计算机组成原理习题答案

第一章

1. 模拟计算机的特点是数值由连续量来表示，运算过程也是连续的。数字计算机的主要特点是按位运算，并且不连续地跳动计算。模拟计算机用电压表示数据，采用电压组合和测量值的计算方式，盘上连线的控制方式，而数字计算机用数字 0 和 1 表示数据，采用数字计数的计算方式，程序控制的控制方式。数字计算机与模拟计算机相比，精度高，数据存储量大，逻辑判断能力强。
2. 数字计算机可分为专用计算机和通用计算机，是根据计算机的效率、速度、价格、运行的经济性和适应性来划分的。
3. 科学计算、自动控制、测量和测试、信息处理、教育和卫生、家用电器、人工智能。
4. 主要设计思想是：采用存储程序的方式，编制好的程序和数据存放在同一存储器中，计算机可以在无人干预的情况下自动完成逐条取出指令和执行指令的任务；在机器内部，指令和数据均以二进制码表示,指令在存储器中按执行顺序存放。主要组成部分有： 运算器、逻辑器、存储器、输入设备和输出设备。
5. 存储器所有存储单元的总数称为存储器的存储容量。每个存储单元都有编号，称为单元地址。如果某字代表要处理的数据，称为数据字。如果某字为一条指令，称为指令字。
6. 计算机硬件可直接执行的每一个基本的算术运算或逻辑运算操作称为一条指令，而解算某一问题的一串指令序列，称为程序。
7. 取指周期中从内存读出的信息流是指令流，而在执行器周期中从内存读出的信息流是数据流。
8. 半导体存储器称为内存，存储容量更大的磁盘存储器和光盘存储器称为外存，内存和外存共同用来保存二进制数据。运算器和控制器合在一起称为中央处理器，简称 CPU，它用来控制计算机及进行算术逻辑运算。适配器是外围设备与主机联系的桥梁，它的作用相当于一个转换器，使主机和外围设备并行协调地工作。
9. 计算机的系统软件包括系统程序和应用程序。系统程序用来简化程序设计，简化使用方法，提高计算机的使用效率，发挥和扩大计算机的功能用用途；应用程序是用户利用计算机来解决某些问题而编制的程序。
10. 在早期的计算机中，人们是直接用机器语言来编写程序的，这种程序称为手编程序或目的程序；后来，为了编写程序方便和提高使用效率，人们使用汇编语言来编写程序，称为汇编程序；为了进一步实现程序自动化和便于程序交流，使不熟悉具体计算机的人也能很方便地使用计算机，人们又创造了算法语言，用算法语言编写的程序称为源程序，源程序通过编译系统产生编译程序，也可通过解释系统进行解释执行；随着计算机技术的日益发展，人们又创造出操作系统；随着计算机在信息处理、情报检索及各种管理系统中应用的发展，要求大量处理某些数据，建立和检索大量的表格，于是产生了数据库管理系统。
11. 第一级是微程序设计级，这是一个实在的硬件级，它由机器硬件直接执行微指令；第二级是一般机器级，也称为机器语言级，它由程序解释机器指令系统；第三级是操作系统级，它由操作系统实现；第四级是汇编语言级，它给程序人员提供一种符号形式语言，以减少程序编写的复杂性；第五级是高级语言级，它是面向用户的，为方便用户编写应用程序而设置的。用一系列的级来组成计算机的接口对于掌握计算机是如何组成的提供了一种好的结构和体制，而且用这种分级的观点来设计计算机对保证产生一个良好的系统结构也是很有帮助的。

12. 因为任何操作可以由软件来实现，也可以由硬件来实现；任何指令的执行可以由硬件完成，也可以由软件来完成。实现这种转化的媒介是软件与硬件的逻辑等价性。

13. 计算机应用和应用计算机在概念上是不等价的。

计算机应用是计算机学科与其他学科相结合的交叉学科，是计算机学科的组成部分，分为数值计算和非数值应用两大领域。

应用计算机是借助计算机为实现特定的信息系统功能的手段。在计算机系统的层次结构中，应用计算机是多级计算机系统层次结构的最终目标，是高级语言级之上的服务层次。

第二章

1.（1）$-35=(-100011)_2$

[−35]原＝10100011

[−35]补＝11011100

[−35]反＝11011101

（2）

$[127]_{原}=01111111$

$[127]_{反}=01111111$

$[127]_{补}=01111111$

（3）$-127=(-1111111)_2$

[−127]原＝11111111

[−127]补＝10000001

[−127]反＝10000000

（4）$-1=(-00000001)_2$

[−1]原＝10000001

[−1]补＝11111111

[−1]反＝11111110

2．[x]补 $=a_0.a_1a_2\cdots a_6$

解法一、

（1） 若 $a_0=0$，则 $x>0$，也满足 $x>-0.5$

此时 $a_1 \rightarrow a_6$ 可任意

（2） 若 $a_0=1$，则 $x<=0$，要满足 $x>-0.5$，需 $a_1=1$

即 $a_0=1, a_1=1, a_2 \rightarrow a_6$ 有一个不为 0

解法二、

$-0.5=-0.1_{(2)}=-0.100000=1,100000$

（1） 若 $x>=0$，则 $a0=0, a_1 \rightarrow a_6$ 任意即可

[x]补 = x = $a_0.a_1a_2\cdots a_6$

（2） 若 x < 0, 则 x > -0.5

只需-x < 0.5, -x > 0

[x]补 = -x, [0.5]补 = 01000000

即[-x]补 < 01000000

$$\overline{a_0} * \overline{a_1} * \overline{a_2} \ldots \overline{a_6} + 1 < 01000000$$

$$\overline{a_0} * \overline{a_1} * \overline{a_2} \ldots \overline{a_6} < 00111111$$

$$a_0 a_1 a_2 \ldots a_6 > 11000000$$

即 $a_0a_1 = 11$, $a_2 \rightarrow a_6$ 不全为 0 或至少有一个为 1（但不是“其余取 0”）

3．字长 32 位浮点数，阶码 8 位，用移码表示，尾数 23 位，用补码表示，基为 2

Es	$E_1 \rightarrow E_8$	Ms	M_{21}	M_0

（1） 最大的数的二进制表示

E = 11111111

Ms = 0, M = 11…1（全 1）

1 11111111 01111111111111111111111

（2） 最小的二进制数

E = 11111111

Ms = 1, M = 00…0（全 0）

1 11111111 10000000000000000000000

（3） 规格化范围

正最大 E = 11…1（8 个），M = 11…1（22 个），Ms = 0

即：$2^{2^7-1} \times (1 - 2^{-22})$

正最小 E = 00…0（8 个），M = 100…0（21 个），Ms = 0

即：$2^{-2^7} \times 2^{-1}$

负最大 E = 00…0（8 个），M = 011…1（21 个），Ms = 1

（最接近 0 的负数）即：$-2^{-2^7} \times (2^{-1} + 2^{-22})$

负最小 E = 11…1（8 个），M = 00…0（22 个），Ms = 1

即：$2^{2^7-1} \times (-1)$

规格化所表示的范围用集合表示为：

$$[2^{-2^7} \times 2^{-1} \ , \ 2^{2^7-1} \times (1 - 2^{-22})] \cup [2^{2^7-1} \times (-1) \ , -2^{-2^7} \times (2^{-1} + 2^{-22})]$$

4．在 IEEE754 标准中，一个规格化的 32 位浮点数 x 的真值表示为：

$$X = (-1)^S \times (1.M) \times 2^{E-127}$$

（1）$27/64=0.011011=1.1011\times 2^{-2}$

E= -2+127 = 125= 0111 1101　　S= 0　　M= 1011 0000 0000 0000 0000 000

最后表示为：0 01111101 10110000000000000000000

（2）$-27/64=-0.011011=1.1011\times 2^{-2}$

E= -2+127 = 125= 0111 1101　　S= 1　　M= 1011 0000 0000 0000 0000 000

最后表示为：1 01111101 10110000000000000000000

5．（1）用变形补码进行计算：

[x]补=00 11011 [y]补=00 00011

```
[x]补 =      00 11011
[y]补 =  + 00 00011
[x+y]补=     00 11110
```

结果没有溢出，x+y=11110

(2) [x]补=00 11011 [y]补=11 01011

```
[x]补 =      00 11011
[y]补 =  + 11 01011
[x+y]补=     00 00110
```

结果没有溢出，x+y=00110

（3）[x]补=11 01010 [y]补=11 111111

```
[x]补 =      00 01010
[y]补 =  + 00 11111
[x+y]补=     11 01001
```

结果没有溢出，x+y=-10111

6．[x-y]补=[x]补+[-y]补

（1）[x]补=00 11011　　[-y]补=00 11111

```
[x]补 =      00 11011
[-y]补 =  + 00 11111
[x-y]补=     01 11010
```

结果有正溢出，x-y=11010

（2）[x]补=00 10111　　[-y]补=11 00101

```
[x]补 =      00 10111
[-y]补 =  + 11 00101
[x-y]补=     11 11100
```

结果没有溢出，x-y=-00100

（3）[x]补=00 11011　　[-y]补=00 10011

[x]补 =　　00 11011
[-y]补 =　+ 00 10011
[x-y]补=　　01 01110

结果有正溢出，x-y=10010

7.（1） 用原码阵列乘法器：

[x]原=0 11011 [y]原=1 11111

因符号位单独考虑，|x|=11011 |y|=11111

```
                    1   1   0   1   1
                ×)  1   1   1   1   1
  ------------------------------------------
                    1   1   0   1   1
                1   1   0   1   1
            1   1   0   1   1
        1   1   0   1   1
    1   1   0   1   1
  ------------------------------------------
1   1   0   1   0   0   0   1   0   1
```

[x×y]原=1 1101000101

用补码阵列乘法器：

[x]补=0 11011 [y]补=1 00001

乘积符号位为：1

|x|=11011　|y|=11111

```
                    1   1   0   1   1
                ×)  1   1   1   1   1
  ------------------------------------------
                    1   1   0   1   1
                1   1   0   1   1
            1   1   0   1   1
        1   1   0   1   1
```

```
    1   1   0   1   1
  ---------------------------------------------------------------
1   1   0   1   0   0   0   1   0   1
```

[x×y]补=1 0010111011

(2) 用原码阵列乘法器：

[x]原=1 11111 [y]原=1 11011

因符号位单独考虑，|x|=11111 |y|=11011

```
                    1   1   1   1   1
                ×)  1   1   0   1   1
  ---------------------------------------------------------------
                    1   1   1   1   1
                1   1   1   1   1
            0   0   0   0   0
        1   1   1   1   1
    1   1   1   1   1
  ---------------------------------------------------------------
1   1   0   1   0   0   0   1   0   1
```

[x×y]原=0 1101000101

用补码阵列乘法器：

[x]补=1 00001 [y]补=1 00101

乘积符号位为：1

|x|=11111 |y|=11011

```
                    1   1   1   1   1
                ×)  1   1   0   1   1
  ---------------------------------------------------------------
                    1   1   1   1   1
                1   1   1   1   1
            0   0   0   0   0
        1   1   1   1   1
```

```
          1   1   1   1   1
        ---------------------------------
    1   1   0   1   0   0   0   1   0   1
```

