。

27．某计算机存储器系统参数如下：

①TLB 共有 256 项，和页表之间按 2 路组相联方式组织映射；

②64KB 的数据 Cache，块大小为 64B，组织方式也是 2 路组相联；

③虚拟地址 32 位，物理地址 24 位；

④页面大小固定为 4KB。

如图 3-4-16 给出了系统的简单示意，请分别计算其中各字段 A、B、C、D、E、F、G、H 和 I 各段所占的位数，并给出计算过程。

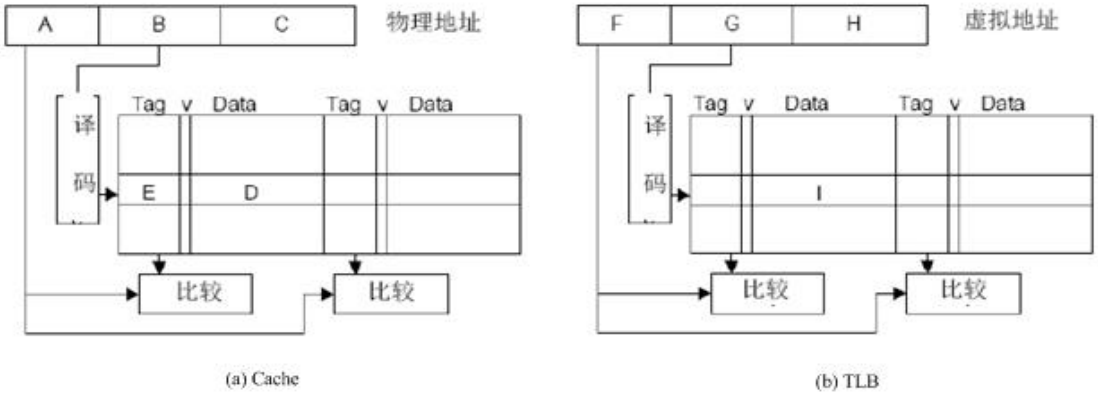


图 3-4-16 存储系统示意

【答】

①TLB共有256项，按两路组相连方式组织：

TLB分成128组，每组2个页表项；页表也按每组128个页表项分组；

②64KB的数据Cache，块大小为64B，组织方式也是两路组相连：

Cache共有1024块，等分成512组，每组2块

③虚拟地址32位，物理地址24位，页面大小为4KB：

题中为说明编址方式，这默认为按字节编址。

页内地址12位，虚页号20位，主存（物理）页号12位。

主-Cache两路组相联映射，块大小为64B，块内地址6位。

主存按每组512（2^9）块分组，组内序号（0～511）占9位地址，内存组号占9位（24-6-9=9），因此：

A=9（内存组号），B=9（组内序号），C=6（块内地址）

Cache中，数据块的大小为64B，64×8=512位，因此：

D=512（Cache数据块的总位数）；

标记字段宽度为E，与内存组号对应，因此：

E=9（Cache块的标记字段）；

TLB-页表采用两路组相联：则TLB分成128组，每组2个页表项；

虚地址32位，其中虚页号20位，页内地址12位（4K），则页表中共有2^{20}=1M个页表项，因此页表项分组时，组内序号为7位（128项），组号为20-7=13位；

F=13（页表项组号），**G=7**（组内序号），**H=12**（页内地址）

TLB中，DATA字段对应的是主存（物理）的页号，物理页号为12位，**I=12**（物理页号）。

28．某计算机的存储器按字节编址，虚拟（逻辑）地址空间为 16 MB，主存（物理）地址空间为 1 MB，采用每页固定为 4 KB 的页式虚拟存储管理。…

【答】

（1）虚拟地址共有几位，哪几位表示虚页号？物理地址共有几位，哪几位表示页框号（物理页号）？

由于虚拟地址空间大小为16MB，且按字节编址，所以虚拟地址共有24位（2^{24}=16M）。由于页面大小为4KB（2^{12}=4K，则页内地址12位），故虚页号为高12位（24-12=12）。由于主存（物理）地址空间大小为1MB，所以物理地址共有20位（2^{20}=1M）。

由于主内的页内地址12位，所以20-12=8，即主存地址的高8位为页框号。

（2）使用物理地址访问Cache 时，物理地址应划分成哪几个字段？要求说明每个字段的位数及在物理地址中的位置。

由题有Cache共分8块，块号需占3位（2^3=8）；且块大小为32B，则块内地址应为5位（2^5=32）；主存-Cache采用直接映射，则主存也按每组8块进行统一分组，每组内的块号占3位（2^3=8），则组号占12位（物理地址20位-5位块内地址-3位块号=12位）；故物理地址分三段：高12位（内存组号）+中间3位（组内块序号）+5位（块内地址）。

（3）虚拟地址001C60H 所在的页面是否在主存中？若在主存中，则该虚拟地址对应的物理地址是什么？访问该地址时是否Cache 命中？要求说明理由。

虚地址 001C60H，则 C60H 为页内地址，剩余为虚页号即 001H=1，故应查图 A 中虚页号为 1 的那行，即第 1 行：有效位为 1、对应页框号为 04H；故该虚页面已在主存中，且对应的物理地址为应为页框号 04H 与页内地址 C60H 拼接，结果为04C60H；

