

第 1 章 绪论

1.6 早期的冯·诺依曼计算机硬件系统主要由哪几部分组成?各部分的功能是什么?

答:

冯诺依曼型电脑的五大组成部分和各部分的功能如下:

1、运算器: 计算机中执行各种算术和逻辑运算操作的部件。运算器的基本操作包括加、减、乘、除四则运算, 与、或、非、异或等逻辑操作, 以及移位、比较和传送等操作, 亦称算术逻辑部件 (ALU);

2、控制器: 由程序计数器、指令寄存器、指令译码器、时序产生器和操作控制器组成, 它是发布命令的“决策机构”, 即完成协调和指挥整个计算机系统的操作。运算器和控制器统称中央处理器, 也叫做 CPU。中央处理器是电脑的心脏;

3、存储器: 存储器分为内存和外存。内存是电脑的记忆部件, 用于存放电脑运行中的原始数据、中间结果以及指示电脑工作的程序。外存就像笔记本一样, 用来存放一些需要长期保存的程序或数据, 断电后也不会丢失, 容量比较大, 但存取速度慢。当电脑要执行外存里的程序, 处理外存中的数据时, 需要先把外存里的数据读入内存, 然后中央处理器才能进行处理。外存储器包括硬盘、光盘和优盘;

4、输入设备: 输入设备是向计算机输入数据和信息的设备。是计算机与用户或其他设备通信的桥梁。输入设备是用户和计算机系统之间进行信息交换的主要装置之一。键盘, 鼠标, 摄像头, 扫描仪, 光笔等都属于输入设备。

5、输出设备: 是计算机硬件系统的终端设备, 用于接收计算机数据的输出显示、打印、声音、控制外围设备操作等。也是把各种计算结果数据或信息以数字、字符、图像、声音等形式表现出来。常见的输出设备有显示器、打印机等。

1.7 叙述冯·诺依曼计算机的基本工作过程。

冯·诺依曼描述的计算机基本工作原理的主要思想是程序存储。

存储程序原理又称“冯·诺依曼原理”(1946 年提出)。将程序像数据一样存储到计算机内部存储器中的一种设计原理。程序存入存储器后, 计算机便可自动地从一条指令转到执行另一条指令。现代电子计算机均按此原理设计。

1.11 说明 Flynn 对计算机分类的依据, 并描述不同类型计算机的特点。

按照计算机在执行程序的过程中信息流的特征进行分类。

四类计算机的特点见 P21。

第 2 章 计算机中的数据表示

2.1 实现下列各数的转换

$$(1) (97.8125)_{10} = (\quad)_2 = (\quad)_8 = (\quad)_{16}$$

$$(2) (110101.011)_2 = (\quad)_{10} = (\quad)_8 = (\quad)_{16} = (\quad)_{8421BCD}$$

$$(3) (001101101001.0101)_{8421BCD} = (\quad)_{10} = (\quad)_2 = (\quad)_{16}$$

$$(4) (2A7C.5E)_{16} = (\quad)_2 = (\quad)_{10}$$

答

$$(1) (97.8125)_{10} = (1100001.1101)_2 = (141.64)_8 = (61.D)_{16}$$

$$(2) (110101.011)_2 = (53.375)_{10} = (65.3)_8 = (35.6)_{16} = (01010011.001101110101)_{8421BCD}$$

$$(3) (001101101001.0101)_{8421BCD} = (369.5)_{10} = (101110001.1)_2 = (171.8)_{16}$$

$$(4) (2A7C.5E)_{16} = (0010101001111100.01011110)_2 = (10876.3671875)_{10}$$

2.2 已知 $[X]_{\text{原}}$ ，求 $[X]_{\text{补}}$ 和 $[X]_{\text{反}}$

$$(1) [X]_{\text{原}} = 0.1010110, [X]_{\text{补}} = \underline{\hspace{2cm}}, [X]_{\text{反}} = \underline{\hspace{2cm}}$$

$$(2) [X]_{\text{原}} = 1.0010110, [X]_{\text{补}} = \underline{\hspace{2cm}}, [X]_{\text{反}} = \underline{\hspace{2cm}}$$

$$(3) [X]_{\text{原}} = 01010110, [X]_{\text{补}} = \underline{\hspace{2cm}}, [X]_{\text{反}} = \underline{\hspace{2cm}}$$

$$(4) [X]_{\text{原}} = 11010010, [X]_{\text{补}} = \underline{\hspace{2cm}}, [X]_{\text{反}} = \underline{\hspace{2cm}}$$

答案: (1) 0.1010110; 0.1010110

(2) 1.1101010; 1.1101001

(3) 01010110; 01010110

(4) 10101110; 10101101

2.3 已知 $[X]_{\text{补}}$ ，求 X

$$(1) [X]_{\text{补}} = 1.1101101, X = \underline{\hspace{2cm}}$$

$$(2) [X]_{\text{补}} = 0.1010110, X = \underline{\hspace{2cm}}$$

$$(3) [X]_{\text{补}} = 10000000, X = \underline{\hspace{2cm}}$$

$$(4) [X]_{\text{补}} = 11010010, X = \underline{\hspace{2cm}}$$

答案: (1) -0.0010011 (2) 0.1010110 (3) -10000000 (-128) (4) -101110

2.4 假设机器字长为 8 位，求下列补码所对应的 X 的十进制真值。

$$(1) [2X]_{\text{补}} = 90H \quad (2) [\frac{1}{2}X]_{\text{补}} = C2H \quad (3) [-X]_{\text{补}} = FEH$$

答案:

$$[2X]_{\text{补}} = 90H = (10010000)_2$$

$$2X = [10010000]_{\text{补}} = 11110000 = -112$$

$$\therefore X = -56$$

或 $2X = (10010000)_2$ 右移一位得到 X，即 $(10111000)_2 = -56$

$$[\frac{1}{2}X]_{\text{补}} = C2H = (11000010)_2$$

$$\frac{1}{2}X=[11000010]_{\text{补}}=10111110$$

$$\frac{1}{2}X \text{ 左移一位得到 } X, \text{ 即 } X=(11111100)_2=-124$$

$$[-X]_{\text{补}}=\text{FEH}=(11111110)_2$$

$$-X=[11111110]_{\text{补}}=(10000010)_2=-2$$

$$\therefore X=2$$

2.6 假设机器字长位 8 位，已知 $[X]_{\text{补}}=3\text{AH}$ ， $[Y]_{\text{补}}=\text{C5H}$ ，求 $[2X]_{\text{补}}$ ， $[2Y]_{\text{补}}$ ， $[X/2]_{\text{补}}$ ， $[Y/4]_{\text{补}}$ ， $[-X]_{\text{补}}$ ， $[-Y]_{\text{补}}$ ， $[X]_{\text{原}}$ ， $[Y]_{\text{原}}$ ， $[X]_{\text{反}}$ ， $[Y]_{\text{反}}$ ， $[X]_{\text{移}}$ ， $[Y]_{\text{移}}$ 。注意：结果用十六进制表示

答： $[X]_{\text{补}}=3\text{AH}=00111010$ ， $\therefore X=00111010=58$

$$[Y]_{\text{补}}=\text{C5H}=11000101, \therefore Y=10111011=-59$$

$$[2X]_{\text{补}}=01110100=74\text{H}; \quad [2Y]_{\text{补}}=10001010=8\text{AH}$$

$$[X/2]_{\text{补}}=00011101=1\text{DH}; \quad [Y/4]_{\text{补}}=[10001110]_{\text{补}}=11110001=\text{F1H}$$

$$[-X]_{\text{补}}=[X]_{\text{补}}_{\text{求补}}=11000110=\text{C6H}; \quad [-Y]_{\text{补}}=[Y]_{\text{补}}_{\text{求补}}=00111011=3\text{BH}$$

$$[X]_{\text{原}}=00111010=3\text{AH}; \quad [Y]_{\text{原}}=10111011=\text{BBH}$$

$$[X]_{\text{反}}=00111010=3\text{AH}; \quad [Y]_{\text{反}}=11000100=\text{C4H}$$

$$[X]_{\text{移}}=10111010=\text{BAH}; \quad [Y]_{\text{移}}=01000101=45\text{H}$$

2.7 若机器字长为 32 位，则整数补码和小数补码可表示数的个数是多少？可表示的真值 X 的范围分别是多少？

答：整数补码和小数补码可表示数的个数是 2^{32} ；

整数补码的范围： $-2^{31} \leq X \leq 2^{31}-1$

小数补码的范围： $-1 \leq X \leq 1-2^{-31}$

2.8 分别写出下列各十进制数的原码，反码和补码，用 8 位二进制表示（最高位为符号位）。如果是小数，小数点在最高位（符号位）之后；如果是整数，小数点在最低位之后，并写出其对应的移码。