[x×y]补=0 1101000101

8．(1) [x]原=[x]补=0 11000　　[-|y|]补=1 00001

```
被除数 X    0 11000
+[-|y|]补   1 00001
-----------------------------------------
余数为负  1 11001 →q0=0
左移      1 10010
+[|y|]补    0 11111
-----------------------------------------
余数为正  0 10001 →q1=1
左移      1 00010
+[-|y|]补   1 00001
-----------------------------------------
余数为正  0 00011 →q2=1
左移      0 00110
+[-|y|]补   1 00001
-----------------------------------------
余数为负  1 00111 →q3=0
左移      0 01110
+[|y|]补    0 11111
-----------------------------------------
余数为负  1 01101 →q4=0
左移      0 11010
+[|y|]补    0 11111
-----------------------------------------
余数为负  1 11001 →q5=0
+[|y|]补    0 11111
-----------------------------------------
余数  0 11000
```

故 [x÷y]原=1.11000 即 x÷y= -0.11000
余数为 0 11000

(2)　[|x|]补=0 01011　　[-|y|]补=1 00111

```
被除数 X    0 01011
+[-|y|]补   1 00111
-----------------------------------------
余数为负  1 10010 →q0=0
```

左移 1 00100
+[|y|]补 0 11001

余数为负 1 11101 →q1=0
左移 1 11010
+[|y|]补 0 11001

余数为正 0 10011 →q2=1
左移 1 00110
+[-|y|]补 1 00111

余数为正 0 01101 →q3=1
左移 0 11010
+[-|y|]补 1 00111

余数为正 0 00001 →q4=1
左移 0 00010
+[-|y|]补 1 00111

余数为负 1 01001 →q5=0
+[|y|]补 0 11001

余数 0 00010

x÷y= −0.01110
余数为 0 00010

9．(1) $x = 2^{-011}*0.100101$, $y = 2^{-010}*(-0.011110)$

$[x]_{浮}$ = 11101,0.100101

$[y]_{浮}$ = 11110,-0.011110

Ex-Ey = 11101+00010=11111

$[x]_{浮}$ = 11110,0.010010(1)

x+y
0 0. 0 1 0 0 1 0 (1)
+ 1 1. 1 0 0 0 1 0
1 1. 1 1 0 1 0 0 (1)

规格化处理: 1.010010 阶码 11100

x+y= $1.010010*2^{-4} = 2^{-4}*-0.101110$

x-y
0 0. 0 1 0 0 1 0 (1)
+ 0 0. 0 1 1 1 1 0
0 0 1 1 0 0 0 0 (1)

规格化处理: 0.110000 阶码 11110

x-y=$2^{-2}*0.110001$

(2) $x = 2^{-101}*(-0.010110)$, $y = 2^{-100}*0.010110$

$[x]_{浮}$= 11011,-0.010110

$[y]_{浮}$= 11100,0.010110

Ex-Ey = 11011+00100 = 11111

$[x]_{补}$= 11100,1.110101(0)

x+y

```
    1 1. 1 1 0 1 0 1
+   0 0. 0 1 0 1 1 0
-------------------
    0 0. 0 0 1 0 1 1
```

规格化处理: 0.101100 阶码 11010

$x+y= 0.101100*2^{-6}$

x-y

```
    1 1.1 1 0 1 0 1
+   1 1.1 0 1 0 1 0
-------------------
    1 1.0 1 1 1 1 1
```

规格化处理: 1.011111 阶码 11100

$x-y=-0.100001*2^{-4}$

10. (1) Ex = 0011, Mx = 0.110100

Ey = 0100, My = 0.100100

Ez = Ex+Ey = 0111

Mx*My

```
      0. 1 1 0 1
  *   0.1 0 0 1
-----------------
      0 1 1 0 1
    0 0 0 0 0
  0 0 0 0 0
0 1 1 0 1
0 0 0 0 0
-----------------
0 0 1 1 1 0 1 0 1
```

规格化： $2^{6}*0.111011$

(2) Ex = 1110, Mx = 0.011010

Ey = 0011, My = 0.111100

Ez = Ex-Ey = 1110+1101 = 1011

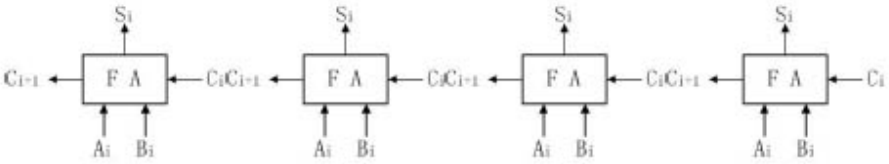
[Mx]补 = 00.011010

[My]补 = 00.111100, [-My]补 = 11.000100

```
          0 0 0 1 1 0 1 0
+[-My]    1 1 0 0 0 1 0 0
          1 1 0 1 1 1 1 0        0
          1 0 1 1 1 1 0 0
+[My]     0 0 1 1 1 1 0 0
          1 1 1 1 1 0 0 0        0.0
          1 1 1 1 0 0 0 0
+[My]     0 0 1 1 1 1 0 0
          0 0 1 0 1 1 0 0        0.01
          0 1 0 1 1 0 0 0
+[-My]    1 1 0 0 0 1 0 0
          0 0 0 1 1 1 0 0        0.011
          0 0 1 1 1 0 0 0
+[-My]    1 1 0 0 0 1 0 0
          1 1 1 1 1 1 0 0      0.0110
          1 1 1 1 1 0 0 0
+[My]      0 0 1 1 1 1 0 0
          0 0 1 1 0 1 0 0       0.01101
          0 1 1 0 1 0 0 0
+[-My]     1 1 0 0 0 1 0 0
          0 0 1 0 1 1 0 0     0.01101
```

商 $=0.110110*2^{-6}$，　余数$=0.101100*2^{-6}$

11.



4 位加法器如上图，

$$C_i = A_iB_i + A_iC_{i-1} + B_iC_{i-1}$$
$$= A_iB_i + (A_i + B_i)C_{i-1}$$
$$= A_iB_i + (A_i \oplus B_i)C_{i-1}$$

(1)串行进位方式

$C_1 = G_1+P_1C_0$　　其中：$G_1 = A_1B_1$　　$P1 = A_1 \oplus B_1$（A_1+B_1也对）

$C_2 = G_2+P_2C_1$　　$G_2 = A_2B_2$　　$P_2 = A_2 \oplus B_2$

$C_3 = G_3+P_3C_2$　　$G_3 = A_3B_3$　　$P_3 = A_3 \oplus B_3$

$C_4 = G_4+P_4C_3$　　$G_4 = A_4B_4$　　$P_4 = A_4 \oplus B_4$

(2)并行进位方式

$C_1 = G_1+P_1C_0$

$C_2 = G_2+P_2G_1+P_2P_1C_0$

$C_3 = G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1C_0$

$C_4 = G_4+P_4G_3+P_4P_3G_2+P_4P_3P_2G_1+P_4P_3P_2P_1C_0$

12. (1)组成最低四位的 74181 进位输出为：

$C_4 = C_{n+4} = G+PC_n = G+PC_0$， C_0 为向第 0 位进位

其中，$G = y_3+y_2x_3+y_1x_2x_3+y_0x_1x_2x_3$，$P = x_0x_1x_2x_3$，所以

$C_5 = y_4+x_4C_4$

$C_6 = y_5+x_5C_5 = y_5+x_5y_4+x_5x_4C_4$

(2)设标准门延迟时间为 T，“与或非”门延迟时间为 1.5T，则进位信号 C_0，由最低位传送至 C_6 需经一个反相器、两级“与或非”门，故产生 C_0 的最长延迟时间为

$T+2*1.5T = 4T$

(3)最长求和时间应从施加操作数到 ALU 算起：第一片 74181 有 3 级“与或非”门（产生控制参数 x_0, y_0, C_{n+4}），第二、三片 74181 共 2 级反相器和 2 级“与或非”门（进位链），第四片 74181 求和逻辑（1 级与或非门和 1 级半加器，设其延迟时间为 3T），故总的加法时间为：

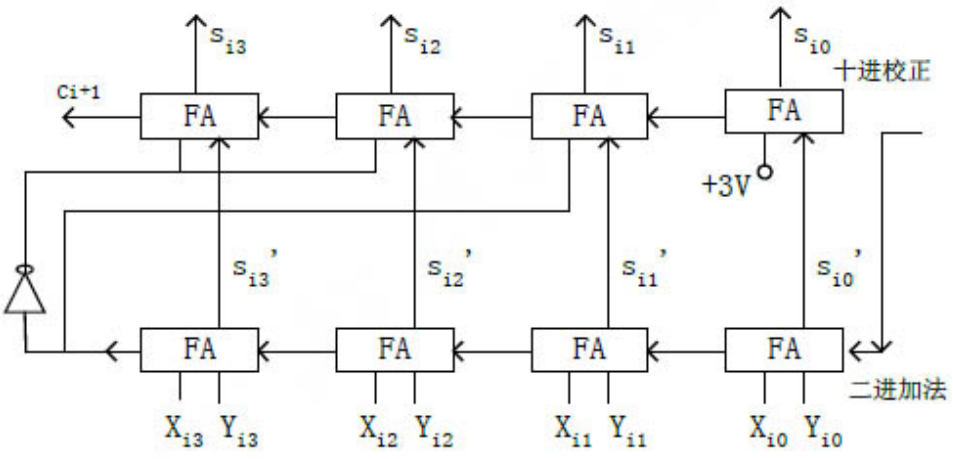
$t_0 = 3*1.5T+2T+2*1.5T+1.5T+3T = 14T$

13. 设余三码编码的两个运算数为 X_i 和 Y_i，第一次用二进制加法求和运算的和数为 S_i'，进位为 C_{i+1}'，校正后所得的余三码和数为 S_i，进位为 C_{i+1}，则有：