物理地址为 04C60H，主存←→Cache 直接映射，则内存分组后，12 位的组号为 04CH；而地址码 60H=0110 0000（二进制）故 3 位的组内块序号为 011（二进制）=3（十进制），所以该主存页应直接映射到 Cache 块序号（行号）为 3 的存储块位置，但查图 B 中行号为 3 的那行，得标记位为 105H≠04CH，故 Cache 访问不命中。

（4）假定为该计算机配置了一个用于4 路组相联映射的TLB，可以存放8 个页表项，若其当前内容（十六进制）如表C 所示，则虚拟地址024BACH 所在的页面是否已被调入到主存中？

页表←→TLB 之间四路组相联，而 TLB 可存 8 个页表项，则每组存 4 个页表项，共分成 2 组；此外，页表也应按每组 2 项进行分组，而虚页号是 12 位，每 1 个虚页在页表中对应 1 行，故页表中应有 212行，则页表行号应为 12 位所以页表分组后：组号 11 位+组内序号 1 位；

虚地址是 024BACH，则 12 位的虚页号为 024H（0000 0010 0100），而组号是其高 11 位，组内序号是其低 1 位，则分解成：组号 0000 0010 010（十六进制数 012H）+组内序号 0；因此该虚地址所在虚页所对应的页表项应映射到 TLB 的第 0 组（页表项的组内序号=TLB 的组号）。

因页表←→TLB 是组相联映射，故应分别查找图 C 显示的 TLB 表中第 0 组的 4 个 TLB 项，将各项的标记字段与 012H 比较，可知本组内第 4 项标记字段为 012H，与该虚地址所在页面对应的页表项的组号 012H 一致，且有效为 1，页框号为 1F；据此可判断虚地址 024BACH 所在的虚页面已在主存中（且页框号为 1F），且虚地址对应的主存地址应为：页框号+业内地址=1FH+BACH=**1FBACH**。

3.5 第 5 章习题解答

1．简要解释下列名词术语

【答】

系统总线：计算机系统内各功能部件之间、或各插件板之间互连的总线。

外总线：计算机系统之间，或计算机系统与其他系统之间互连的总线。

局部总线：直接与 CPU 连接的一段总线，称为局部总线。

通信总线：连接远距离信息传送的通信工具之间的传送线路，称为通信总线，常采用串行总线做通信总线。

同步总线：数据的传送操作由统一的系统时钟同步定时的总线，称为同步总线。

异步总线：无固定时钟周期划分，总线周期时间由各部件操作的实际需要决定，采用异步应答方式控制传送操作的总线。

缓冲深度：接口中设置的数据缓冲寄存器的数量或缓冲区的容量，称为缓冲深度。

中断接口：如果主机与外围设备之间的信息传送采用程序中断方式控制，则接口需有相应的中断系统所需的逻辑，这样的接口称为中断接口。

DMA 接口：如果主机与高速外围设备之间的信息传送采用 DMA 方式控制，则接口中需有相应的 DMA 逻辑，这样的接口被称为 DMA 接口。

总线宽度：任何总线的线路在功能上可分为 3 个组：数据线、地址线和控制线。所谓总线宽度即各功能组中的信号线数。

主设备：申请并得到总线控制权的设备，称为主设备。

从设备：响应主设备请求的设备，称为从设备。

直接程序传送方式：CPU 直接利用 I/O 指令编程，实现数据的输入和输出的方式，称为直接程序传送方式。

中断：CPU 暂时中断现行程序的执行，转去执行处理某些随机事态的中断服务程序，处理完后自动恢复原程序的执行。

软中断：软中断是由执行软中断指令所引起的中断。

实时处理：实时处理是指在事件出现的实际时间内及时地进行处理。

硬件中断：硬件中断指由某个硬件中断请求信号引发的中断。

可屏蔽中断：CPU 可以通过送屏蔽字或开/关中断来禁止和开放中断，中断请求能否响应由 CPU 开/并决定。

非屏蔽中断：中断请求的响应不受屏蔽字影响，与 CPU 的开/关中断状态无关。

向量中断：将各个中断服务程序的入口地址（或包括状态字）组织成中断向量表；响应中断时，由硬件直接产生对应于中断源的向量地址；据此访问中断向量表，从中读取服务程序入口地址，由此转向服务程序。

非向量中断：以软件查询的方式提供中断服务程序入口地址的中断称为非向量中断。

中断向量：中断服务程序入口地址和状态字在一起，称为中断向量。

向量地址：访问中断向量表的地址码，即读取中断向量所需的地址。

中断向量表：用来存放中断向量的一种表格，称为中断向量表。

中断隐指令操作：在中断周期中，直接依靠硬件实现关中断、保存断点、通过中断类型码计算中断程序的入口地址并转中断服务程序等操作，称为中断隐指令操作。

现场保护：执行中断服务程序时，可能使用某些寄存器，这就会破坏它们原先保存的内容。因此，需要事先将它们的内容保存起来，称为现场保护。

DMA 方式：即直接存储器访问，DMA 方式是直接依靠硬件实现主存储器和外部设备间进行数据传送，传送时不需要 CPU 的干预。

DMA 初始化：在 DMA 传送操作开始前，为实现有关控制，CPU 需要事先向 DMA 控制器送出有关控制信息。在调用 I/O 设备时，通过程序所做的这些准备工作，称作 DMA 初始化。