(1) 用整数表示的-1； (2) 用小数表示的-1； (3) 用整数表示的+0；

(4) 用小数表示的-0； (5) 45/64； (6) -1/128；

(7) +128； (8) -128； (9) +127； (10) -127； (11) 89； (12) -32

		原码	反码	补码	移码
--	--	----	----	----	----

(1)	用整数表示的-1	1000 0001	1111 1110	1111 1111	0111 1111
(2)	用小数表示的-1	无	无	1.000 0000	无
(3)	用整数表示的+0	0000 0000	0000 0000	0000 0000	1000 0000
(4)	用小数表示的-0	1.000 0000	1.111 1111	0.000 0000	无
(5)	45/64	0.1011010	0.1011010	0.1011010	无
(6)	-1/128	1.0000001	1.1111110	1.1111111	无
(7)	+128	无	无	无	无
(8)	-128	无	10000000	无	00000000
(9)	+127	01111111	01111111	01111111	11111111
(10)	-127	1111 1111	1000 0000	1000 0001	0000 0001
(11)	89	01011001	01011001	01011001	10011001
(12)	-32	10100000	11011111	11100000	01100000

2.9 若约定小数点在 8 位二进制数的最右端（定点整数）。试分别写出下列各种情况下 W、X、Y、Z 的真值。

(1) $[W]_{\text{补}} = [X]_{\text{原}} = [Y]_{\text{反}} = [Z]_{\text{移}} = 00H$

(2) $[W]_{\text{补}} = [X]_{\text{原}} = [Y]_{\text{反}} = [Z]_{\text{移}} = 80H$

(3) $[W]_{\text{补}} = [X]_{\text{原}} = [Y]_{\text{反}} = [Z]_{\text{移}} = FFH$

答：

	$[W]_{\text{补}}$	$[X]_{\text{原}}$	$[Y]_{\text{反}}$	$[Z]_{\text{移}}$
00H=00000000	+0	+0	+0	-128
80H=10000000	-128	-0	-127	0
FFH=11111111	-1	-127	-0	127

2.10 多项选择题

- (1) 在数值数据的编码中，0 有唯一表示的编码是 (C,D)
- (2) 符号位用 0 表示，用 1 表示负的编码有 (A,B,C)
- (3) 满足若真值大，则码值越小现象的编码是 (D)
- (4) 存在负数的真值越大，则码值越小现象的编码是 (A)

(5) 负数的码值大于正数的码值的编码有 (A, B, C)

可供选择的答案: A. 原码 B. 反码 C. 补码 D. 移码

2.11 假设机器字长为 8 位

(1) 码值为 80H, 若表示真值 0, 则为 () 码; 若表示真值 -128, 则为 () 码; 若表示真值 -127, 则为 () 码; 若表示真值 -0, 则为 () 码。

(2) 码值为 FFH, 若表示真值 127, 则为 () 码; 若表示真值 -127, 则为 () 码; 若表示 -1; 则为 () 码; 若表示真值 -0, 则为 () 码。

答: (1) 移 补 反 原 (2) 移 原 补 反

2.12 设 X 为小数, X 在什么范围内, 有 $[X]_{\text{补}} > [X]_{\text{原}}$? 在什么范围内, 有 $[X]_{\text{补}} = [X]_{\text{原}}$? 当 $X < 0$ 时, 试求出满足 $[X]_{\text{补}} = [X]_{\text{原}}$ 的真值 X 。

答:

当 $X \geq 0$ 时, $[X]_{\text{补}} = [X]_{\text{原}}$

当 $X < 0$ 时, $[X]_{\text{补}} > [X]_{\text{原}}$ 则有 $2+X > 1-X$

$\therefore X > -1/2$

当 $X < 0$ 时, 当 $X = -1/2$ 时, $[X]_{\text{补}} = [X]_{\text{原}} = 1.1000000$

2.13 对于一个用 8 位二进制表示的整数补码, 如何判读其正负? 如何判断其有无十进制的百位? 如何判断其奇偶性? 如何判断其能否被 8 整除?

答:

补码的首位为 0, 则为正; 首位为 1, 则为负。

$100 = 64 + 42 + 4$ 对应的二进制表示: 01100100, 二进制数大于 01100100 时, 就会有十进制的百位。

通过判断最低位的值判断奇偶性: 0 为偶数, 1 为奇数。

8 对应的二进制为 1000, 如果一个二进制编码的最后三位为 000, 则就可以被 8 整除。

2.14 若机器字长为 n 位二进制, 可用来表示多少个不同的数? 就下列三种情况, 分别写出所能表示的最大值和最小值。

- (1) 无符号数
- (2) 用原码表示的整数
- (3) 用补码表示的小数

答：n 位二进制可表示 2^n 个不同的数。

n 位二进制	最大值	最小值
(1) 无符号数	2^n-1	0
(2) 用原码表示的整数	$2^{n-1}-1$	$-(2^{n-1}-1)$
(3) 用补码表示的小数	$1-2^{-(n-1)}$	-1

2.15 试写出下列各种情况下用 16 位二进制所能表示的数的范围（十进制表示）以及对应的二进制代码。

- (1) 无符号的整数
- (2) 补码表示的有符号整数
- (3) 补码表示的有符号小数
- (4) 移码表示的有符号整数
- (5) 原码表示的有符号小数

答：

16 位二进制	表示数的范围	最小值和最大值的二进制代码
(1) 无符号的整数	$0 \leq X \leq 2^{16}-1$	$0000...00 \leq X \leq 1111...11$
(2) 补码表示的有符号整数	$-2^{15} \leq X \leq 2^{15}-1$	$1000...00 \leq X \leq 0111...11$
(3) 补码表示的有符号小数	$-1 \leq X \leq 1-2^{-15}$	$1000...00 \leq X \leq 0111...11$
(4) 移码表示的有符号整数	$0 \leq X \leq 2^{16}-1$	$1000...00 \leq X \leq 0111...11$
(5) 原码表示的有符号小数	$-(1-2^{-15}) \leq X \leq 1-2^{-15}$	$1111...11 \leq X \leq 0111...11$

2.16 下列代码可看作 ASCII 码、整数补码、BCD 码时分别代表什么？

- (1) 78H
- (2) 39H

答

	ASCII 码	整数补码	BCD 码
78H=01111000	x	120	78
39H=00111001	9	57	39

2.17 设二进制浮点数字长 16 位，其中阶码 6 位（含 1 位阶符），用移码表示，尾数 10 位（含 1 位数符），用补码表示。浮点数编码格式如下图所示：

数符	阶符	尾数
----	----	----

（1） 确定所能表示的规格化浮点数的范围，填入下表中。并与 16 位定点补码整数和定点补码小数的表示范围进行比较。

数值	阶码(十六进制)	尾数（十六进制）	真值（十进制）
最大正数	11 1111=3FH	01 1111 1111=1FFH	$(1-2^{-9}) \times 2^{31}=2.1433e+09$
最小正数	00 0000=00H	01 0000 0000=100H	$2^{-1} \times 2^{-32}=1.1642e-10$
最大负数	00 0000=00H	10 1111 1111=2FFH	$-(2^{-1}+2^{-9}) \times 2^{-32}= -1.1687e-10$
最小负数	11 1111=3FH	10 0000 0000=200H	$-1 \times 2^{31}=-2.1475e+09$