$X_i = X_{i3}X_{i2}X_{i1}X_{i0}$

$Y_i = Y_{i3}Y_{i2}Y_{i1}Y_{i0}$

$S_i' = S_{i3}'S_{i2}'S_{i1}'S_{i0}'$



当 $C_{i+1}' = 1$ 时，$S_i = S_i'+0011$

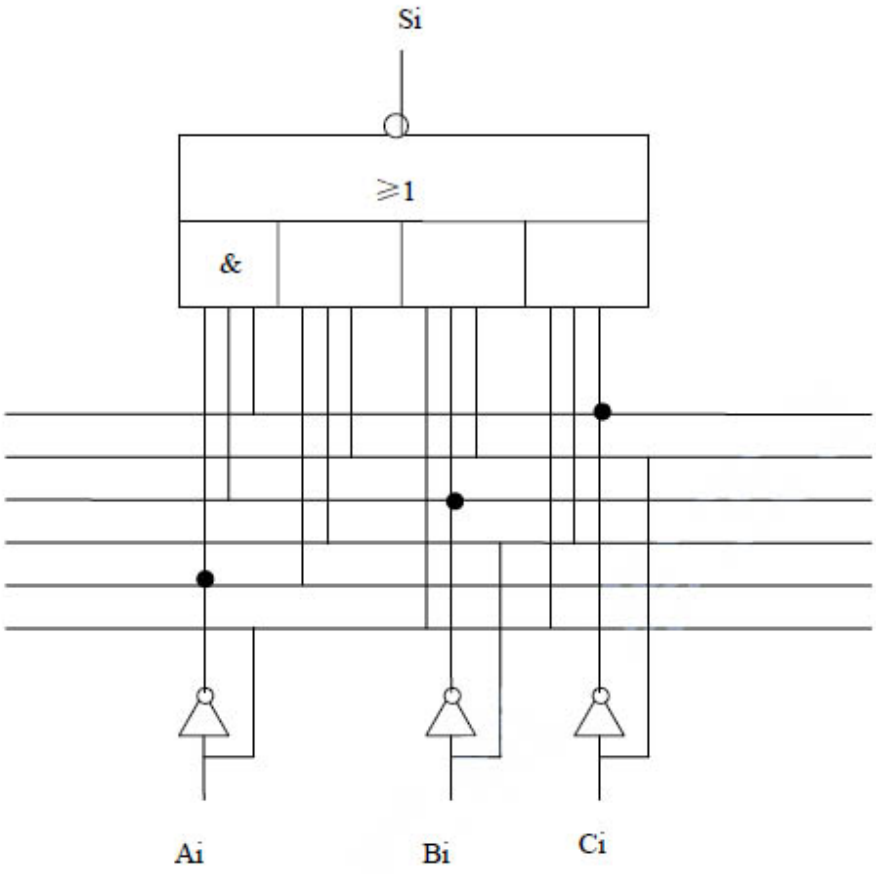
当 $C_{i+1}' = 0$ 时，$S_i = S_i'+1101$

} 并产生 C_{i+1}

根据以上分析，可画出余三码编码的十进制加法器单元电路如图所示。

14. $Si=\overline{A}i\overline{B}iCi+\overline{A}iBi\overline{C}i+A\overline{i}\overline{B}i\overline{C}i+AiBiCi$

图如下：



15. 设计思想：电路由三部分构成：ALU 完成定点加减法运算和逻辑运算，专用的阵列乘法器完成乘法运算，专用的阵列除法器完成除法操作。逻辑图可参考主教材图 2.7 和图 2.9。

16. 设计思想：因为有八种运算，所以控制信号采用三位，S0,S1,S2。加法和减法操作利用 4 位补码加减法器完成；加 1 操作可以单独设计电路实现，也可以将被加数强制为+1 利用加减法器实现；传送操作可以利用加减法器实现，第二加数强制为 0；逻辑乘和取反操作可设计单独的逻辑运算电路，用与门和反相器实现；取补电路单独设计，参见主教材图 2.6；乘法操作可单独设计高速乘法器，电路参见主教材图 2.7。

17. 设计思想：将 74181 的 S3～S0 及 M 等五个控制信号缩减为 S2～S0 三根信号，主教材表 2.5（功能表中的算术运算和逻辑运算相应进行简化，去除冗余操作和可替代操作：

000: 逻辑 0

001: AB

010: A+B

011: A⊕B

100: A 加 B

101: A 减 B 减 1

110: A 加 A

111: A

其中，000～011 为四种逻辑运算，100～111 为四种算术运算。根据功能表可以很容易地设计出简化的函数发生器。

第三章

1. (1) $2^{20} * \frac{32}{8} = 4M$字节

(2) $\frac{1024K * 32}{512K * 8} = 2 * 4 = 8$片

(3)1 位地址作芯片选择

2. (1) $2^{26}/2^{24}$=4（块）

(2) ($2^{24}/2^{22}$)×(64 位/8 位)=32（片）

(3)主存共需 DRAM 芯片为：4×32=128 (片)

每个内存条有 32 片 DRAM 芯片，容量为 16M×64 位，需 24 根地址线(A23~A0)完成内存条内存储单元寻址。一共有 4 块内存条，采用 2 根高位地址线(A25~A24)，通过 2：4 译码器译码产生片选信号对各模块板进行选择。

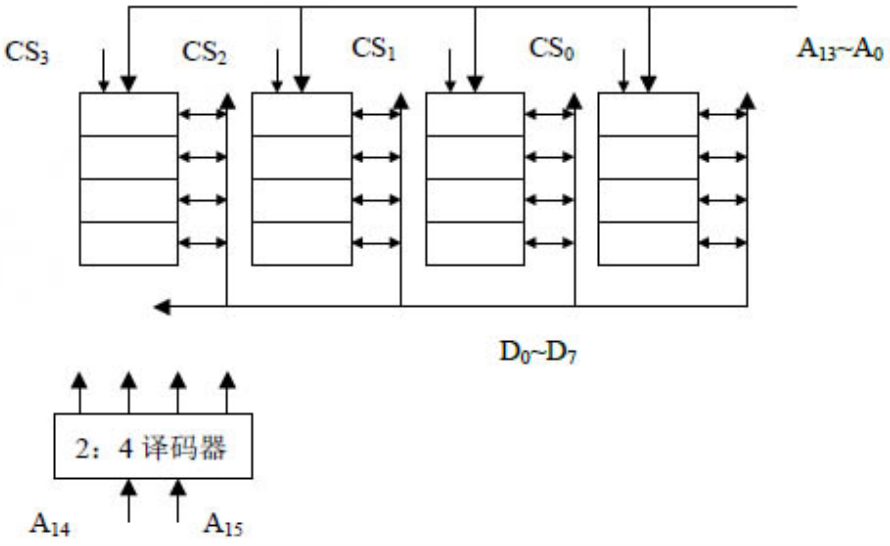
3. (1)根据题意，存储总容量为 64KB，故地址总线需 16 位。现使用 16K*8 位 DRAM 芯片，共需 16 片。芯片本身地址线占 14 位，所以采用位并联与地址串联相结合的方法来组成整个存储器，其组成逻辑图如图所示，其中使用一片 2：4 译码器。

(2)根据已知条件，CPU 在 1us 内至少访存一次，而整个存储器的平均读/写周期为 0.5us，如果采用集中刷新，有 64us 的死时间，肯定不行

如果采用分散刷新，则每 1us 只能访存一次，也不行所以采用异步式刷新方式。

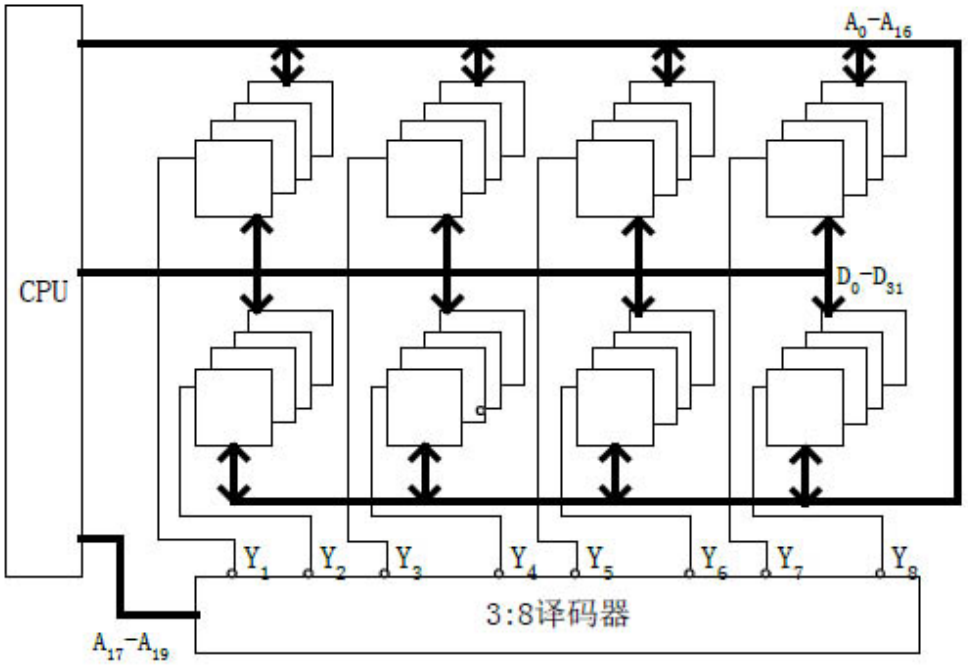
假定 16K*1 位的 DRAM 芯片用 128*128 矩阵存储元构成，刷新时只对 128 行进行异步方式刷新，则刷新间隔为 2ms/128＝15.6us，可取刷新信号周期 15us。

刷新一遍所用时间＝15us×128＝1.92ms



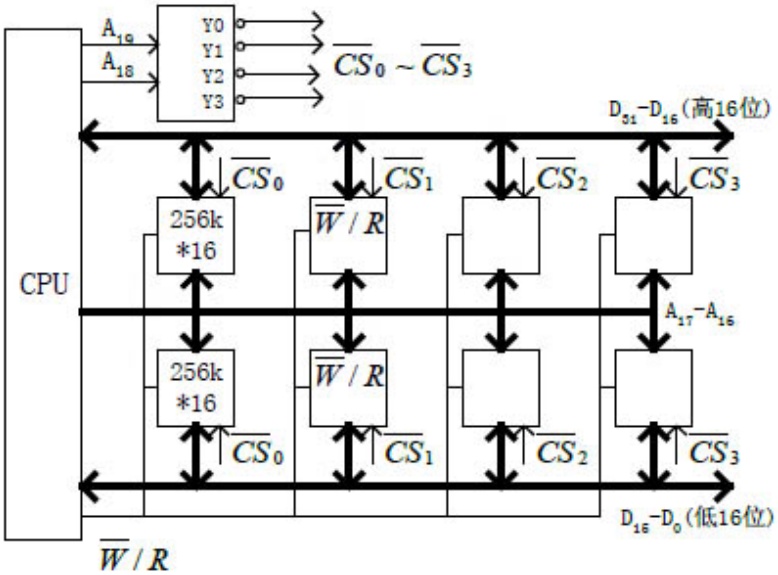
4. (1) $\frac{1024K * 32}{128K * 8} = 32$片

(2)



(3)如果选择一个行地址进行刷新，刷新地址为 A_0-A_8，因此这一行上的 2048 个存储元同时进行刷新，即在 8ms 内进行 512 个周期。刷新方式可采用：在 8ms 中进行 512 次刷新操作的集中刷新方式，或按 8ms/512 = 15.5us 刷新一次的异步刷新方式。

5. 所设计的存储器单元数为 1M，字长为 32，故地址长度为 20 位（A19~A0），所用芯片存储单元数为 256K，字长为 16 位，故占用的地址长度为 18 位（A17~A0）。由此可用位并联方式与地址串联方式相结合的方法组成组成整个存储器，共 8 片 RAM 芯片，并使用一片 2：4 译码器。其存储器结构如图所示。