2．比较并说明下述几种 I/O 控制方式的优缺点及其适用场合。

（1）直接程序传送方式（含程序查询方式）。

（2）程序中断方式。

（3）DMA 方式

【答】

（1）直接程序传送方式的优点：在直接程序传送方式中 CPU 只需实现 I/O 指令功能，不需要增加 CPU 硬件，因此硬件简单、控制简单。缺点：CPU 与外围设备完全不能并行地工作，CPU 利用率低，实时处理能力低。适用场合：CPU 速度不是很高，效率问题不是很重要的场合。

（2）程序中断方式：由于中断是通过执行程序进行对事件的服务处理，处理程序可以根据需要进行扩展，所以程序中断方式的处理能力很强，可以处理复杂事态，随机性和实时性强。缺点：为了实现程序切换，决定 CPU 在中断周期中要执行隐指令操作：保存断点、读取服务程序入口地址，以及在转入服务程序后首先应执行现场保护等操作；在返回原程序前，还需恢复现场、读取返回地址等。这一系列的操作要花费一定的时间，使中断方式难于适应高速数据传送。适用场合：适用于处理中、低速的 I/O 操作与随机请求，所处理的对象可以是复杂的随机事态。

（3）DMA 方式：优点：可以响应随机请求，传送的实现是直接由硬件控制，CPU 不必为此执行指令，其程序不受影响，DMA 方式仅需占用系统总线，不需切换程序，因此不存在保存断点、保护现场、恢复现场、恢复断点等操作。缺点：仅仅依靠硬件，只能实现简单数据传送，难于识别与处理复杂事态。适用场合：适用于主存与高速 I/O 设备间的简单数据传送。

3．某机连接四台 I/O 设备，序号由 0#～3#，采用软件查询确定其中断优先级。请分别按下列两种要求拟定查询程序流程图。

（1）固定优先级。

（2）轮流优先，使机会均衡。

【答】

（1）在固定优先级方式中，CPU 响应中断请求后，先转入查询程序，按优先顺序依次询问各中断源是否提出中断请求，如果是，则转入相应的服务处理程序。假设在公共接口中设置一个中断请求寄存器，用来存放各中断源提出的请求信息，如对应位为 1 表示该中断源提出了请求。如设置为连接在最高位的中断源的优先级最高，依次优先级逐位降低。在进行软件查询时，先将中断请求寄存器的内容取回 CPU，然后依次判断，其程序流程如图3-5-1 所示。

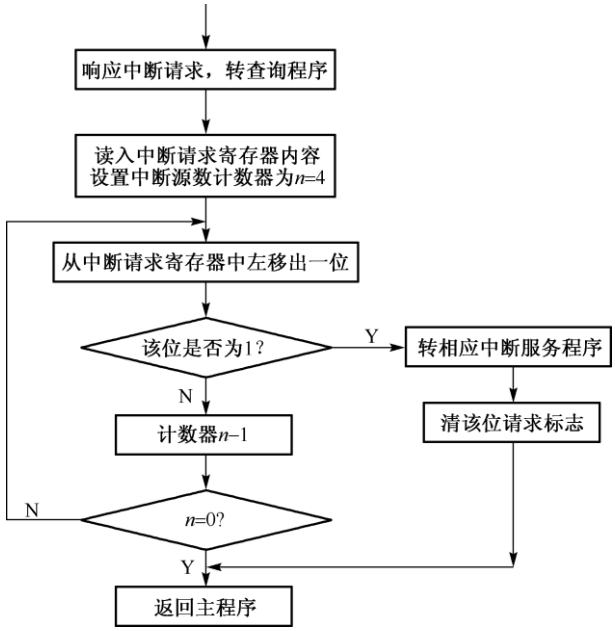


图 3-5-1　固定优先级流程图

（2）轮流优先

在轮流优先方式中，需设置中断请求寄存器，用于记忆当前有哪些中断源提出了中断请求。当某中断源提出中断请求后，中断请求寄存器的相应位设置为 1。另需设置记忆中断优先级状态的寄存器，描述各中断源的优先级，设优先权的等级为 3 时最低，为 0 时最高。如 0#~3#设备的初始优先级状态为 0、1、2、3，即：

	设备 3#	设备 2#	设备 1#	设备 0#
优先级状态	3	2	1	0

查询程序的流程如图 3-5-2 所示。

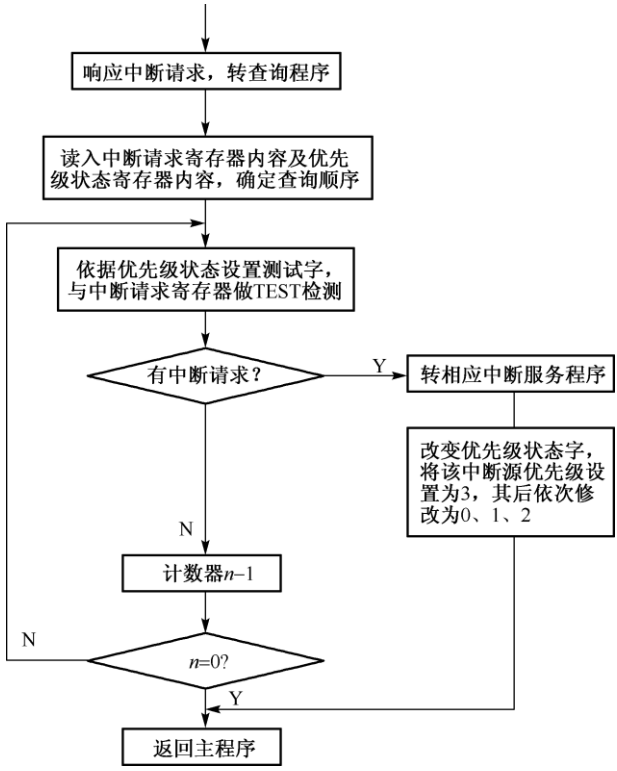


图 3-5-2　轮流优先级流程图

4、分别对教材中图 5－20 所示两种结构，设计优先链排队逻辑，画出门级逻辑电路。

【答】

（1）针对多重查询方式其优先链排队逻辑图如图 3-5-3 示，在上半部分由门 1～门 6 组成一个串行的优先链，优先顺序从高到低为：$INTR_1$、$INTR_2$、$INTR_3$。若要扩充中断源可根据其优先级别的高低串接于优先链的左端或右端。