（2） 判断下列十进制数能否表示成此格式的规格化浮点数（允许有误差）。若可以，请写出对应的码值。

- ① 3.14 ② -1917 ③ 105/512 ④ -10^{-6} ⑤ 10^{10}

答：

① 3.14 的二进制表示为：+11.00100100 00111001010110

尾数精度不够，不能准确表示 3.14

② -1917 的二进制表示为：-11101111101，超出尾数能表示的精度，无法准确表示

③ $105/512 = 0.001101001 = 0.1101001000 \times 2^{-100010}$

尾数的补码：0.1101001000，阶码的补码：111110 移码：011110

0	011110	1101001000
---	--------	------------

④ -10^{-6}

$$2^{19}=524288 \quad 2^{20}=1048576$$

因此，利用尾数的精度无法表示该数。

⑤ 10^{10} 超出可表示数的最大范围

2.18 以 IEEE754 单精度浮点数格式（32 位）表示下列二进制数

(1) +5.3125 (2) -365.59375 (3) +21 (4) -35/8 (5) 324 (6) 56789.25

答：

$$(1) +5.3125=+101.0101=+1.010101 \times 2^2=(-1)^0 \times 2^{e-127} \times 1.010101$$

$$\therefore s=0; \quad e=129=10000001$$

IEEE 754 表示: 0 10000001 0101 0100 0000 0000 0000 000

$$(2) -365.59375=-101101101.10011=-1.0110110110011 \times 2^8=(-1)^1 \times 2^{e-127} \times 1.0110110110011$$

$$\therefore s=1, \quad e=135=10000111$$

IEEE 754 表示: 1 10000111 0110 1101 1001 1000 0000 000

$$(3) +21=+10101=+1.0101 \times 2^4=(-1)^0 \times 2^{e-127} \times 1.0101$$

$$\therefore s=0, \quad e=131=10000011$$

IEEE 754 表示: 0 1000 0011 0101 0000 0000 0000 000

$$(4) -35/8 \quad -35=-100011, \quad -35/8=-100.011=-1.00011 \times 2^2=(-1)^1 \times 2^{e-127} \times 1.00011$$

$$\therefore s=1, \quad e=129=10000001$$

IEEE 754 表示: 1 1000 0001 0001 1000 0000 0000 0000 000

$$(5) 324=101000100=1.010001 \times 2^8=(-1)^0 \times 2^{e-127} \times 1.010001$$

$$\therefore s=0, \quad e=135=10000111$$

IEEE 754 表示: 0 10000111 0100 0100 0000 0000 0000 000

$$(6) \quad 56789.25=1101110111010101.01=1.10111011101010101 \times 2^{15}=(-1)^0 \times 2^{e-127} \times 1.10111011101010101$$

$$\therefore s=0, \quad e=142=10001110$$

IEEE 754 表示: 0 10001110 1011 1011 1010 1010 1000 000

2.19 假设计算机使用的是 24 位字，试在如下情况下，利用 24 位来表示 365。

(1) 如果计算机使用原码表示定点整数，如何表示十进制数 365？

(2) 如果计算机使用 8 位 ASCII 码和奇校验码，如何表示字符串“365”？

答: (1) $(365)_{10} = (101101101)_2$

\therefore 365 的原码表示整数为: 00000000 00000001 01101101

(2) 字符串“365”的 ASCII 编码: 3-0110011; 6-0110110; 5-0110101, 采用奇校验后, 8 位 ASCII 码分别为: 3-10110011; 6-10110110; 5-10110101;

\therefore 字符串“365”的表示为 10110011 10110110 10110101

2.20 已知字母“A”的 ASCII 编码为 1000001, 字母“a”的 ASCII 编码为 1100001, 数字“0”的 ASCII 编码为 0110000。求字符“D”, “K”, “f”, “h”, “5”, “7”的 7 位 ASCII 编码, 并在最高位加入偶校验位, 形成带奇偶校验位的 8 位 ASCII 码。

答: 0001-0010-0011-0100-0101-0110-0111-1000-1001-1010-1011-...

字母“A”的 ASCII 编码为 1000001, 因此字符“D”的带偶校验的 ASCII 编码为 01000100, “K”的带偶校验的 ASCII 编码为 01001011。

字母“a”的 ASCII 编码为 1100001, 因此字符“f”的带偶校验的 ASCII 编码为 01100110, “h”的带偶校验的 ASCII 编码为 11101000。

数字“0”的 ASCII 编码为 0110000, 因此, 字符“5”的 ASCII 编码为带偶校验的 00110101, “7”的 ASCII 编码为带偶校验的 10110111。

2.21 约定生成多项式为 $G(x) = x^3 + x + 1$, 试计算下述信息字的 CRC 编码, 并在接收端进行校验。

(1) 1010110 (2) 01011001

答; $G(x)$ 多项式对应的除数为 1011,

(1) 1010110000 除以 1011, 商为 1001100, 余数为 100, 因此发送端的 CRC 编码为 101100100

$$\begin{array}{r} 1001100 \\ 1011 \overline{) 1010110000} \\ \underline{1011} \\ 1110 \end{array}$$

1011—

1010

1011—

100

接收端校验过程如下，得到的余数为 0，因此正确无误。

$$\begin{array}{r}
 1001100 \\
 \hline
 1011 \overline{) 1010110100} \\
 \underline{1011} \\
 1110 \\
 \underline{1011} \\
 1011 \\
 \underline{1011} \\
 000
 \end{array}$$

(2) 01011001000 除以 1011，商为 01000001，余数为 011，因此发送端的 CRC 编码为 01000001011

$$\begin{array}{r}
 1000001 \\
 \hline
 1011 \overline{) 01011001000} \\
 \underline{1011} \\
 001000 \\
 \underline{1011} \\
 11
 \end{array}$$

接收端校验过程如下，得到的余数为 0，因此正确无误。

$$\begin{array}{r}
 1000001 \\
 \hline
 1011 \overline{) 01011001011} \\
 \underline{1011} \\
 001011 \\
 \underline{1011} \\
 0000
 \end{array}$$

2.22 假设主存中存储的数据位 16 位，欲利用海明码纠正一位错，海明码码长最少为几位？为什么？

答：假设需要增加的校验位为 r 位，则 r 需要满足不等式 $2^r \geq 16+r+1$ ，不等式中的 $16+r$ 代表了码字任何一位都可能出错的情况，1 代表了所有位均无错误的情况，增加的校验位的状态需要能够表示区分出所有的出错和不出错的情况。满足不等式的最小 r 为 5，因此海明码码

长最少为 21 位。

2.23 假设正在使用的一种纠错码可以纠正长度为 8 的存储字的全部 1 位错误。计算结果表明，需要 4 位校验位，编码字的全部长度为 12。编码字的产生方式采用本章介绍的海明编码算法。现在接收器收到如下的代码字：010111010110，请问：收到的这个字是否为正确的编码字。

参考答案：k=8，r=4 的海明码分组如下表：

序号 分组	H ₁₂	H ₁₁	H ₁₀	H ₉	H ₈	H ₇	H ₆	H ₅	H ₄	H ₃	H ₂	H ₁
	D ₈	D ₇	D ₆	D ₅	P ₄	D ₄	D ₃	D ₂	P ₃	D ₁	P ₂	P ₁
P ₄	√	√	√	√	√							
P ₃	√					√	√	√	√			
P ₂		√	√			√	√			√	√	
P ₁		√		√		√		√		√		√

H₁₂~H₁=0101 1101 0110，

则 P₄P₃P₂P₁=1001,表明 H₉ 出错，将其取反得正确的海明码为 0100 1101 0110。

第 3 章 运算方法与运算器

3.1 简要说明运算器的功能

答：计算机中的运算器用于数值运算及加工处理数据，它由 CPU 中的算术逻辑单元、通用寄存器等部件构成，其结构取决于指令系统、数据表示方法、运算方法及所选用的硬件。

3.2 说明采用单符号位检测溢出的方法。

- ① 利用参加运算的两个数据和结果的符号进行判断：两个符号位相同的数相加，若结果的符号位和加数的符号位相反，则表明有溢出发生；两个符号位相反的数相减，若结果的符号位与被减数的符号位相反，则表明有溢出发生。其他情况无溢出。
- ② 利用编码的进位情况来判断溢出： $OF=C_n \oplus C_{n-1}$ ，其中， C_n 为符号位的进位， C_{n-1} 为最高数值位的进位。 $OF=1$ 表示溢出。

3.3 说明采用双符号位检测溢出的方法。

答：

双符号位检测溢出是采用两位二进制表示符号，即正数的符号位为 00，负数的符号位为 11。运算时，符号位也参与运算，结果的两个符号位如果不同，则表示有溢出产生。若为 01，表示运算结果大于允许范围的最大正数，为正溢。若为 10，表示运算结果大于允许范围的最大负数，为负溢。