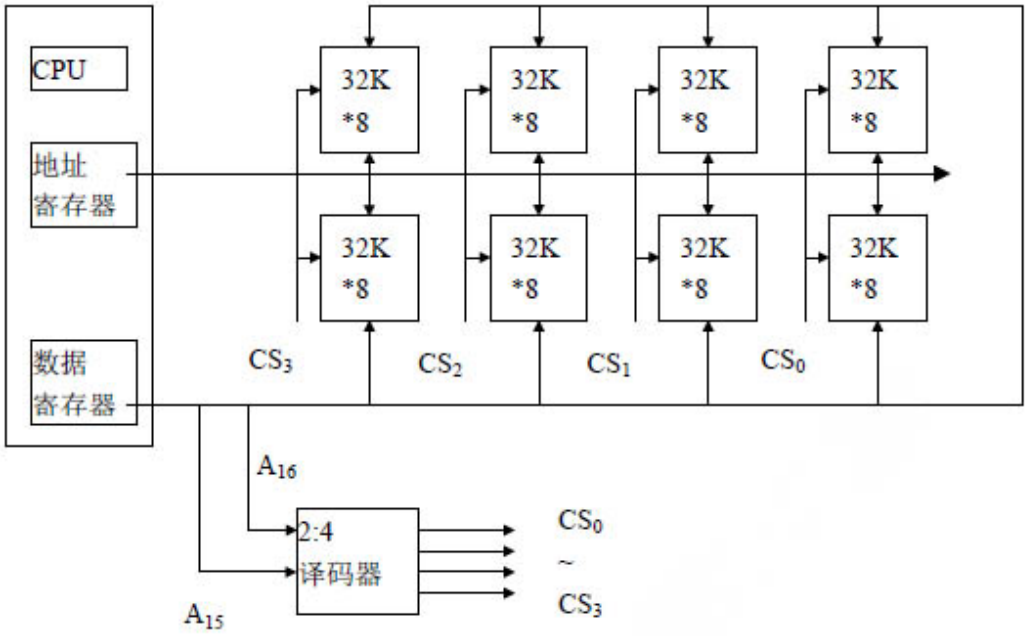


6.（1）系统 16 位数据，所以数据寄存器 16 位

（2）系统地址 128K＝2^{17}，所以地址寄存器 17 位

（3） 共需要 8 片

（4） 组成框图如下

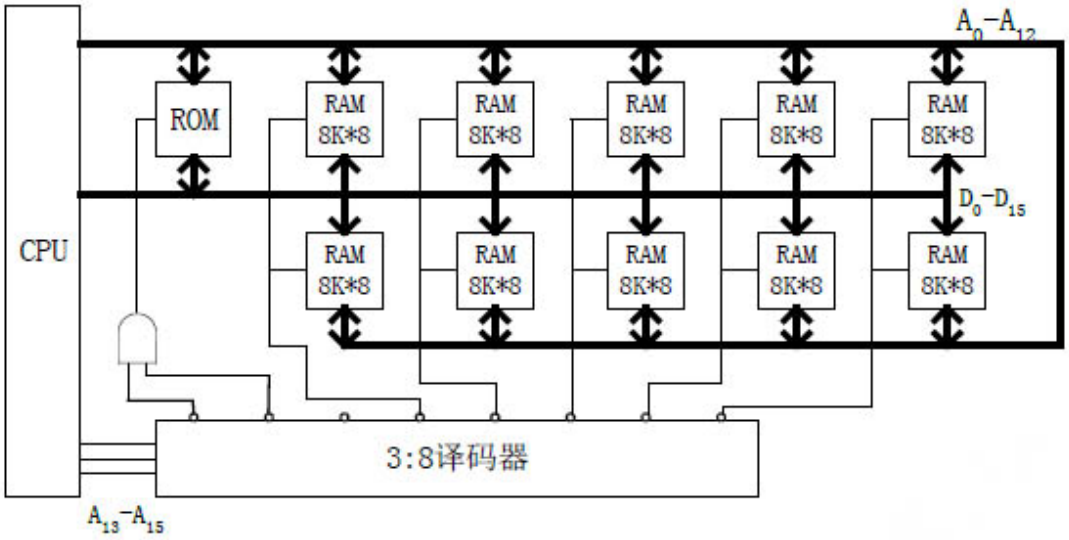


7.(1)组内地址用 A_{12}~A_0

(2)小组译码器使用 3：8 译码器

(3)RAM_1~RAM_5 各用两片 8K*8 的芯片位并联连接

0000H	ROM
4000H	
6000H	RAM_1
8000H	RAM_2
A000H	RAM_3
C000H	RAM_4
E000H	RAM_5



8.顺序存储器和交叉存储器连续读出 m = 8 个字的信息总量都是：

q = 64 位*8 = 512 位

顺序存储器和交叉存储器连续读出 8 个字所需的时间分别是：

$t_1 = mT = 8*100ns = 8*10^{-7}s$

$$t_2 = T + (m-1)\tau = 100ns + 7*50ns = 450ns = 4.5*10^{-7}ns$$

顺序存储器和交叉存储器的带宽分别是：

$$W_1 = q/t_1 = 512 \div (8*10^{-7}) = 64\times10^7[位/s]$$

$$W_2 = q/t_2 = 512 \div (4.5*10^{-7}) = 113.8\times10^7[位/s]$$

9.cache 的命中率

$$H = \frac{N_c}{N_c + N_m} = \frac{2420}{2420+80} = 0.968$$

$$r = \frac{T_m}{T_c} = \frac{240}{40} = 6$$

cache/主存系统效率 e 为

$$e = \frac{1}{r + (1-r)H} * 100\% = \frac{1}{6 + (1-6)*0.968} * 100\% = 86.2\%$$

平均访问时间 T_a 为

$$T_a = \frac{T_c}{e} = \frac{40ns}{0.862ns} = 46.4ns$$

10. $h*t_c+(1-h)*t_m = t_a$

$$h = \frac{t_a - t_m}{t_c - t_m} = \frac{50-200}{40-200} = 93.75\%$$

11.设取指周期为 T，总线传送周期为 τ，指令执行时间为 t_0

(1)$t = (T+5\tau+6t_0)*80 = 80T+400\tau+480t_0$

(2) $t = (T+7\tau+8t_0)*60 = 60T+420\tau+480t_0$

故不相等。

12.D

第四章

1.不合理。指令最好半字长或单字长，设 16 位比较合适。

2.70 条指令，所以操作码至少为 7 位。

双操作数指令格式可以为：

7	12	12

单操作数指令格式可以为：

7	25

无操作数指令格式可以为：

7	—

3.(1)RR 型指令

(2)寄存器寻址

(3)单字长二地址指令

(4)操作码字段 OP 可以指定 2^6=64 种操作

4.(1)双字长二地址指令，用于访问存储器。操作码字段可指定 64 种操作。

(2)RS 型指令，一个操作数在通用寄存器（共 16 个），另一个操作数在主存中。

(3)有效地址可通过变址寻址求得，即有效地址等于变址寄存器（共 16 个）内容加上位移量。

5.(1)双操作数指令

(2)2^3=8 种寻址方式

(3)2^4=16 种操作

6.(1)直接寻址方式

(2)相对寻址方式

(3)变址寻址方式

(4)基址寻址方式

(5)间接寻址方式

(6)基址间接寻址方式

7.40 条指令至少需要操作码字段 6 位，所以剩下的长度为 26 位。主存的容量为 64M 字，则设寻址模式（X）2 位，格式如下：

31 26	25 24	23 0
OP	X	D

X= 0 0 直接寻址 有效地址 E=D

X= 0 1 立即寻址 D 字段为立即数

X= 1 0 变址寻址 有效地址 E= (RX)+D （可寻址 64M 个存储单元）

X= 1 1 相对寻址 有效地址 E=（PC）+D （可寻址 64M 个存储单元）

其中 RX 为变址寄存器（32 位），PC 为程序计数器（32 位）。在相对寻址时，位移量 D

可正可负。

8.(1)50 种操作码占 6 位，4 种寻址方式占 2 位。以单地址指令为例：

OP（6）	X（2）	D（24）

X＝00　寄存器寻址方式。D 字段实际使用 4 比特选择 16 个通用寄存器。

X＝01　寄存器间接寻址方式。D 字段实际使用 4 比特选择 16 个通用寄存器。E= (RX)。

X＝10　立即寻址方式。D 字段给出 24 位立即数。

X＝11　直接寻址方式。D 字段给出 24 位内存地址。E＝D。

(2) 寻址模式字段变成 3 位，可以支持更多的寻址方式。可增加相对寻址方式，其有效地址 E＝PC+D；还可使用内存间接寻址，此时有效地址 E＝（D）。

9. 16 个通用寄存器占 4 位，64 种操作占 6 位，剩下 22 位用于存储器地址，

OP（6）	R（4）	D（22）

采用 R 为基址寄存器寻址，地址＝（R）＋D

当基址最大，D 也是最大的时候，寻址能力最大

而寄存器是 32 位的，

故最大存储空间是 $2^{32}+2^{22}$＝4GB＋4MB。

10. 表 4.9 的指令数为 29，则指令的操作码至少为 5 位。设这些指令支持立即寻址、寄存器寻址、直接寻址、堆栈寻址、相对寻址、内存间接寻址、寄存器间接寻址、变址寻址、基址寻址等 9 种寻址方式。并设计算机字长为 32 位：

6	4	8	4	8
OP	目标寻址方式	目标操作数	源寻址方式	源操作数

11.C

12.(1)寄存器

(2)寄存器间接

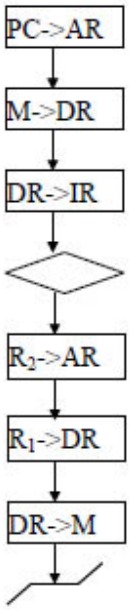
(3)立即

(4)直接

(5)相对、基址、变址

第五章

1. （1）IR、(2)AR、(3)DR、通用寄存器
2. STO R_1，(R_2)



PC_0, G, AR_i

R/$\overline{W}$=R

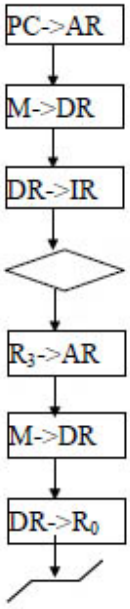
DR_0, G, AR_i

$R2_0$, G, AR_i

$R1_0$, G, DR_i

R/$\overline{W}$=W

3. LAD (R_3), R_0

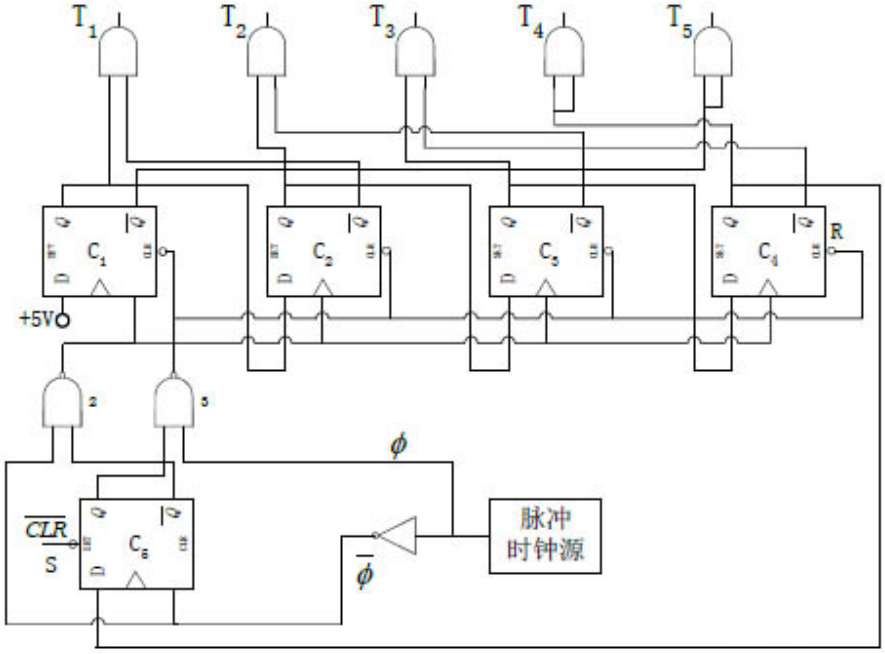


$R3_0$, G, AR_i

R/$\overline{W}$=R

DR_0, G, $R0_i$

4.