图 3.5.9 的下半部分是一个编码电路，由总线向 CPU 发出被选中的设备码以供判定中断源。以下简单说明其工作过程，$INTR_i$ 为设备的中断请求信号，$INTS_i$ 为设备的中断排队选中信号。INTA 为中断许可信号。INTI 为中断排队输入，INTO 为中断排队输出信号。

当一个设备提出中断请求时，它将提供两个信号：$\overline{INTR}$ 在链中以低电平封锁优先级别比它低的请求；如果请求被选中，它的 INTI 在链中以高电平打开自己的编码门，以提供设备码。

可以分成三种情况讨论线路的状态：

① 若优先级高的 I/O 设备无请求时，则对优先级低的 I/O 设备请求开放。例如，没有比本优先链中更高级别的请求时，则 $\overline{INTI} = 0$，门 1 的输出为“1”（高电平），即 $INTS_1 = 1$，允许设备 1 被选中。

② 若设备 1 此刻无中断请求，$INTR_1 = 1$，门 2 输出为“0”，则 $INTS_2 = 1$，就允许设备 2 被选中。若优先级高的 I/O 设备有请求，$INTR_1 = 0$，门 2 输出为“1”，则 $INTS_2 = 0$，$INTS_3 = 0$，即设备 2，设备 3 的请求都被封锁。

③ 若有两个以上的 I/O 设备同时提出请求，则只有优先级别高的 I/O 设备被选中。如设备 1 和设备 2 同时提出请求，当 CPU 接到请求信号并响应时，可通过程序发出一个中断询问命令查询是谁被选中，此时只有 $INTS_1 = 1$，而 $INTS_2 = 0$，所以是设备 1 被选中，取回该设备的设备码。