3.4 根据图 3.8 说明 BCD 加法器的工作原理。

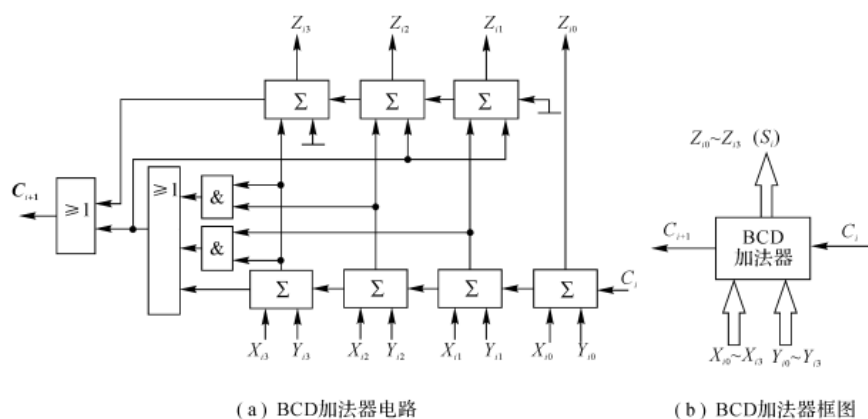


图 3.8 一位 8421BCD 加法器

见课上讲解内容。

3.5 说明浮点数运算中溢出的处理方法。

溢出是超出了机器数所能表示的数据范围，浮点数的范围是有阶码决定的，因此当运算结果的阶码大于最大阶码时，就会发生溢出。当阶码小于最小负阶码时，计算机将把结果按 0 处理。

3.6 某微型机字长 16 位，若采用定点补码整数表示数值，最高 1 位为符号位，其余 15 位为数值部分，则所能表示的最小正数和最大负数分别为多少？

答：最小正数：0001H=1；最大负数为 FFFFH=-1

3.7 在进行定点原码乘法运算时，乘积的符号位是由被乘数的符号位和乘数的符号位进行何种运算获得的？

答：异或

3.8 用字长为 16 的定点补码纯小数所能表示的最大正数及最小负数分别为多少？

答：

最大正数：1-2⁻¹⁵

最小负数：-1

3.9 当两个符号相同的数相加或两个符号相异的数相减时，如何利用所得结果的符号位 SF 和进位标志 CF 来确定所得结果是否产生了溢出？

答：溢出标志位 OF=SF⊕CF，如果 OF=1，表示溢出。

3.10 若浮点数的阶码用移码表示，尾数用补码表示。两规格化浮点数相乘，最后对结果进行规格化，右规时尾数右移位数最多为几位？左规时尾数左移位数最多为几位？为什么？

答：假定尾数为 n 位补码，其中有一位符号，其规格化正数的范围为+1/2~+(1-2⁻⁽ⁿ⁻¹⁾)，规格化负数的范围为 -(1/2+2⁻⁽ⁿ⁻¹⁾)~-1。当两个乘数的尾数均为规格化数时，两者之积的绝对值肯定不小于 1/4。

因此，乘积的尾数如果需要左规，也只需要左移一位，便可使积之尾数变为规格化数。

同样，两者之积有可能为+1，即两个 -1 相乘。而 +1 不是规格化数，因此，乘积的尾数需要右规 1 位，便可变成规格化数。

3.11 分析图 3.36 并回答问题：该框图所完成的功能是什么？在运算开始时，D、A、B 寄存器中分别存放的是什么？

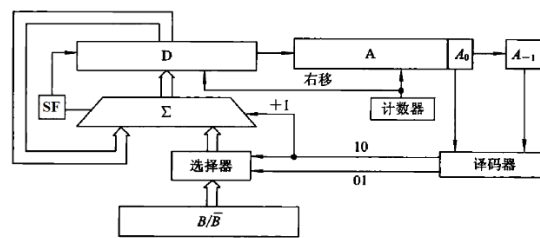


图3.36 习题3.11附图

答：

功能是布斯法补码一位乘法器。

D 中存放部分积，初始值置为 0；

A 中存放乘数，B 中存放被乘数。

3.12 若机器字长为 8 位，定点小数表示。已知 $X=-0.10110$ ， $Y=0.10010$

(1) 求 $[X]_{\text{补}}$ 、 $[Y]_{\text{补}}$ 和 $[-Y]_{\text{补}}$

(2) 用变形码计算 $[X+Y]_{补}$ 和 $[X-Y]_{补}$ ，并判断结果是否溢出。

答：

(1) $X = -0.10110$, $Y = 0.10010$

$[X]_{\text{原}}=1.10110, [Y]_{\text{原}}=0.10010$

$[X]_{\text{补}}=1.01010$, $[Y]_{\text{补}}=0.10010$, $[-Y]_{\text{补}}=1.01110$

(2) $[X+Y]_{\text{补}}=1.11100$

11.01010

$$\underline{+ 00.10010}$$

11.11100

双符号位=11，无溢出。

[X-Y]_補

11.01010

$$\underline{+ 11.01110}$$

10.11000

双符号位=10，负溢。

3.13 图 3.37 是一位算术逻辑单元（ALU）的结构框图。根据此图判断控制信号 S_2 、 S_1 、 S_0 为何种状态时 $F_i=0$ ， S_2 、 S_1 、 S_0 为何种状态时 $F_i=X_i+Y_i$ （加法）。

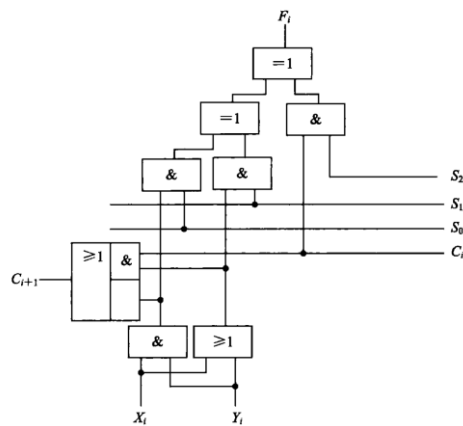


图 3.37 习题 3.13 附图

答： $S_2S_1S_0=000$ 时, $F_i=0$;

$S_2S_1S_0=010$ 时, $F_i=X_i+Y_i$ （加法）。

3.14 已知 $[X]_{\text{补}}=11001100$ 和 $[Y]_{\text{补}}=10101001$ 分别为纯小数，且最高位为符号位，求两者之和并说明是否有溢出。

答： $11001100+10101001=01110101$, 两个负数相加结果为正数，结果显然是错误的，结果溢出。

3.15 已知 $[2X]_{\text{补}}=1.1001001$ ， $[Y/2]_{\text{原}}=1.0101100$ ，试利用变形码计算 $[X]_{\text{补}}+[Y]_{\text{补}}$ ，并判断结果是否有溢出。

答： $[2X]_{\text{补}}=1.1001001$ ，因此，将其右移一位得到 $[X]_{\text{补}}=11.1100100$

$[Y/2]_{\text{原}}=1.0101100$ ，因此，将其左移一位得到 $[Y]_{\text{原}}=1.1011000$ ， $\therefore [Y]_{\text{补}}=11.0101000$

$[X]_{\text{补}}+[Y]_{\text{补}}=1.0001100$

11.1100100

$+ 11.0101000$

11.0001100

双符号位=11，无溢出

3.16 已知 $[X/2]_{\text{补}}=1.1001001$ ， $[2Y]_{\text{原}}=1.0101100$ ，试利用变形码计算 $[X]_{\text{补}}+[Y]_{\text{补}}$ ，并判断结果是否有溢出。

答: $[X/2]_{\text{补}}=1.1001001$, 左移一位得到 $[X]_{\text{补}}=1.0010010$

$[2Y]_{\text{原}}=1.0101100$ 右移一位得到 $[Y]_{\text{原}}=1.1010110$, 所以 $[Y]_{\text{补}}=1.0101010$

$[X]_{\text{补}}+[Y]_{\text{补}}=$

11.0010010

+11.0101010

10.0111100

双符号位=10, 负溢

3.17 依据下列二进制数用补码计算 $[X]_{\text{补}}+[Y]_{\text{补}}$, 并判断结果是否有溢出。

(1) $X=0.01001$, $Y=-0.10111$ 。

(2) $X=0.10010$, $Y=0.11000$ 。

(3) $X=-0.01101$, $Y=0.00101$ 。

(4) $X=-0.11011$, $Y=-0.10010$ 。

答:

(1) $X=0.01001$, $Y=-0.10111$ 。

$[X]_{\text{补}}=00.01001$, $[Y]_{\text{补}}=11.01001$

00.01001

+ 11.01001

11.10010

双符号位=11, 无溢出, 因此 $[X]_{\text{补}}+[Y]_{\text{补}}=1.10010$

(2) $X=0.10010$, $Y=0.11000$ 。

$[X]_{\text{补}}=00.10010$, $[Y]_{\text{补}}=00.11000$

00.10010

+ 00.11000

01.01010

双符号位=01, 正溢, 溢出。

(3) $X=-0.01101$, $Y=0.00101$ 。

$[X]_{\text{补}}=11.10011$, $[Y]_{\text{补}}=00.00101$

11.10011

+ 00.00101

11.11000

双符号位=11，未溢出。[X]_补+ [Y]_补=1.11000

(4) X=-0.11011, Y=-0.10010。

[X]_补=11.10011, [Y]_补=00.00101

11.10011

+ 00.00101

11.11000

双符号位=11，未溢出。[X]_补+ [Y]_补=1.11000

3.18 将下列十进制数变换为 6 位移码表示，其中最高 1 位为符号位，其余 5 位为数值位。

计算[X + Y]_移，并判断结果是否有溢出。

(1) X=16, Y=15

(2) X=-22, Y=-15

答：

(1) [X]_移=110000; [Y]_移=101111;

两者之和为：[X]_移+ [Y]_移=110000+101111=1011111

进位判决法 OF=C_n⊕C_{n-1}=1⊕ 0 = 1, 溢出

(2) [X]_移=001010; [Y]_移=010001;

两者之和为：[X]_移+ [Y]_移=001010+010001=011011

进位判决法 OF=C_n⊕C_{n-1}=0⊕ 0 = 0, 无溢出

将结果符号取反：[X]_移+ [Y]_移=111011

3.19 将下列十进制数用 8421 码表示，并进行运算及校正。

(1) 88+99

(2) 27+15

答：

(1) 88=1000 1000; 99=1001 1001

1000 1000

+1001 1001

10010 0001

+ 0110 0110

11000 0111

(11000 0111)₈₄₂₁ = (187)₁₀

(2) 27=0010 0111; 15=0001 0101

0010 0111

+0001 0101

0011 1100

+ 0000 0110

0100 0010

(0100 0010)₈₄₂₁ = (42)₁₀

3.20 分别用原码一位乘法及布斯算法求乘积 XY，要求写出计算过程。

(1) X= - 0.1101, Y= + 0.0110

(2) X= - 0.1110, Y= - 0.1101

答：

(1) X= - 0.1101, Y= + 0.0110

被乘数：[X]_原=1.1101，乘数：[Y]_原=0.0110

乘积符号位：1⊕0=1

CF	D	A A ₀	操作
0	0 0 0 0	0 1 1 0	A ₀ =0, +0
+ 0	0 0 0 0		
0	0 0 0 0		右移 1 位
0	0 0 0 0	0 1 1	A ₀ =1, + X
+ 0	1 1 0 1		
0	1 1 0 1		右移 1 位

0	0 1 1 0	1 0 0 1	$A_0=1, + X $
+ 0	1 1 0 1		
1	0 0 1 1		右移 1 位
0	1 0 0 1	1 1 0 0	$A_0=0, +0$
+ 0	0 0 0 0		
0	1 0 0 1		右移 1 位
0	0 1 0 0	1 1 1 0	

拼接符号得到: $[XY]_{\text{原}}=1.01001110$

布斯法求补码一位乘:

被乘数: $[X]_{\text{补}}=11.0011$, 乘数: $[Y]_{\text{补}}=00.0110$

$[-X]_{\text{补}}=00.1101$

符号	D	A A ₋₁	操作
00	0000	001100	$A_0 A_{-1}=00, +0$
00	0000		
00	0000		右移 1 位
00	0000	000110	$A_0 A_{-1}=10, +[-X]_{\text{补}}$
00	1101		
00	1101		右移 1 位
00	0110	100011	$A_0 A_{-1}=11, +0$
00	0000		
00	0110		右移 1 位
00	0011	010001	$A_0 A_{-1}=01, +[X]_{\text{补}}$
11	0011		
11	0110		右移 1 位
11	1011	001000	$A_0 A_{-1}=00, +0$
00	0000		
11	1011		右移 1 位

1 1	1 1 0 1	1 0 0 1 0 0	
-----	---------	-------------	--

$[XY]_{\text{补}} = 1.10110010$

对应的 $[XY]_{\text{原}} = 1.01001110$ ，与上面的计算结果一致。

(2) $X = -0.1110, Y = -0.1101$

被乘数: $[X]_{\text{原}} = 1.1110$, 乘数: $[Y]_{\text{原}} = 1.1101$

乘积符号位: $1 \oplus 1 = 0$

CF	D	A A ₀	操作
0	0 0 0 0	1 1 0 1	A ₀ =1, + X
+ 0	1 1 1 0		
0	1 1 1 0		右移 1 位
0	0 1 1 1	0 1 1 0	A ₀ =0, +0
+ 0	0 0 0 0		
0	0 1 1 1		右移 1 位
0	0 0 1 1	1 0 1 1	A ₀ =1, + X
+ 0	1 1 1 0		
1	0 0 0 1		右移 1 位
0	1 0 0 0	1 1 0 1	A ₀ =1, + X
+ 0	1 1 1 0		
1	0 1 1 0		右移 1 位
0	1 0 1 1	0 1 1 0	

拼接符号得到: $[XY]_{\text{原}} = 0.10110110$

布斯法求补码一位乘:

$X = -0.1110, Y = -0.1101$

被乘数: $[X]_{\text{补}} = 11.0010$, 乘数: $[Y]_{\text{补}} = 11.0011$

$[-X]_{\text{补}} = 00.1110$

符号	D	A A ₋₁	操作
00	0000	10011 0	A ₀ A ₋₁ =10, +[-X] _补
00	1110		
00	1110		右移1位
00	0111	01001 1	A ₀ A ₋₁ =11, +0
00	0000		
00	0111		右移1位
00	0011	10100 1	A ₀ A ₋₁ =01, +[X] _补
11	0010		
11	0101		右移1位
11	1010	1101 00	A ₀ A ₋₁ =00, +0
00	0000		
11	1010		右移1位
11	1101	01101 0	A ₀ A ₋₁ =10, +[-X] _补
00	1110		
00	1011		右移1位
00	0101	10110 1	

[XY]_补=0.10110110

对应的[XY]_原=0.10110110，与上面的计算结果一致。

3.21 分别用原码加减交替法和补码加减交替法完成二进制数 X 和 Y 的 X÷Y 运算，要求写出计算过程。

(1) X=-0.10101, Y=+0.11011

(2) X=+0.1001110001, Y=-0.10101

答：

3.22 已知 X=1101010110, Y=1000110011, 求 X·Y、X+Y、X⊕Y 的逻辑运算结果。

答：

X=1101010110

Y=1000110011

X·Y=1000010010

X+Y=1101110111

X⊕Y=0101100101

3.23 已知 X=1010110011, Y=0111001010, CF=1, 求对 X 逻辑左移 2 位、算术右移 3 位的结果, 以及 Y 不带进位循环左移 3 位和带进位循环右移 3 位的结果。

答:

X=1010110011, Y=0111001010, CF=1

X 逻辑左移 2 位: CF=0, X=1011001100

X 算术右移 3 位: CF=0, X=1111010110

Y 不带进位循环左移 3 位: CF=1, Y=1001010011

Y 带进位循环右移 3 位: CF=0, Y=1010111001

3.24 定点运算器内部总线互连有三种结构, 下面的描述中, D 用于三总线结构。

- A. 执行一次运算操作需要三步
- B. 在此运算器中至少需要设置两个暂存器
- C. 在运算器中的两个输入和一个输出上至少需要设置一个暂存器
- D. 在运算器中的两个输入和一个输出上不需要设置暂存器

3.25 浮点数的尾数用补码表示, 其 n 位 (包括 1 位符号位) 规格化负尾数的取值范围是多少?