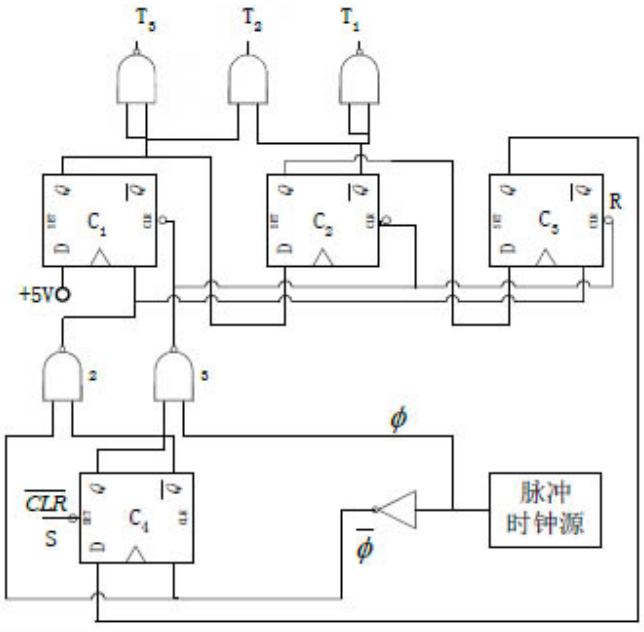


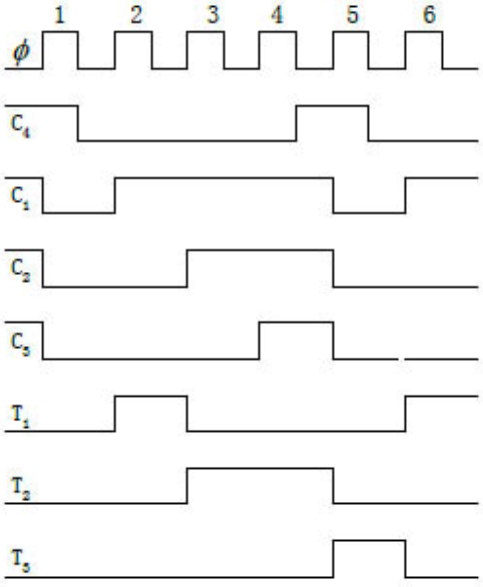
5.节拍脉冲 T_1，T_2，T_3 的宽度实际上等于时钟脉冲的周期或是它的倍数。此处 $T_1 = T_2 = 200ns$，$T_3 = 400ns$，所以主脉冲源的频率应为 $f = \frac{1}{T} = 5MHz$ 。

为了消除节拍脉冲上的毛刺，环形脉冲发生器采用移位寄存器形式。图中画出了题目要求的逻辑电路图与时序信号关系图。根据时序信号关系，T_1，T_2，T_3 三个节拍脉冲的逻辑表达式如下：

$$T_1 = C_1 * \overline{C_2} \qquad T_2 = C_2 \qquad T_3 = \overline{T_1}$$

T_1 用与门实现，T_2 和 T_3 则用 C_2 的 $\overline{Q}$ 端和 C_1 的 Q 端加非门实现，其目的在于保持信号输出时延时间的一致性并与环形脉冲发生器隔离。





6. $(80*3+1)*\frac{32}{8}=964$字节

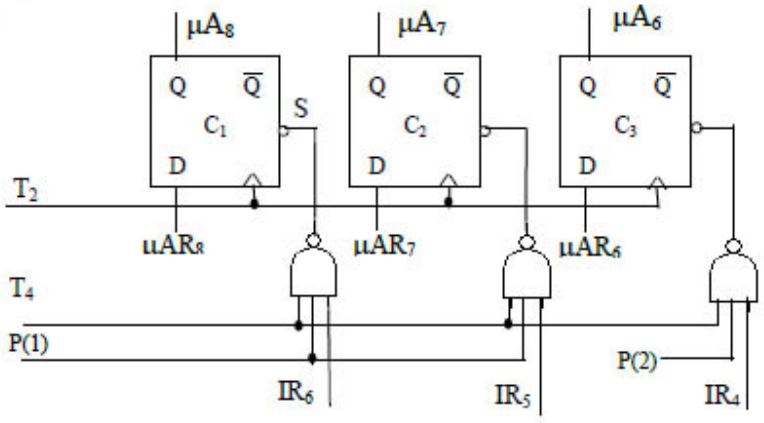
7. M = G
   S3 = H+D+F
   S2 = A+B+H+D+E+F+G
   S1 = A+B+F+G
   C = H+D+Ey+Fy+G ϕ

8. 经分析，（d, i, j）和（e, f, h）可分别组成两个小组或两个字段，然后进行译码，可得六个微命令信号，剩下的 a, b, c, g 四个微命令信号可进行直接控制，其整个控制字段组成如下：

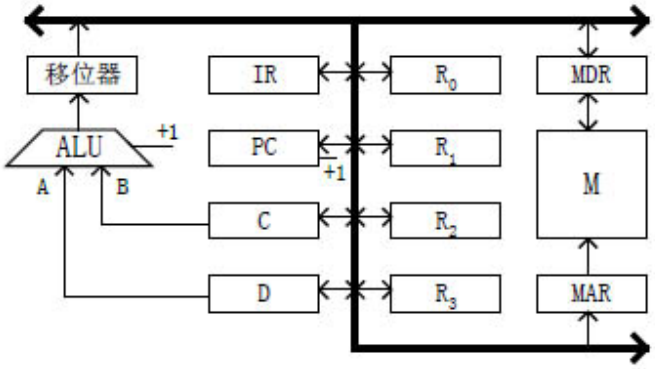
＊＊＊＊	＊＊	＊＊
a b c g	01 d	01 e
	10 i	10 f
	11 j	11 h

9. P1 = 1，按 IR6、IR5 转移
   P2 = 1，按进位 C 转移

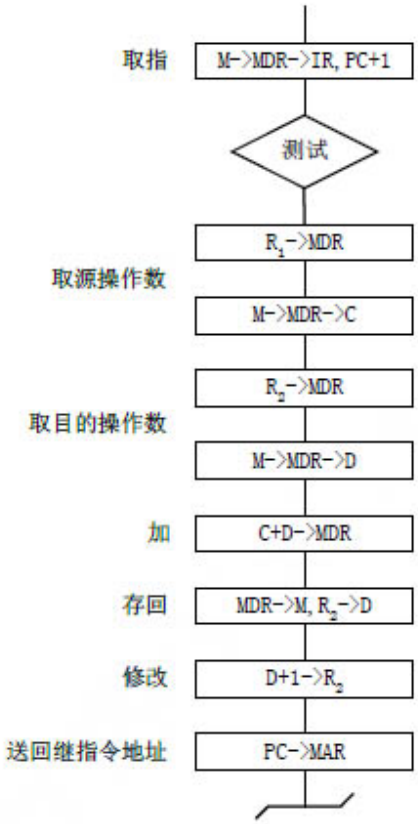
微地址转移逻辑图：



10. (1)将 C，D 两个暂存器直接接到 ALU 的 A，B 两个输入端上。与此同时，除 C，D 外，其余 7 个寄存器都双向接到单总线上。

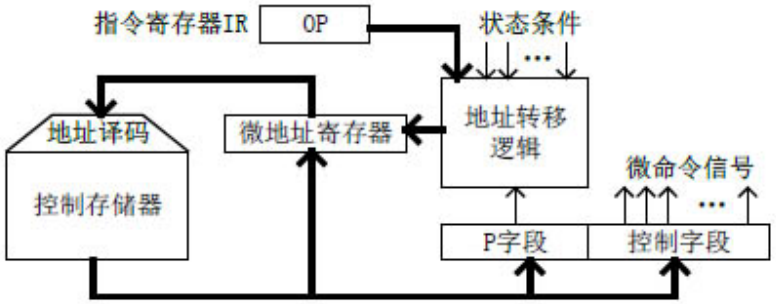


(2)



11. (1)假设判别测试字段中每一位作为一个判别标志，那么由于有 4 个转移条件，故该字段为 4 位。下地址字段为 9 位，因为控存容量为 512 单元。微命令字段则是（48-4-9）=35 位。

(2)对应上述微指令格式的微程序控制器逻辑框图如图所示。其中微地址寄存器对应下地址字，P 字段即为判别测试字段，控制字段即为微命令字段，后两部分组成微指令寄存器。地址转移逻辑的输入是指令寄存器的 OP 码、各种状态条件以及判别测试字段所给的判别标志（某一位为 1），其输出修改微地址寄存器的适当位数，从而实现微程序的分支转移。就是说，此处微指令的后继地址采用断定方式。



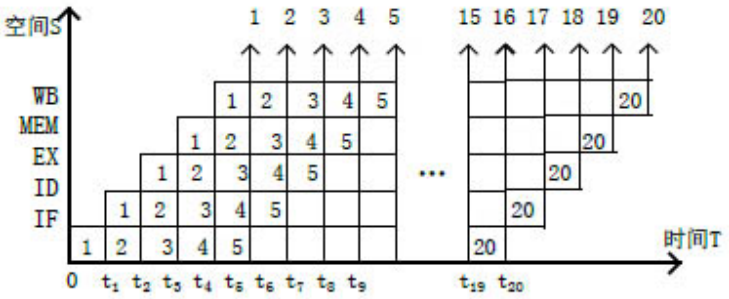
12. (1)流水线的操作周期应按各步操作的最大时间来考虑，即流水线时钟周期性

$$\tau = \max\{\tau_i\} = 100ns$$

(2)遇到数据相关时，就停顿第 2 条指令的执行，直到前面指令的结果已经产生，因此至少需要延迟 2 个时钟周期。

(3)如果在硬件设计上加以改进，如采用专用通路技术，就可使流水线不发生停顿。

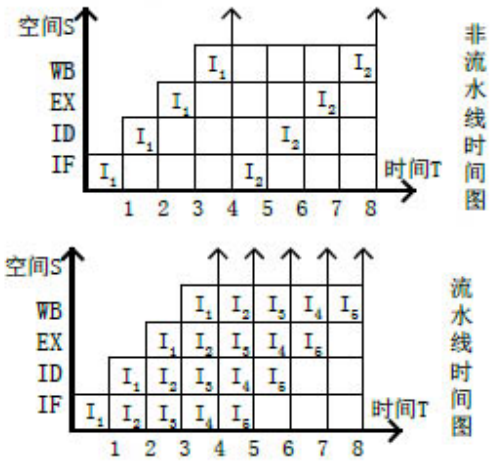
13. (1)



(2) $H = \dfrac{n}{(K+n-1)\tau} = \dfrac{20}{(5+20-1)*100*10^{-9}} = 8.33*10^6$ 条/秒

(3) $S = \dfrac{Ts}{Tp} = \dfrac{n\tau K}{(K+n-1)\tau} = \dfrac{20*5}{20+5-1} = 4.17$

14.