由上得出，经过链式排队，谁的优先级别高谁就向主机提供自己的设备码，主机就执行它的服务程序。

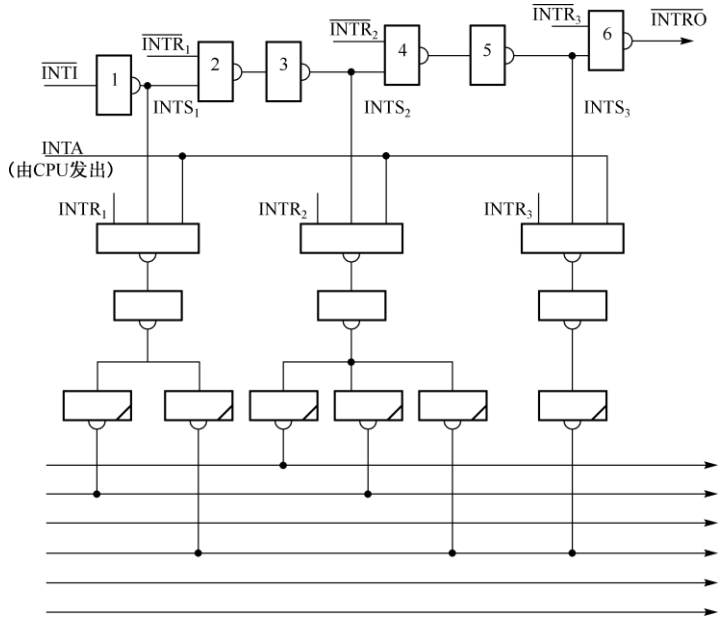


图 3-5-3　多重查询方式式判优逻辑图

（2）若以菊花链式判优，则其连接逻辑如图 3-5-4 所示。

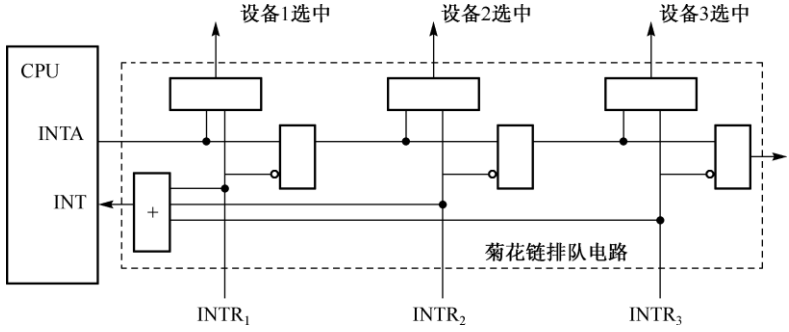


图 3-5-4　菊花链式判优逻辑图

5．串行接口和并行接口的实质区别是什么？

【答】

接口和总线之间，数据都是并行的。但对于串行接口，在接口和外设之间的数据是串行的，而对于并行接口，在接口和外设之间的数据则是并行的。

6．I/O 接口的主要功能有哪些？

【答】

外设寄存器寻址；数据传输和缓冲；数据格式变换等；产生中断及代码请求等控制逻辑功能。

7．程序中断方式与转子有何不同？

【答】

中断与转子程序两者的本质区别主要表现在：

① 转子子程序的执行是由程序员事先安排好的，中断服务程序的执行则是由随机的中断事件引起的调用。

② 转子子程序的执行受到主程序或上层子程序的控制，中断服务程序一般与被中断的现行程序没有关系。

③ 不存在一个程序同时调用多个转子子程序的情况，却经常可能发生多个外设同时请求 CPU 为自己服务的情况。

8．什么是向量中断方式与非向量中断方式？各有何优缺点？

【答】

非向量中断：将所有中断源的中断服务程序入口地址组织在公共的查询程序中；CPU 响应时执行此查询程序，确定中断源对应的服务程序入口地址，再转入相应服务程序执行。

向量中断：将所有中断源的中断服务程序入口地址(中断向量)组织在中断向量表中；CPU 响应时由硬件产生向量地址，据此查表确定服务程序入口地址，再转入相应服务程序执行。

非向量中断通过软件来确定中断源，可以简化硬件逻辑，便于灵活修改中断源优先级，但非向量中断的响应速度比向量中断更慢。

9．某机连接四台 I/O 设备，序号由 0#～3#，允许多重中断。

（1）优先顺序为 0#～3#，分别拟定响应各设备请求后应送出的屏蔽字。

（2）若采取轮流优先策略，则屏蔽字应如何变化？

【答】

（1）在该中断系统中，由于允许多重中断，在第一种情况下，当某个设备的请求得到响应后，应屏蔽比该中断源优先级低和同级的设备中断请求，而开放比该中断源优先级高的设备的中断请求信号。根据题意这 4 台设备的优先顺序分别为 0#～3#，当要屏蔽某中断源的中断请求时，需将该中断源的屏蔽位置为“1”，所以响应各设备请求后，应送出的屏蔽字分别为 0111、0011、0001、0000。

（2）若采取轮流优先的策略，则当某设备的中断请求得到响应后，将把比该设备优先级低一级的设备优先级置为最高，所以在轮流优先中响应各设备请求后，应送出的屏蔽字应分别为 1011、1101、1110、0111。

10．某 I/O 设备的工作状态可抽象为空闲、忙、完成，CPU 发来清除命令使其进入空闲

状态，启动命令使其进入“忙”状态，设备完成一次操作使其进入完成状态。若进入完成状态，且 CPU 没有对其屏蔽，则提出中断申请。试为此设计中断接口，画出逻辑图。

【答】

其中断接口如图 3-5-5 所示。

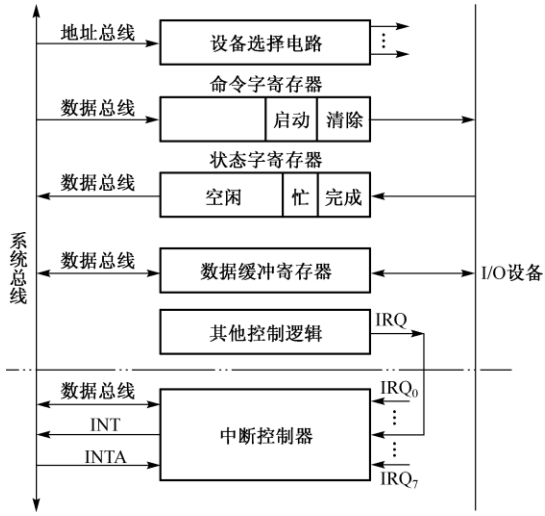


图 3-5-5　中断接口逻辑图

11．某机用于控制 8 层楼电梯系统，请根据你对电梯运行方式的了解，设计中断接口，画出寄存器级逻辑粗框图，并拟定其命令字与状态字格式。

【答】

用于电梯控制的命令字状态字格式如图 3-5-6 所示。

命令字寄存器中存放 CPU 控制电梯动作所发出的命令。当有人按下电梯按钮时，状态字寄存器中存放当前是哪一层有使用电梯的请求，及该请信号来自电梯内还是电梯外，分别用两个状态字寄存器存放梯内和梯外各层的请求。

分别用两个数据缓冲寄存器存放电梯运行当前到达的位置和运行的速度。CPU 读入数据缓冲寄存器中的数据，与状态字寄存器中的请求状态做比较，向命令字寄存器中送控制命令，以确定当前是应该启动电梯，或是停止；是提速（从一层启动时）、减速（要到达有请求的楼层时）还是保持当前速度运行，是向上运动还是向下运动等。