答: $-1 \sim (-1/2 + 2^{-n+1})$

3.26 浮点数字长为 16 位, 其中, 阶码为 7 位 (包括 1 位阶符), 数符为 1 位, 尾数为 8 位。若阶码用移码表示, 尾数用补码表示, 则该浮点数所能表示的数值范围为多少?

答:

设浮点数字长 16 位, 基底为 2。其中阶码 6 位 (含 1 位阶符), 用移码表示; 尾数 10 位 (含 1 位数符), 用补码表示。

	真值	浮点数表示
最小数	$-1 \times 2^{31} = -2^{31}$	1, 11111 1.000000000
最大负数	$-2^{-9} \times 2^{-32} = -2^{-41}$	0, 00000 1.111111111
最小正数	$2^{-9} \times 2^{-32} = 2^{-41}$	0, 00000 0.000000001
最大数	$(1-2^{-9}) \times 2^{31}$	1, 11111 0.111111111

	真值	浮点数表示
最小负数	$-1 \times 2^{63} = -2^{63}$	1 111111 1 00000000
最大负数	$-2^{-8} \times 2^{-64} = -2^{-72}$	0 000000 1 11111111
最小正数	$2^{-8} \times 2^{-64} = 2^{-72}$	0 000000 0 00000001
最大正数		1 111111 1 00000000

3.27 浮点数字长为 12 位，其中，阶码为 5 位（包括 1 位阶符），尾数为 7 位（含 1 位数符）。阶码用移码表示，尾数用补码表示，阶码在前，尾数（包含数符）在后。请按照浮点数加减法的步骤计算 $X \pm Y$ 。

(1) $X = 11/16 \times 2^{-4}$, $Y = 35/64 \times 2^{-3}$

(2) $X = +0.101101B \times 2^{-11B}$, $Y = -0.100101B \times 2^{-01B}$

答：(1) $[X]_{\text{浮}} = 0.101100 \times 2^{01100}$ $[Y]_{\text{浮}} = 0.100011 \times 2^{01101}$

$$[X]_{\text{浮}} = 01100 \quad 0.101100 \quad [Y]_{\text{浮}} = 01101 \quad 0.100011$$

① 对阶

$$[\Delta E]_{\text{移}} = [E_x]_{\text{移}} - [E_y]_{\text{移}} = 01100 + 00011 = 01111$$

$\therefore X$ 阶码比 Y 的阶码小

$$X \text{ 的尾数右移 1 位, } [X]_{\text{浮}} = 00110 \quad 0.010110$$

② 尾数求和

$$\begin{array}{r} 00.010110 \\ + 00.100011 \\ \hline 00.11001 \end{array}$$

尾数求差

$$00.010110$$

$$\begin{array}{r} + 11.011101 \\ 11.110011 \end{array}$$

③规格化

$$[X+Y]_{\text{浮}} = 01101 \quad 0.111001$$

X-Y 的结果为非规格化尾数，需左规 2 位，阶码-2，阶码为 $01101+11110=01011$

$$[X+Y]_{\text{浮}} = 01011 \quad 1.001100$$

3.28 进行浮点数算术运算（加、减、乘、除）时，在什么情况下会产生上溢出？

答：加减乘除

3.29 图 3.38 为某运算器的简化框图。其中，A、B 既有寄存器的功能、又有多路选择器的功能，M 具有图中标功能及多路选择功能，ALU 为算术逻辑单元，R0 和 R1 为通用寄存器。图中粗线为数据通路，是一种单总线结构。试写出下列运算的步骤：

$$(1) (R0) + 2 (R1) \rightarrow R0$$

$$(2) 2 (R0) - (R1) \rightarrow R0$$

$$(3) (R0) - 1 \rightarrow R0$$

$$(4) (R0) \oplus (R1) \rightarrow R0$$

答：(1) $A \leftarrow R1; B \leftarrow R1; A \leftarrow A+B (A \rightarrow ALU, B \rightarrow ALU, ADD, V \text{ 直送}); B \leftarrow R0; R0 \leftarrow A+B;$

(2) $A \leftarrow R0; B \leftarrow R0; A \leftarrow A+B; B \leftarrow R1; R0 \leftarrow A + \bar{B}+1 (ALU \text{ 选择 ADD 和 } +1 \text{ 有效});$

(3) $B \leftarrow R0; A \text{ 设为全 } 1; R0 \leftarrow A+B;$

(4) $A \leftarrow R0; B \leftarrow R1; R0 \leftarrow A \oplus B (XOR);$

第 4 章 存储系统

4.1 试以单元电路说明为什么 DRAM 的功耗比 SRAM 小。

答：当电路加电且不进行读写及刷新操作时，电路中没有晶体管导通，也就没有电流流过，故功耗很小。

4.2 某存储器容量为 64KB，按字节编址，地址 C000H~FFFFH 为 ROM 区，其余为 RAM 区。若采用 16K×4 位的 SRAM 芯片进行设计，计算需要该 SRAM 芯片的数量。

答：地址 C000H~FFFFH 为 ROM 区：容量为 $2^{14}=2^4\text{KB}=16\text{KB}$

因此，RAM 区的容量为 $64-16=48\text{KB}$ ， $(48\text{K}/16\text{K})\times(8\text{b}/4\text{b})=6$ ，因此，需要 6 块 16K×4 位的 SRAM 芯片。

4.5 主存按字节编址，地址从 40000H~BBFFFH 共有多少字节（以 KB 为单位）？

答：

地址空间大小为 $\text{BBFFFH}-40000\text{H}+1=7\text{C000H}$ ，也即 7C000H 字节。

$(7\times 16+12)\times 2^{12}=124\times 4\text{KB}=496\text{KB}$

∴ 共有 496KB

4.6 试判断图 4.68 所示的存储器译码电路 74LS138 的输出 \overline{Y}_0 、 \overline{Y}_4 、 \overline{Y}_6 和 \overline{Y}_7 所决定的主存地址范围。

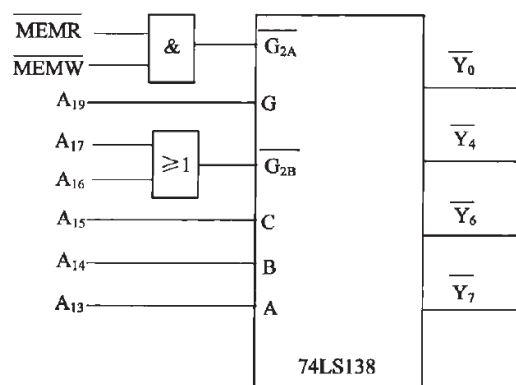


图 6.68 习题 4.6 附图

答：74LS138 正常工作时，使能端应该有效， $A_{19}=1$ ， $A_{17}A_{16}=00$

输出 \overline{Y}_0 有效时， $A_{15}\sim A_{13}=000$

⇒ 80000H~81FFFH, C0000H~C1FFFH

\overline{Y}_4 时, $A_{15} \sim A_{13} = 100$

⇒ 88000H~89FFFH, C8000H~C9FFFH

\overline{Y}_6 时, $A_{15} \sim A_{13} = 110$

⇒ 8C000H~8DFFFH, CC000H~CDFFFH

\overline{Y}_7 时, $A_{15} \sim A_{13} = 111$

⇒ 8E000H~8FFFFH, CE000H~CFFFFH

4.7 将 4 片 6264 芯片连接到 8086 系统总线上, 要求其主存地址范围为 70000H~77FFFH, 画出连接图。

仅了解, 不作考试要求

4.8 EEPROM 98C64A 芯片的各引脚功能是什么? 如果要将一片 98C64A 与 8 为 ISA 总线相连接, 并能随时改写 98C64A 中各单元的内容, 试画出连接电路图 (地址空间为 40000H~41FFFH)。

4.9 与 EPROM 相比, EEPROM 有什么不同?