如上两图所示，执行相同的指令，在 8 个单位时间内，流水计算机完成 5 条指令，而非

流水计算机只完成 2 条，显然，流水计算机比非流水计算机有更高的吞吐量。

15. 证：设 n 条指令，K 级流水，每次流水时间 τ

则用流水实现　Tp＝K τ +(n-1)　τ

$$Hp = \frac{n}{Tp}$$

非流水实现　Ts＝K τ n

$$Hs = \frac{n}{Ts}$$

$$\frac{Hp}{Hs} = \frac{\frac{n}{Tp}}{\frac{n}{Ts}} = \frac{Ts}{Tp} = \frac{Kn\tau}{K\tau + (n-1)\tau} = \frac{Kn}{K+n-1} = \frac{K}{\frac{K-1}{n}+1}$$

n->∞时，　$\frac{Hp}{Hs} -> \infty$

n=1 时，　$\frac{Hp}{Hs} = 1$，　则可见 n>1 时 Ts>Tp，故流水线有更高吞吐量

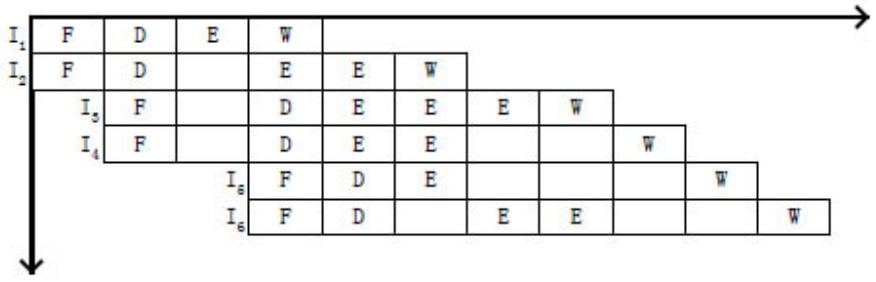
16.(1)写后读　RAW

(2)读后写　WAR

(3)写后写　WAW

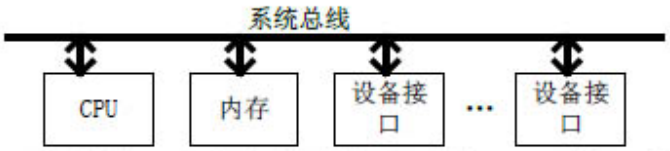
17.(1)

(2)

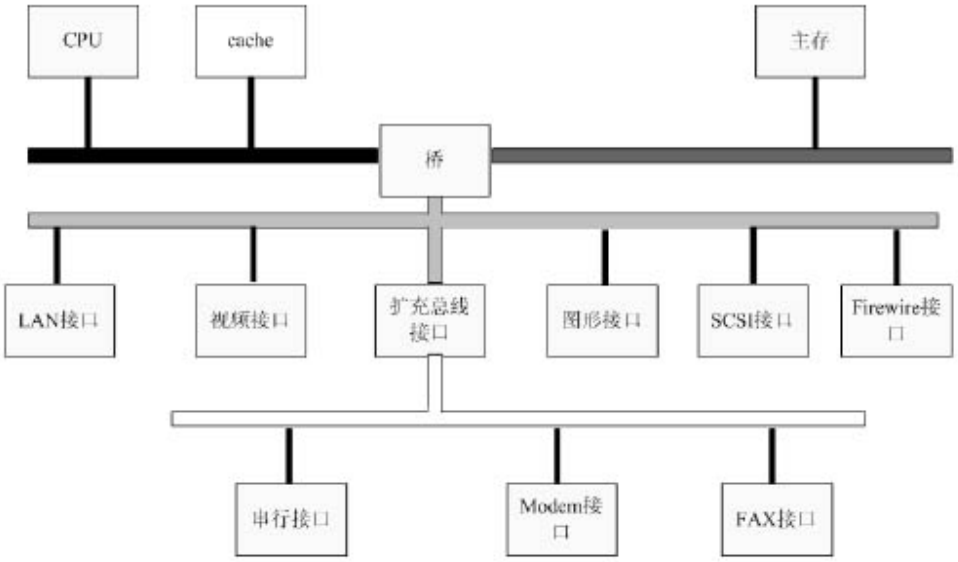


第六章

1. 单总线结构：它是一组总线连接整个计算机系统的各大功能部件，各大部件之间的所有的信息传送都通过这组总线。其结构如图所示。单总线的优点是允许 I/O 设备之间或 I/O 设备与内存之间直接交换信息，只需 CPU 分配总线使用权，不需要 CPU 干预信息的交换。所以总线资源是由各大功能部件分时共享的。单总线的缺点是由于全部系统部件都连接在一组总线上，所以总线的负载很重，可能使其吞量达到饱和甚至不能胜任的程度。故多为小型机和微型机采用。



多总线结构： 多总线系统结构是通过桥，CPU 总线，系统总线和高速总线彼此相连，各大部件的信息传送不是只通过系统总线；体现了高速，中速，低速设备连接到不同的总线上同时进行工作，以提高总线的效率和吞吐量，而且处理器结构的变化不影响高速总线。



2. (1)简化了硬件的设计。从硬件的角度看，面向总线是由总线接口代替了专门的 I/O 接口，由总线规范给出了传输线和信号的规定，并对存储器、I/O 设备和 CPU 如何挂在总线上都作了具体的规定，所以，面向总线的微型计算机设计只要按照这些规定制作 CPU 插件、存储器插件以及 I/O 插件等，将它们连入总线即可工作，而不必考虑总线的详细操作。

(2)简化了系统结构。整个系统结构清晰，连线少，底板连线可以印刷化。

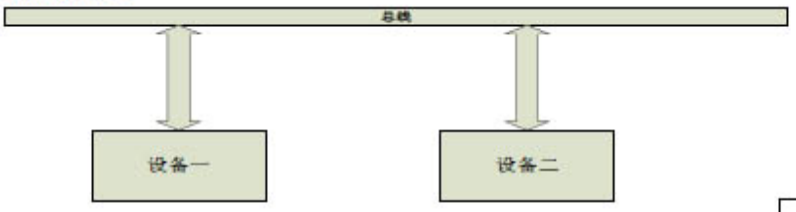
(3)系统扩充性好。一是规模扩充，二是功能扩充。规模扩充仅仅需要多插一些同类型的插件；功能扩充仅仅需要按总线标准设计一些新插件。插件插入机器的位置往往没有严格的限制。这就使系统扩充既简单又快速可靠，而且也便于查错。

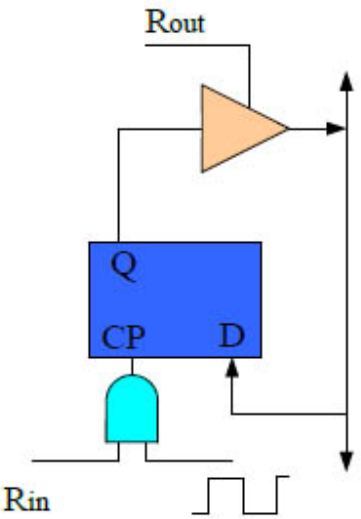
(4)系统更新性能好。因为 CPU、存储器、I/O 接口等都是按总线规约挂到总线上的，因而只要总线设计恰当，可以随时随着处理器芯片以及其他有关芯片的进展设计新的插件，新的插件插到底板上对系统进行更新，而这种更新只需更新需要更新的插件，其他插件

和底板连线一般不需更改。

3. “A”的 ASCII 码为 41H = 01000001B，1 的个数为偶数，故校验位为 0；“8”的 ASCII 码为 38H＝00111000B，1 的个数为奇数，故校验位为 1。

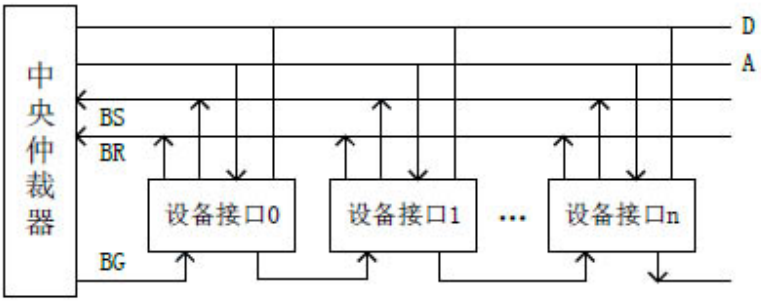
4. 逻辑图如下：



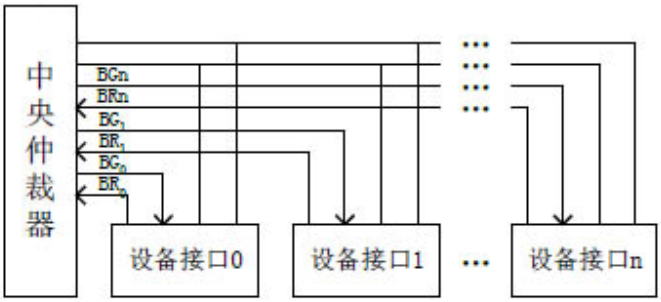


说明：两个设备共用总线，每个设备的总线接口部分如右图所示。通过锁存器保存接收数据，并通过三态门向总线发送数据。每个设备的 Rin 控制端有效时，锁存器保存接收数据；每个设备的 Rout 信号有效时，锁存器保存的数据被送上总线。当 Rout 信号无效时，设备与总线在电气上断开。

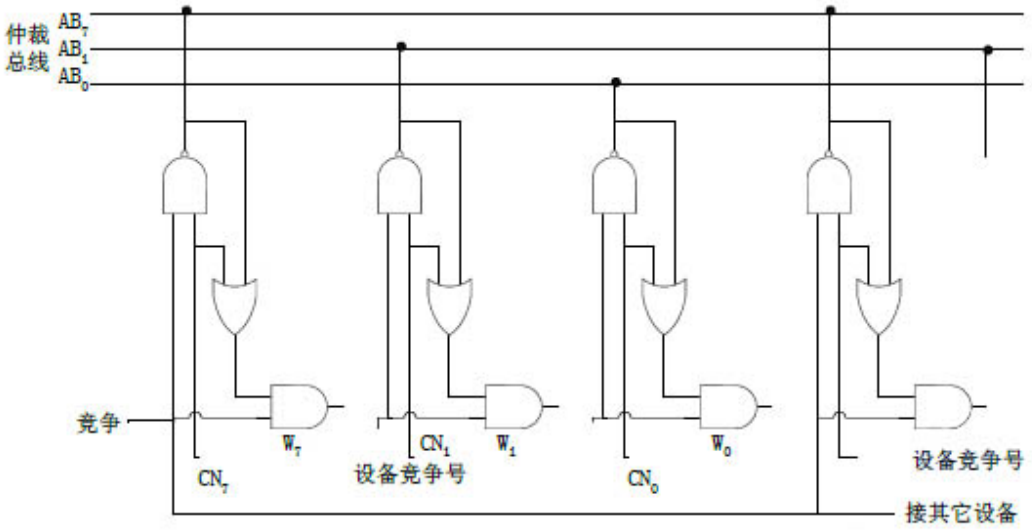
5.



6.