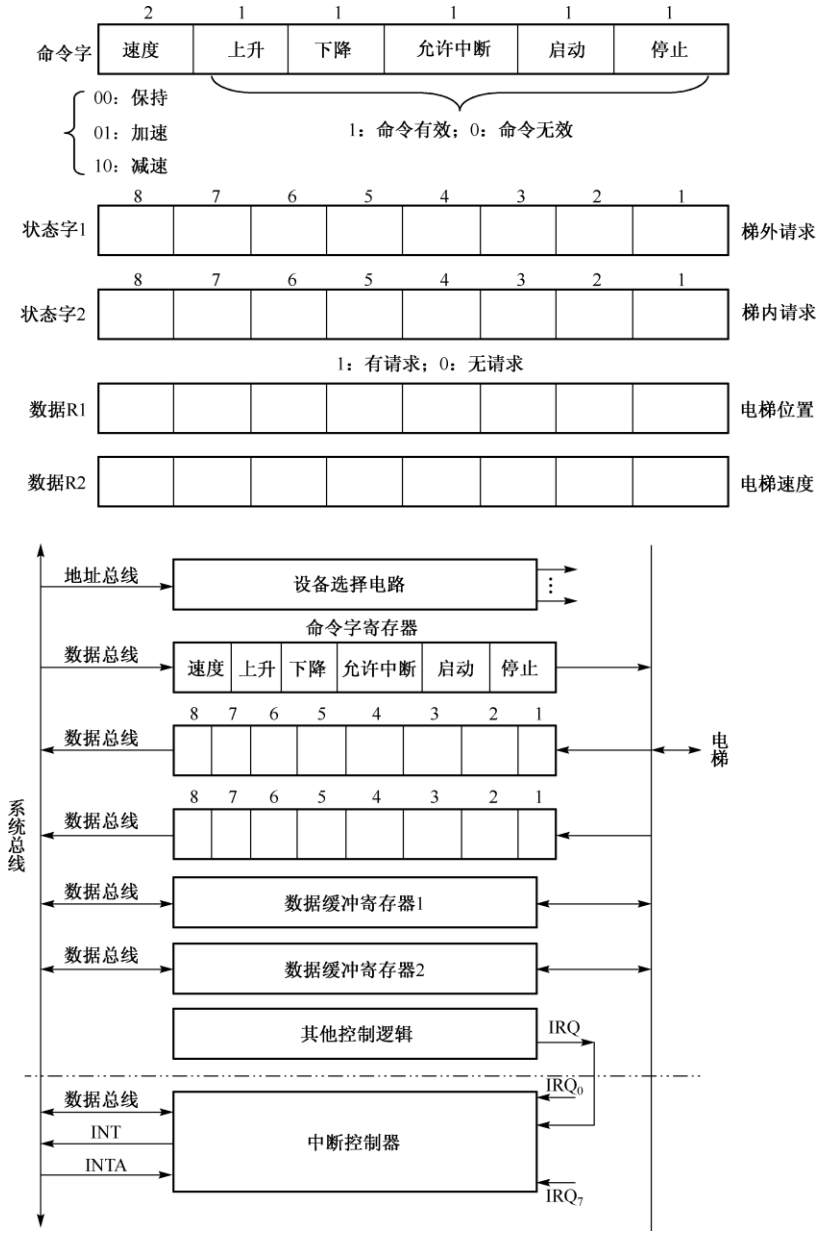


图 3-5-6　电梯控制中断接口图

12．在 DMA 的初始化阶段中，主要应完成哪些任务？

【答】

（1）向接口送出 I/O 设备的寻址信息。例如，要从磁盘中读出一个文件，则需送出该文件所在磁盘的驱动器号、圆柱面号、磁头号（记录面号）、起始扇区号（或数据块号）。

（2）向 DMA 控制器送出控制字，主要是数据的传输方向，输入主存还是从主存输出。

（3）向 DMA 控制器送出主存缓冲区首地址。数据的输入或输出，往往需在主存储器中设置相应的缓冲区，这是一段连续的存储区，为此在初始化时送出其首地址。

（4）向 DMA 控制器送出交换量，即数据的传输量。视设备的需要，传输量可以是字节数、字数、数据块数。

13．总线系统的信号类型主要有哪几种？

【答】

数据，地址和控制信号。

14．通道的类型有哪几种？分别应用在什么场合？

【答】

根据数据传输方式的不同，通道总体上可以分为两种：选择型通道和多路型通道。

（1）选择型（Selector）通道

选择型通道（只能以突发模式工作）可连接多个外设，但这些设备不能同时工作，在某个时间段内只能选通 1 个设备进行工作，只有当这个设备的全部通道程序执行完后，才能选择其他设备进行工作。选择通道主要用于连接高速的外部设备，例如磁盘等。

（2）多路型（Multiplexer）通道

多路型通道也可以连接多个外设，兼容选择型通道，在某一时段内可以同时选通多路设备进行工作，且允许这些外设按一定的单位轮流进行数据的输入、输出。根据每次数据传输单位的差异，多路型通道又分为字节多路型通道和数组多路型通道。

① 字节多路通道。字节多路通道是一种简单的分时共享通道，主要用于连接多路低速或中速设备。通道选择一路设备后，该设备开始传输数据，但只能传输 1 字节，若该设备有多字节数据需要传输，则该外设需要多次请求，被通道多次选择才能完成传输。

字节多路型将通道的可用传输时间分为若干个时间片，每个时间片传输 1 字节，由多个外设分时方式共享一个通道。字节多路型通道主要用于连接多种低速外设。

② 数组多路通道。数组多路型通道在物理上可连接多路高速外设，数据以成组（块）交叉的方式进行传输。通常，一个外设在进行工作时，除了数据传输，还包括寻址等操作。数组多路通道的基本思想是：当某个设备进行数据传输时，通道只为该设备服务；当设备在执行寻址等非传输型操作时，通道暂时断开与这个设备的连接，挂起该设备的通道程序，去执行其他设备的通道程序，为该设备的输入输出服务。

15．1 字节多路通道连接 A、B、C、D、E 共 5 台设备，这些设备分别每 10 μs、20 μs、30 μs、50 μs 和 75 μs 向通道发出一次数据传输的服务请求，请回答下列问题：