答:

EPROM: 紫外线擦除可多次重写的只读存储器

EEPROM: 电擦除可多次编程只读存储器

EEPROM 在擦除及编程方面比 EPROM 更加方便, 随着 EEPROM 的集成度提高、价格下降, 使得 EPROM 逐渐退出了历史舞台。

4.10 已知 80486 主存最大寻址空间为 4GB, 按字节编址, 系统总线有 32 条数据线, 主存采用四个 8 位的存储体交叉编址, 各存储体的选择信号分别为 \overline{BE}_0 , \overline{BE}_1 , \overline{BE}_2 , \overline{BE}_3 。在 80486 系统中, 某 SRAM 芯片容量为 256K×8bit, 试用这样的芯片构成 40000000H~400FFFFFFH 的主存模块, 画出连接电路图。

4.11 以 DDR4 SDRAM 为例说明动态存储器的读、写过程。

答:

4.12 标准的动态存储器 DRAM 与同步动态存储器 SDRAM 的不同主要表现在哪些方面?

答：

DRAM (Dynamic Random Access Memory) 动态随机存储器

DRAM 只能保持很短的时间。为了保持数据，DRAM 使用电容存储，所以必须每隔一段时间刷新 (refresh) 一次。如果存储单元未刷新，存储的信息会丢失。

SDRAM (Synchronous Dynamic Random Access Memory) 同步动态随机存储器

同步是指存储器工作需要同步时钟。内部的命令发送与数据传输都以它为基准；动态是指存储阵列需要不断地刷新来保证数据不丢失；随机是指数据不是线性存储，而是自由制定地址进行数据读写。

4.13 DDR SDRAM 的 DDR 是什么意思？与一般的 SDRAM 相比有什么不同？

答：

DDR 是 Double Data Rate SDRAM 的缩写 (双倍数据速率)

一般的 SDRAM 只能在信号上升沿进行数据传输

DDR SDRAM 可以在信号上升沿和下降沿都进行数据传输，所以 DDR 可以在每个时钟周期完成两倍的 SDRAM 的数据传输量。

4.14 动态读写存储器 DRAM 在使用时需要定时刷新，其主要原因是什么？

答：DRAM 的每个存储元是靠电容上的电荷来存储信息的，“有电荷”和“无电荷”两种状态来表示“1”和“0”。通常情况下，电容上的电荷会慢慢减少。如果不给电容充电，电荷消失，那么存储的信息就会消失。所以要定时刷新，给电容充电，以保证存储的信息不会消失。

4.15 掉电后，下面说法正确的是 B

- A. RAM 的数据不会丢失
- B. ROM 的数据不会丢失
- C. EPROM 的数据会丢失
- D. DRAM 的数据不会丢失

4.16 某 CPU 地址总线为 $A_0 \sim A_{19}$ ，数据总线为 $D_0 \sim D_7$ ，主存读信号为 \overline{MEMR} 。主存写信号为 \overline{MEMW} 。某 SRAM 连接电路图如图 4.69 所示。

(1) 分析该图，请指出该 SRAM 芯片占用的主存地址范围。

(2) 采用该 SRAM 芯片实现 $30000H \sim 33FFFH$ 的主存区域，试画连接图。

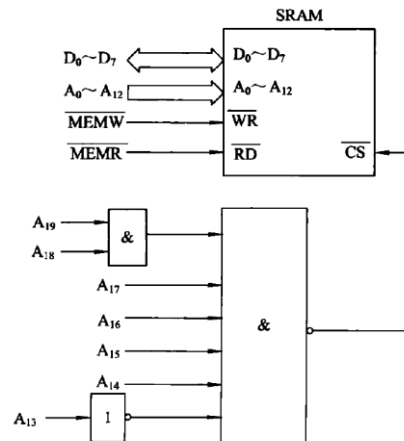


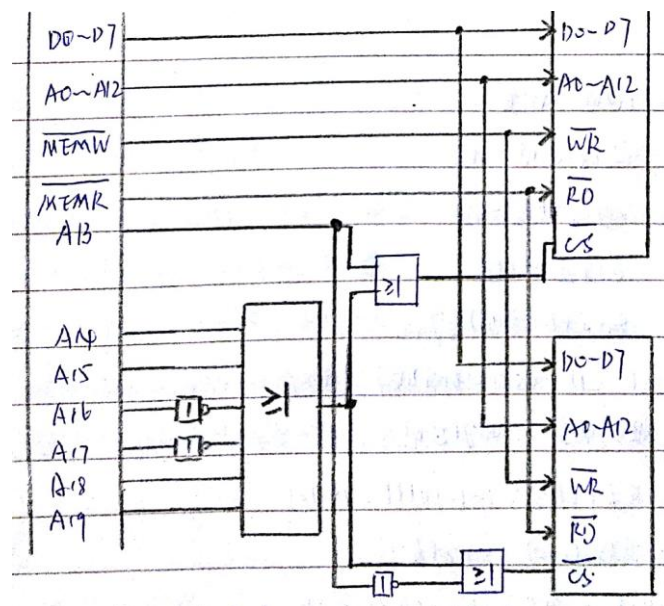
图 4.69 习题 4.16 附图

答：仅了解，不做考试要求。

(1) 1111 110x xxxx xxxx xxxx

地址范围：FC000H~FDFFFH

(2)



4.17 存储器的存取时间 t_A 和存储周期 t_m 的关系是 D

A. $t_A > t_m$ B. $t_A \geq t_m$ C. $t_A = t_m$ B. $t_A < t_m$

4.18 利用硬件测试工具软件，读取你计算机主存的参数以及 SPD 信息，比如主存类型、通道数，数据位宽、内存条数量、主存总容量、频率、SDRAM 时序参数，以及内存条的制造商、制造日期、模块名称等信息，说明这些参数的含义。

4.19 某 DDR4 SDRAM 芯片总容量为 8Gb（即 8G 个二进制位），有 A、B、C 三种型号，对应其存储单元的三种不同组织形式：

型号 A：2G×4b

型号 B：1G×8b

型号 C：512M×16b

请问，要构成单个 Rank 的内存条，用哪个型号的芯片可以获得最大的主存容量？

4.20 设计一个具有 8 个存储单元的读相联存储器，画出电路框图，并说明其工作过程。

4.21 主存与 CPU 之间增加高速缓存 Cache 的目的是 A

A. 解决主存与 CPU 之间的速度匹配问题 B. 扩大主存的存储容量 C. 扩大外存储器的寻址空间 D. 提高外部存储器的速度。

4.22 在计算机中，采用虚拟存储器的目的是 B

A. 提高内部存储器的速度问题 B. 扩大内部存储器的寻址空间 C. 扩大外部存储器的寻址空间 D. 提高外部存储器的速度。

4.23 在计算机中，CPU 对其访问速度最快的是 C

A. 主存 B. Cache C. CPU 中的通用寄存器 D. 硬盘

4.24 若 Cache 以字为块，其存取时间为 10ns，主存的存取时间为 100ns，存储系统的平均存取时间为 16ns，则 Cache 的命中率为 C

A. 96% B. 95% C. 94% D. 92%

解析： $16 = 10 + (1 - H) * 100$

∴ H=94%

4.25 高速缓存 Cache 采用 4 路组相联地址映射方式，每路 2 块，每块 1KB。主存最大寻址空间为 1MB，按字节编码。

（1）说明主存地址中的标记、索引、块内地址字段各用多少位表示。

(2) 已知地址映射表如图 4.70 所示，若主存地址为 ABCDEH 和 12345H，请问是否命中 Cache？如果未命中，说明理由；如果命中，试确定命中第几路 Cache，路内地址是什么。

第1路		第2路		第3路		第4路	
有效位	标记	有效位	标记	有效位	标记	有效位	标记
0	058H	1	112H	1	067H	1	157H
1	188H	0	157H	1	157H	1	167H

图 4.70 习题 4.25 附图

答：

- (1) 主存 1MB：地址 20 位
- 每块 1KB：块内地址 10 位
- 每组有 4 路，即 4 块，组内路号 2 位
- 2 组，区内组号 1 位
- 主存区号 $20 - (10 + 2 + 1) = 7$ 位。

- (2) ABCDEH ： 10101011110011011110
- 1010101 1 11 0011011110
- 区号 组号 路号 块内地址
- 1 组，区号+块号：101010111=157H
- 查地址变化表：1 组 3 路
- Cache 地址为：组号+路号+块内地址=1 1100 1101 1110=1CDEH

4.26 高速缓存 Cache 与主存间采用全相联地址映射方式，高速缓存的容量为 4KB，分为 4 块，主存容量为 1MB。

- (1) 地址映射表为____个存储单元，每个存储单元至少包含____位。
- (2)若主存读写时间为 300ns,高速缓存读写时间为 30ns,而存储系统平均读写时间为 32.7ns,则该高速缓存的命中率为____%。
- (3) 若地址映射表如图 4.71 所示，试根据主存地址确定变换后的高速缓存的地址。