7.



8.C

9.B、A、C

10.A

11.D

12.A

13. 存储总线周期用于对内存读写，I/O 总线周期对接口中的端口进行读写。

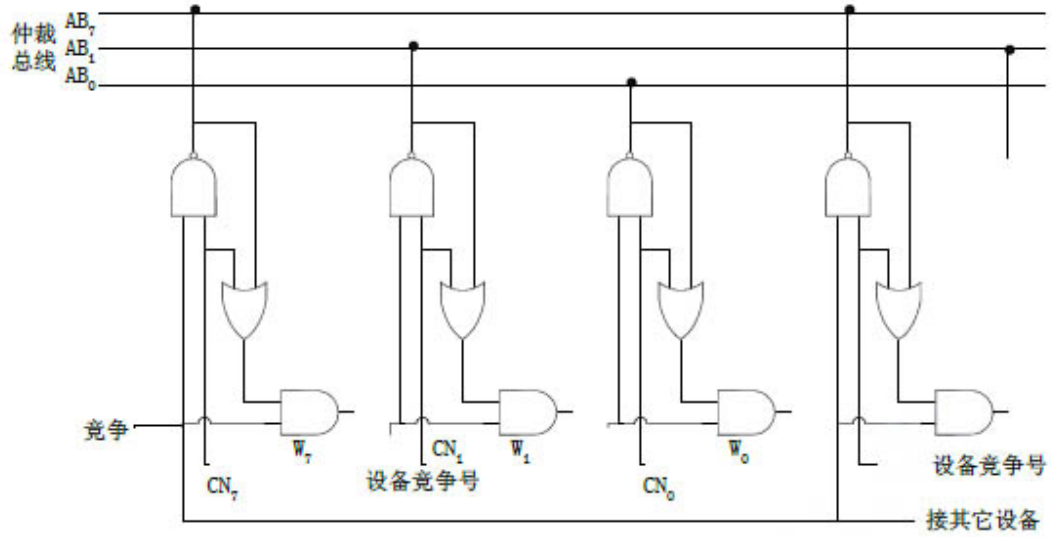
14.D、C、A、B

15.B、A、E、D、C

16.A、B、C、D

17. PCI 总线上有 HOST 桥、PCI/LAGACY 总线桥、PCI/PCI 桥。桥在 PCI 总线体系结构中起着重要作用，它连接两条总线，使彼此间相互通信。桥是一个总线转换部件，可以把一条总线的地址空间映射到另一条总线的地址空间上，从而使系统中任意一个总线主设备都能看到同样的一份地址表。桥可以实现总线间的猝发式传送，可使所有的存取都按 CPU 的需要出现在总线上。由上可见，以桥连接实现的 PCI 总线结构具有很好的扩充性和兼容性，允许多条总线并行工作。

18. 分布式仲裁不需要中央仲裁器，每个潜在的主方功能模块都有自己的仲裁号和仲裁器。当它们有总线请求时，把它们唯一的仲裁号发送到共享的仲裁总线上，每个仲裁器将仲裁总线上得到的号与自己的号进行比较。如果仲裁总线上的号大，则它的总线请求不予响应，并撤消它的仲裁号。最后，获胜者的仲裁号保留在仲裁总线上，分布式仲裁是以优先级仲裁策略为基础。



19.总线的一次信息传送过程，大致可分为：请求总线，总线仲裁，寻址，信息传送，状态返回。

20. 设总线带宽用 Dr 表示，总线时钟周期用 T = 1/f 表示，一个总线周期传送的数据量用 D 表示，

根据定义可得：

$$Dr = T/D = D \times 1/f = 8B \times 70 = 560MHz/s$$

21. PCI 总线：是一种不依附于某个具体处理器的局部总线，支持 10 种外设，并能在高时钟频率下保持高性能。总线时钟频率为 33.3MHz/66MHz，最大数据传输速率 133MB/s，采用时钟同步方式，与 CPU 及时钟频率无关，总线宽度 32 位（5V）/64 位（3.3V），能自动识别外设。总线具有与处理器和存储器子系统完全并行操作的能力，具有隐含的中央仲裁系统，采用多路复用方式（地址线和数据线）减少了引脚数，支持 64 位寻址，具有完全的多总线主控能力。

InfiniBand 标准：针对处理器和智能 I/O 设备之间数据流而提出的一种新体系结构，用于在服务器中取代 PCI 总线，采用 InfiniBand 结构将允许服务器提供更高的带宽和可扩展能力，并增强了存储设备扩充的灵活性。InfiniBand 允许服务器，远程存储器，其他网络设备接入到一个由开关和链路组成的中央开关网带，可连接多达 64000 个服务器，存储系统和网络设备。

第七章

1. D
2. C、D、C、A
3. $\frac{1024*1024*256}{8*8}=1MB$
4. 格式化容量=扇区容量*每道扇区数*磁道总数
   =512*9*100*2=921600B
5. 设读写一块信息所需总时间为 t_B，平均找道时间为 t_s，平均等待时间为 t_l，读写一块信息的传输时间为 t_m，则

   $t_B=t_s+t_l+t_m$

   假设磁盘以每秒 r 转速率旋转，每条磁道容量为 N 个字，则数据传输率 $=rN$ 个字/秒。

   又假设每块的字数为 n，因而一旦读写头定位在该块始端，就能在 $t_m \approx (n/rN)$ 秒的时间中传输完毕。

   t_l 是磁盘旋转半周的时间，$t_l=$ （1/2r）秒。由此可得：

   $$t_B=t_s+\frac{1}{2r}+\frac{n}{rN}[秒]$$

6. $\frac{185000B/s}{4000转/60s}=2775B/转=2775B/道$

   2*220*2775 = 1.16MB
7. (1)275*12288*4 = 12.89MB

   (2) 最高位密度 D1 按最小磁道半径 R1 计算（R1 = 115mm）：

   D1 = 12288 字节 /2πR1 = 17 字节 / mm

   最低位密度 D2 按最大磁道半径 R2 计算：

   R2 = R1 + （275 ÷ 5） = 115 + 55 = 170mm

   D2 = 12288 字节 /2πR2 = 11.5 字节 / mm

   (3) $\frac{3000}{60}*12288=600KB/s$

   (4) $\frac{1}{2}*\frac{60}{3000}*1000=10ms$

   (5)

16 15	14 6	5 4	3 0
台号	柱面(磁道)号	盘面(磁头)号	扇区号

   此地址格式表示有 4 台磁盘，每台有 4 个记录面，每个记录面最多可容纳 512 个磁道，每道有 16 个扇区。
8. 存取时间 = 平均查找时间 + 平均等待时间

   $=60+\frac{1}{2}*\frac{60}{2400}*1000=72.5ms$

$$Dr = 96 * \frac{2400}{60} = 480KB/s$$

9. (1) $D = \frac{C}{v} = \frac{128000字节/秒}{2m/s} = 64000字节/m$

(2)传送一个数据块所需时间为

$$t = \frac{1024字节}{128000字节/秒} = \frac{1}{125}秒$$

一个数据块占用长度为

$$l = v * t = 2m/s * \frac{1}{125}s = 0.016m$$

每块间隙 L = 0.014m，数据块总数为

$$\frac{600-4}{l+L} = 19867块$$

故磁带存储器有效存储容量为

19867 块*1K 字节 = 19867K 字节

10. (1)磁盘内径为：9 英寸-5 英寸 = 4 英寸

内层磁道周长为 $2\pi R = 2*3.14*5 = 31.4$英寸

每道信息量 = 1000 位/英寸*31.4 英寸 = $3.14*10^4$ 位

磁盘有 100 道/英寸*5 英寸 = 500 道

盘片组总容量：$20*500*3.14*10^4 = 3.14*10^8$ 位 = 314 兆位

(2)每转即每道含有信息量 $3.14*10^4$ 位，即 $3.925*10^3$B

$$\frac{1MB/s}{3.925*10^3 B/转} = 267转/s = 16020转/分钟$$

11. (1)$[(30*10^{-3}+10*10^{-3}+3000/500*10^{-3})*2+4*10^{-3}]*1000 = 96s$

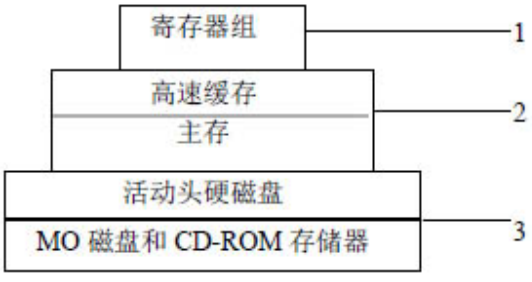
(2) $[(30*10^{-3}+5*10^{-3}+3000/1000*10^{-3})*2+4*10^{-3}]*1000 = 80s$

12.

(1)存储容量从大到小依次为：活动头磁盘存储器， MO 磁盘，CD-ROM 存储器，主存，高速缓存，寄存器组

存储周期从大到小依次为：CD-ROM 存储器，MO 磁盘，活动头磁盘存储器，主存，高速缓存，寄存器组

(2)可构成如下的多级存储体系：



(3)CPU 和高速缓存以及 CPU 和主存之间有直接的数据通路，而 CPU 与外存之间不存在直接的数据通路，CPU 访问硬盘和光盘时都需要先将数据调入主存。

13. 刷新存储器是用来存储一图像信息以不断提供刷新图象的信号。其存储容量由图像分辨率和灰度级决定。

1024*1024*24bit = 3MB

14. (1)1024*768*3 = 2.25MB

(2)1024*768*3B*72/s = 162MB/s

第八章

1.A、B、C

2.B

3.A

4.C

5.组织外围设备和内存进行数据传输；控制外围设备；选择；数组多路；字节多路

6.能响应，因为设备 A 的优先级比设备 B 高。若要设备 B 总能立即得到服务，可将设备 B 从第二级取出来，单独放在第三级上，使第三级的优先级最高，即令 $IM_3 = 0$。

7.依次处理设备 A，设备 D，设备 G 的时间为：

$T_1 = t_1+t_2+t_3+t_4+t_A$

$T_2 = t_1+t_2+t_3+t_4+t_D$

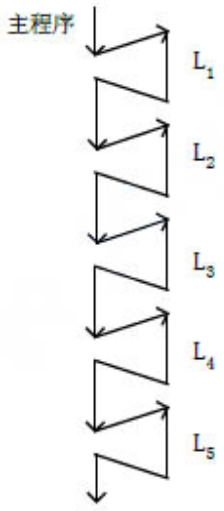
$T_3 = t_1+t_2+t_3+t_4+t_G$

总时间为 $T = T_1+T_2+T_3 = 3*(t_1+t_2+t_3+t_4)+ t_A + t_D + t_G$

8.(1)

中断处理程序	中断处理级屏蔽位				
	L_0级	L_1级	L_2级	L_3级	L_4级
L_0中断处理程序	0	0	0	0	0
L_1中断处理程序	1	0	0	0	0
L_2中断处理程序	1	1	0	0	0
L_3中断处理程序	1	1	1	0	0
L_4中断处理程序	1	1	1	1	0

(2)