（1）这个字节多路通道的实际流量是多少？

（2）如果设计字节多路通道的最大流量正好等于通道实际流量，并假设对于这 5 台设备通道对它们的响应优先级为 A>B>C>D>E。试分析，计算通道能正常为这 5 台设备提供传输通道吗？如果不能，应如何解决这个问题？

【答】

（1）通道的实际流量为各子通道的数据传输率之和，即：

DR=1B/10us+1B/20us+1B/30us+1B/50us+1B/75us=216.7KB/S

（2）通道的工作周期为：1B÷216.7KB/S=4.6uS

各设备的请求和通道响应时间关系如图 3-5-7 所示：

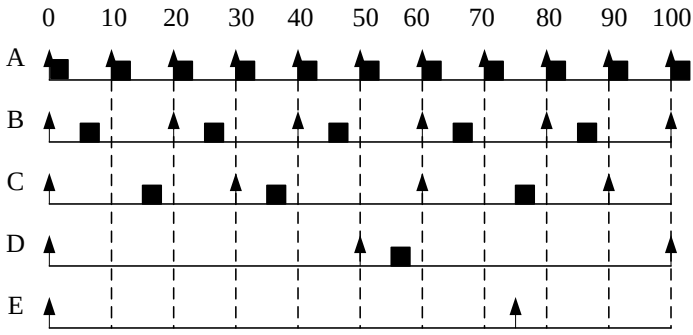


图 3-5-7　通道相应时间关系

如图，设备 D、E 的数据请求得不到及时响应，由此可见通道不能为这 5 台设备提供正常的传输通道。

解决办法：提高通道的最大流量，以减小通道的工作周期。

【注】本题为《计算机系统结构》涉及到的内容，超出了计算机组成原理课程的知识范围，仅供读者参考练习。

3.6　第 6 章习题解答

1．简要解释下列名词术语

【答】

I/0 设备：I/0 设备是计算机系统与人、其它设备或系统之间进行信息交换的装置，主机以外的大部硬件设备都称为外围设备或 I/0 设备。如打印机、显示器和磁盘等设备。

并行传送：同时采用多根传送线，并行地传送一个字节或一个字，称为并行传送。

串行传送：采用单根信号传送线（对公共地形成电位差）或用一对传送线（一根信号线，一根地线）逐位串行地传送数据代码，称为串行传送。

扫描码：通过硬件（或软件）扫描，识别所按下的键的行列位置，表示该行列位置的代码称为扫描码，也称为位置码或座标码。

逐行扫描法：将键连成 m 行×n 列的矩阵，在执行键盘扫描程序时，CPU 从第 0 行开始，逐行为 0（其余各行为 1）的方式，将代码 0 送往行线，并取回列线状态来判别按键位置，这种确定按键位置的方法称为逐行扫描法。

行列扫描法：先逐列为 1 地步进扫描，CPU 测试是哪一列为 1 时行线组输出为“1”，从而判明按键的列号。再逐行为“1”地步进扫描，CPU 测试哪一行为 1 时，列线组输出也是“1”，即判明哪行按了键。CPU 根据行、列扫描结果便能确定按键的位置。行列号形成对应的扫描码（或位置码），这种键盘扫描法称为行列扫描法。

像点（像素）：将显示屏幕沿水平和垂直方向划分成 $m \times n$ 个最少的显示区域，这个最小区域就是显示器的最小显示单元，对应屏上一个显示点，该最小显示区域称为像点或像

素。

分辨率：分辨率是指显示器一屏所能表示的像素的最大个数。

灰度级：在黑白显示器中，灰度级是指所显示的像素点的亮暗程度，在彩色显示器中，则表示颜色的不同。

字符/数字方式：对显示器（或打印机）而言，以字符作为显示（或打印）内容的基本单元，这种方式称为字符/数字方式，也称为文本显示（或打印）方式。

图形方式：以像点作为显示内容的基本单元的显示方式称为圆形方式，某些打印机也有相似的圆形打印方式。

位密度：沿磁道圆周，单位距离可记录的二进制代码位数称为位密度。

道容量：指一条磁道所能存放的二进制代码位数。

2．简答题

（1）磁盘的工作速度是由什么因素决定的？

【答】

磁盘的工作速度是由三部分组成的，即平均寻道时间、平均旋转延迟时间和数据传输率。前两部分的决定因素主要是磁盘主轴转速和磁头移动速度；数据传输率则与磁盘读/写速度、主存速度和接口逻辑线路有关。

（2）举例说明一种实用的键码形成方法。

【答】

设有 4×4 的扫描式键盘矩阵如图 3-6-1 所示，用逐行扫描法来获取键码。

键码形成过程如下：

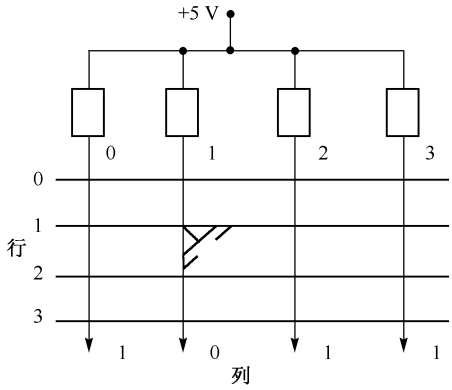


图 3-6-1 一个 4×4 扫描式键盘矩阵

①当 5 号键按下时，产生中断请求，CPU 执行键盘扫描程序。

②在扫描程序中，CPU 使第 0 行线为 0（其余行为 1），同时取出各列线状态，由于 0 行上无键按下，故各列均为 1。

③CPU 接着使第 1 行线为 0，由于 5 号键已按下，于是 CPU 从取出的列线状态中发现第 1 列为 0，于是得到按键扫描码（1，1）。

④根据扫描码（1，1）查找键码转换表，即可获得 5 号键的键码。

（3）简述 CRT 显示器由字符代码到字符点阵的转换过程。

【答】

在字符方式下，首先从显示缓存取出字符编码；将字符编码作为字符发生器地址的高位部分，将 CRT 控制器提供的扫描时序（即线计数器的输出）作为地址低位部分送字符发生器；读取出该字符的一行点阵代码，并行送入移位寄存器；经并→串转换，送出点脉冲信号到 CRT 管，即可在屏上显示该字符的一行（即线）点阵图案。

（4）简述激光打印的基本原理。

【答】

激光打印机利用激光扫描技术将经过调制的、载有字符点阵信息或图形信息的激光束扫描在光导材料上，并利用电子摄影技术让激光照射过的部分曝光，形成图形的静电潜像。再经过墨粉显影、电场转印和热压定影，便在纸上印刷出可见的字符或图形。

（5）字符显示器为了实现同步控制，一般应设置哪几级同步计数器？

【答】

字符显示器一般应设四级同步计数器，依次为点计数器、字符计数器、线计数器和行计数器。

（6）图形显示器应设置哪几级同步计数器？

【答】

一般应设三级同步计数器，依次为点计数器、字节计数器和线计数器。

3．某 CRT 显示器作为字符显示，能显示 64 种字符，每帧可显示的最大容量为 25 行×64 列字符，每个字符采用 7×8 点阵，即横向 7 点，纵向 8 点，则字符发生器的容量为多少？

【答】

每个字符都是采用 7×8 点阵规格来显示，习惯上常将横向 7 点的代码放一个字节单元（最高位为 0），一个字符有 8 行点，故一个字符需要 8 个字节来放它的点阵代码。64 种字符则需要 8×64=512 个字节，故该字符发生器容量为 512 字节。

理论上，字符发生器容量=7×8×64=3584（bit）。

4．某 CRT 显示器作为字符显示，每帧可以显示 25 行×80 列字符，每个字符采用横 7×纵 9 点阵，字符间横向间距 2 点，行间间距 5 点。则点计数器应如何分频？字符计数器应如何分频？线计数器应如何分频？行计数器应如何分频？

【答】

点计数器分频关系为（7+2）：1；

字符计数器分频关系为（80+L）：1；

线计数器分频关系为（9+5）：1；

行计数器分频关系为（25+M）：1；

注：L 为屏幕两边非线性失真和水平回扫消隐部分折合的字符个数，M 为屏幕顶部、底部非线性失真和垂直回扫消隐部份折合的字符行数。

5．某图形显示器的分辨率为横向点 800×纵向点 600，则同步计数器的点计数器应如何分频？字节计数器应如何分频？线计数器应如何分频？

【答】

点计数器分频为 8:1

字节计数器分频为 $(800/8+L):1$

线计数器分频为 $(600+M):1$

注：L 为屏幕两边非线性部分和水平回扫等消隐段折合成的字节数，M 是垂直回扫和屏幕上下边缘非线性失真的消隐段折合的数线。

6．在字符显示器与点阵针式打印机中都有字符发生器，它们的主要区别是什么？

【答】

字符显示器中的字符发生器是按点阵行进行读出，即每次读取字符点阵的一行代码，故字符的点阵代码在字符发生器中是按行存储。点阵针式打印机的字符发生器是将字符的点阵代码按列存放，每次读取字符点阵的一列代码，进行打印。

7．若要将图形显示的分辨率从 800×600 提高到 1024×1024，在适配卡上应采用哪些措施？

【答】

应采取的主要措施是：增加刷新存储器（即显示缓存）容量，以便存放一帧像点代码；提高点脉冲和锯齿波频率，缩短显示缓存的存取周期；重新设置 CRTC 中相应寄存器的初值和同步计数器分频关系等。

8．若要将显示字符从 7×9 点阵放大为 14×18 点阵，请提出一种实现方案。

【答】

扫描显示第一帧时，显示缓存中存放的是字符 7×9 点阵代码。扫描显示第二帧时，将显示缓存的内容修改为字符 14×18 点阵代码，即可实现放大。

9．若要将显示画面（字符型）自下而上地滚动，请提出一种实现方案。

【答】

每帧扫描结束后，字符指针累加 1。

显示不同帧时，自下而上地改变显示画面中字符的编码在显示缓存中的存放位置，即可实现相应滚动。

10．若要将显示字符 A 从屏幕左上角逐渐地移向屏幕右下角，请提出一种实现方案。

【答】

每帧扫描结束后，线计数器和字符计数器分别延迟后再读字符。

显示不同帧时，改变每帧 A 编码在显示缓存中的存放位置，即可实现相应移动。

11．若要让一个图形在屏幕上旋转，请提出一种实现方案。

【答】

可考虑将字符点阵代码循环左右移动，可实现旋转。

显示不同帧时，按旋转方向改变图形在显示缓存中的存放位置，即可实现图形的旋转。

说明：有关 CRT 显示器相关的内容已比较陈旧，在第 4 版教材的相关教学中中已不再重点讲授，读者只需要做常识性了解。