9.要将通用寄存器内容保存到主存中去。只需保存中断处理程序用到的那 2 个寄存器内容。

10. 设计思想：二维中断判优结构如主教材图 8.9（b）所示。其中，主优先级独立请求方式的判优电路在主教材图 8.10 的基础上进行改进：将 PSW 中的 5～7 三位经译码器输出 IR4～IR7 共四个请求信号，参与排队器排队。

11.(1)$IM_2IM_1IM_0 = 011$

(2) $IM_2IM_1IM_0 = 001$

(3) 若要设备 B 总能立即得到服务，可将设备 B 从第二级取出来，单独放在第三级上，使第三级的优先级最高，即令 $IM_3 = 0$。

12.D

13.中断、蔽中断、中断、异常、异常、执行软件中断指令

14.B、A、C、D、E

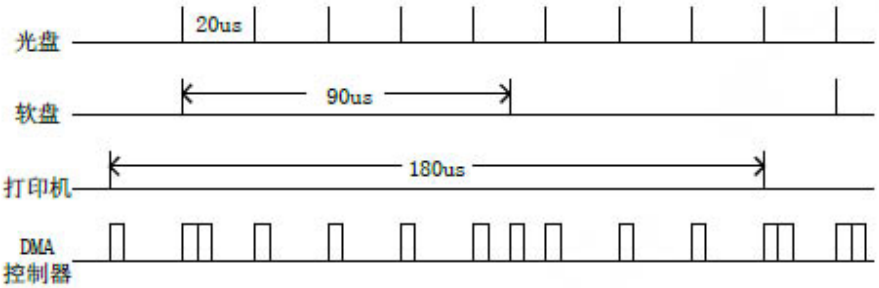
15.B、A、

16.(1)通道方式：可以实现对外设的统一管理和外设与内存之间的数据传送，大大提高了 CPU 的工作效率。

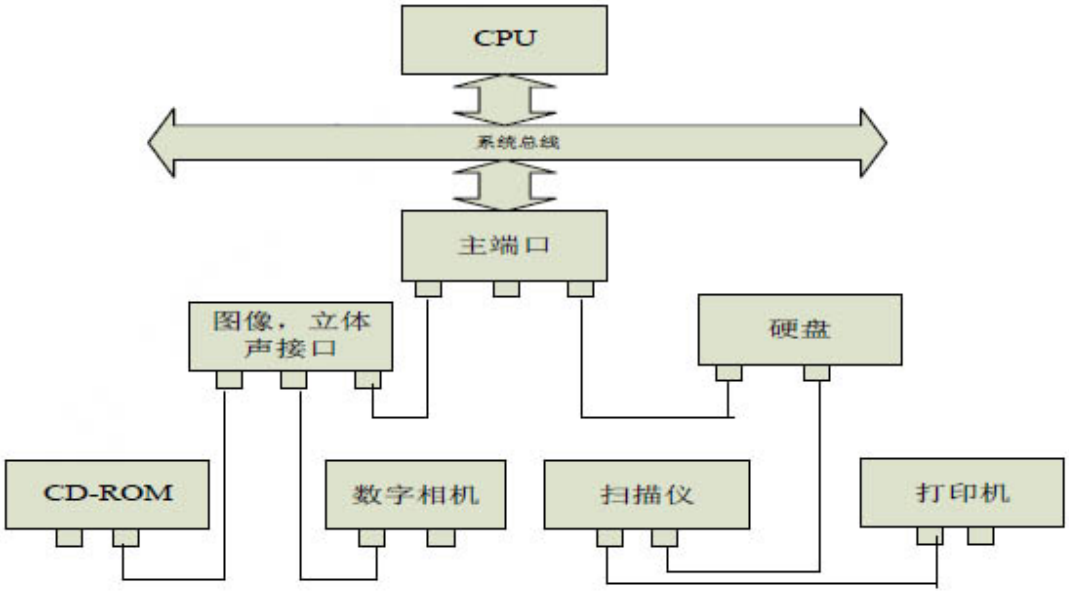
(2)DMA 方式：数据传送速度很高，传送速率仅受到内存访问时间的限制。需要更多硬件，，适用于内存和高速外设之间大批数据交换的场合。

(3)中断方式：一般适用于随机出现的服务，且一旦提出要求应立即进行，节省了 CPU 的时间开销，但硬件结构稍复杂一些。

17.



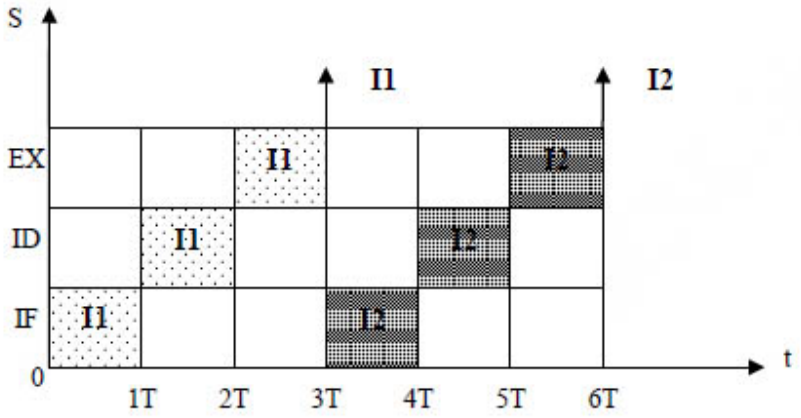
18.主端口是 1394 树形配置结构的根节点。一个主端口最多可连接 63 台设备每个设备称为一个节点，它们构成亲子关系。其中右侧按菊花链式配置，左侧按亲子关系连接。



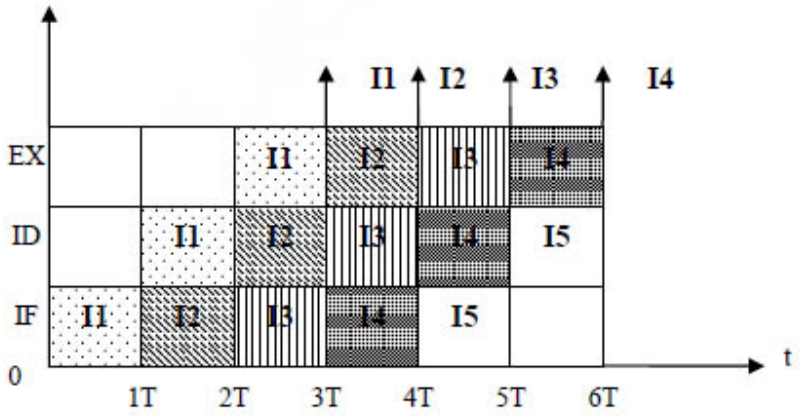
第九章

1. 略
2. 【解】

① 设三个子过程为取指令（IF）、指令译码（ID）、指令执行（EX），则指令顺序执行和流水执行方式时空图如图 9.1（a）和（b）所示。



（a）顺序执行时空图



（b）流水执行时空图

图 9.1　指令执行方式时空图

② 顺序执行方式：n＝1000 条，T＝100ns

总时间：　　$t_1 = 3 \times n \times t = 3 \times 1000 \times 100 = 300000ns$

流水执行方式：　$t_2 = (n+2)T = 1002 \times 100 = 100200ns$

③ 加速比：$Se = t_1 / t_2 = 300000/10020 \approx 2.999$倍

3. 【解】

④ 设 k=4 个，流水段为取指（s1）、译码（s2）、执行（s3）、存结果（s4）。

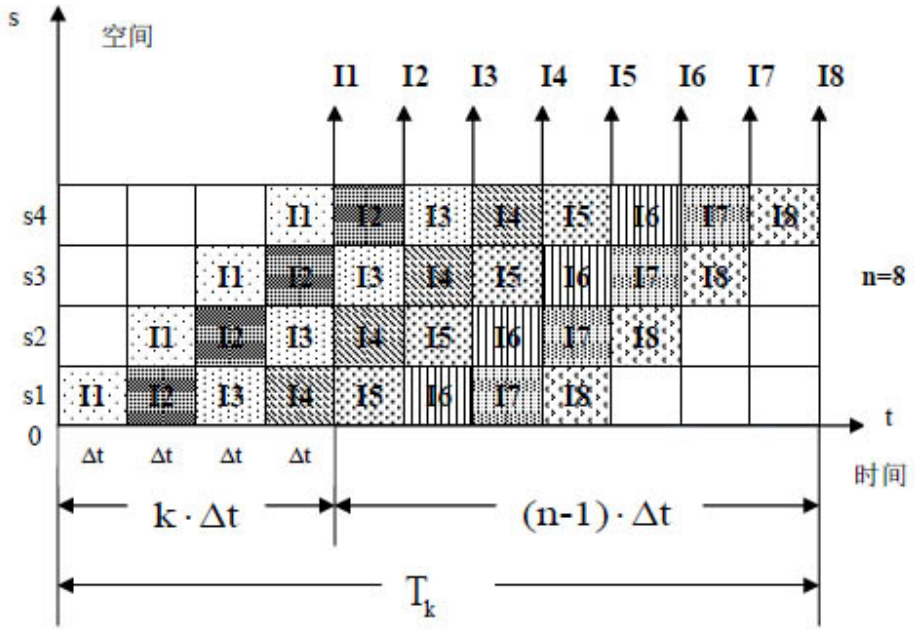


图 9.2　n=8 条指令流水时空图

⑤ 从图 9.2 看出，用 k=4 个时钟周期（Δt）完成第 1 条指令，其余 n-1 个时钟周期完成 n-1 条指令。因此流水线完成 n 条指令所需的总时间为

$$T_k = (k+n-1)\Delta t$$

根据定义，吞吐率 P 为

$$P = \frac{n}{k} = \frac{n}{(k+n-1)\Delta t}$$

⑥ 顺序执行 n 条指令所用的总时间 T0 为

$$T_0 = (k \cdot \Delta t) \cdot n$$

根据定义，加速比的公式为

$$S_e = \frac{T_0}{T_k} = \frac{nk\Delta t}{(k+n-1)\Delta t} = \frac{nk}{k+n-1}$$

4. D

5. C

6. C

7. 【解】

设总指令数为 m，并行指令数为 m（P），顺序指令数为 m（S），则总执行时间 T 为：

$$T=\frac{m(P)}{nx}+\frac{m(S)}{x}=\frac{mF}{nx}+\frac{m(1-F)}{x}$$

有效 MIPS 表达式为：

$$MIPS=\frac{m}{T}=\frac{m}{\frac{mF}{nx}+\frac{m(1-F)}{x}}=\frac{m}{\frac{mF+nm-nmF}{nx}}=\frac{nx}{n(1-F)+F}$$

8.【解】

在上式中代入已知条件：

$$64=\frac{32\times 8}{32(1-F)+F}$$

求得 F=0.90=90%。

9.【解】

设加速比为 k，可加速部分比例为 F_e，理论加速比为 S_e，根据 Amdahl 定律：

$$k=\frac{1}{(1-F_e)+F_e/S_e}$$

为了简单化，假设程序只在两种模式下运作：（1）使用所有处理机的运行模式；（2）只用一个处理机的串行模式。假设并行模式下的理论加速比 S_e 即为多处理机的台数，加速部分的比例 F_e 即并行部分所占的比例，代入上式有：

$$80=\frac{1}{(1-F_e)+F_e/100}$$

求得并行比例 $F_e=0.9975=99.75\%$，串行比例 $1-F_e=0.25\%$