0	388H
1	222H
2	159H
3	367H

图 4.71 习题 4.26 附图

① 主存地址为 88888H，高速缓存地址为_____H。

② 主存地址为 56789H，高速缓存地址为_____H。

(4) 简要说明采用全相联映射的优缺点（主要从 Cache 的利用率、命中率、地址变换硬件的繁简等方面进行说明）

答：

(1) 4,10

主存 1M，即 20 位；每块 1KB，即块内地址为 10 位

主存块号 10 位，块内地址 10 位。

(2) $30 \times H + (1-H) \times 300 = 32.7$ ，所以 $H=99\%$

(3) $88888H \Rightarrow 1000100010 \quad 0010001000$

因此，主存块号 222H，查地址变换表为第 1 块

Cache 地址为 01 0010001000，即 488H

$56789H \Rightarrow 0101011001 \quad 1110001001$

因此，主存块号 159H，查地址变换表为第 2 块

Cache 地址为 10 1110001001，即 B89H

(4) 优点：Cache 利用率高，命中率

缺点：需要相联存储器容量大，地址变换硬件复杂。

4.27 某计算机高速缓存 Cache 采用 2 路组相联地址映射方式，每路 4 块，块的大小为 512B。

主存最大寻址空间为 1MB，按字节编码。构成高速缓存地址映射表的相联存储器容量为(1) bit，每次参与比较的存储单元为(2) 个。

(1) A. 4×10 B. 8×10 C. 4×9 D. 8×9

(2) A. 1 B. 2 C. 4 D. 8

答：(1) B (2) C

主存有 1MB，20 位地址；每块 512B，块内地址为 9 位，块号 2 位，路号 1 位

∴主存区号为 8 位。

每条记录 区号+块号=8+2=10 位，共 8 条记录

所以地址变换表的相联存储器容量为 $8 \times 10 \text{bit}$

每次参与比较的存储单元为 4 个。

4.28 使 Cache 命中率最高的替换算法是： **C**

- A. 先进先出算法 FIFO
- B. 随机替换算法 RAND
- C. 最优替换算法 OPT
- D. 近期最少使用 LRU

4.29 主存的段式管理有许多优点，下面的描述中， **C** 不是段式管理的优点。

- A. 支持程序的模块化设计和并行编程的要求
- B. 各段程序的修改互不影响
- C. 地址变换速度快、主存碎片（零头）小
- D. 便于多道程序共享主存的某些段。

4.30 若 Cache 容量为 100 字，并以 50 字分块，起始为空。CPU 从主存单元 0、1、2、...、99 中每次读出一个字，顺序读出 100 个字，并重复读 10 次进行主存访问，则命中率为多少？若主存访问时间 T_M 是 Cache 访问时间 T_C 的 5 倍，则在此情况下比无 Cache 时的速度提高了多少倍？

答：由于每个块中有 50 字，且 Cache 初始为空，因此 CPU 读 0 号单元未命中，必须访问主存，同时将字所在的内存块调入 Cache 的第 0 组的任一块内。接着 1-49 号单元均命中，同理，第 50 号单元也未命中。而后循环 9 次全部命中。

命中率= $(10 \times 100 - 2) / (10 \times 100) \times 100\% = 99.8\%$

提高倍数： $5 \times 1000 / 998 \times 1 + 5 \times 2 - 1 = 3.96$ 倍

4.31 主存容量为 4MB，虚存空间为 1GB，按字节编址，实地址与虚地址各为多少？若页面大小为 4KB，则在主存中的页表应有多少个表项？页表应多大？

答：

主存空间为 4MB，实地址为 22 位

虚存空间为 1GB，虚地址为 30 位

页面大小为 4KB，则 $1GB / 4KB = 256K$ (个)表项

4.32 段式虚拟存储器的段表如表 4.14 所示。若访问段 3 的 1059H 单元和段 2 的 1678H 单

				命中			命中	命中		命中	命中		命中	
--	--	--	--	----	--	--	----	----	--	----	----	--	----	--

(2) FIFO 算法: 命中率=6/15=40%

P3	P3	P3	P3	P3	P3	P7	P7	P7	P7	P7	P7	P7	P7	P6
	P4	P4	P4	P4	P4	P4	P4	P3	P3	P3	P3	P3	P3	P3
		P2	P2	P2	P2	P2	P2	P2	P2	P2	P4	P4	P4	P4
			P6	P6	P6	P6	P6	P6	P6	P6	P6	P8	P8	P8
				命中	命中		命中		命中	命中			命中	

LRU 算法: 命中率=9/15=60%

P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3
	P4	P4	P4	P4	P4	P4	P4	P4	P4	P4	P4	P4	P4	P4
		P2	P2	P2	P2	P7	P7	P7	P7	P7	P7	P8	P8	P8
			P6	P6	P6	P6	P6	P6	P6	P6	P6	P6	P6	P6
				命中	命中		命中	命中	命中	命中	命中		命中	命中

4.35 有如下 C 语言程序段:

```
for(k=0;k<1000;k++)
```

```
    a[k]=a[k]+10;
```

若数组 a 及变量 k 均为 int 型, int 型数据占 4B, 变量 k 在寄存器中; 数据 Cache 采用直接映射方式, 容量为 1KB, 块大小为 32B, 该程序段执行前 Cache 为空。求该程序段执行过程中访问数组 a 的 Cache 缺失率。

解: a[k] 的访问步骤是: 先访问 cache, cache 缺失, 之后从主存中取出一个块调入 cache, 这个块中的后几个数据都是命中的, 本题中一个数据占 4B, 一个块大小是 32B, 这说明一个块中有 8 个数据, 关键是后面还有一次写, 这说明一次循环要 16 次访问 cache, 其中只有

第一次是缺失的，后面 15 次都是命中的，所以缺失率是 $1/16=6.25\%$ ；

4.36 某计算机采用页式虚拟存储管理方式，按字节编址，虚拟地址为 32 位，物理地址为 26 位，页大小为 4KB；TLB 采用全相联映射；Cache 数据区大小为 64KB，按 2 路组相联方式组织，主存块大小为 64B。图 4.72 为存储访问过程的示意图。请回答下列问题。

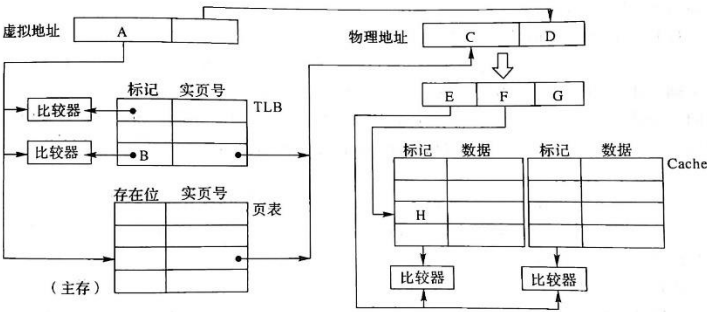


图 4.72 习题 4.36 附图

- (1) 图 4.72 中 A~G 的位数各是多少？TLB 标记字段 B 中存放的是什么信息？
- (2) 将块号 4102 的主存块装入到 Cache 中时，所映射的 Cache 组号是多少？对应的 H 字段内容是什么？
- (3) Cache 缺失处理的时间开销大还是缺页处理的时间开销大？为什么？
- (4) 为什么 Cache 可以采用写直达 (Write Through)策略，而修改页面内容时总是采用写回 (Write Back)策略？

解：

(1)页大小为 4 KB，页内偏移地址为 12 位，故 $A=B=32-12=20$ ； $D=12$ ； $C=26-12=14$ ；
主存块大小为 64 B，故 $G=6$ 。2 路组相联，每组数据区容量有 $64\text{ B}\times 2=128\text{ B}$ ，共有 $64\text{ KB}/128\text{ B}=512$ 组，故 $F=9$ ； $E=26-G-F=26-6-9=11$ 。

因而 $A=20$ ， $B=20$ ， $C=14$ ， $D=12$ ， $E=11$ ， $F=9$ ， $G=6$ 。
TLB 中标记字段 B 的内容是虚页号，表示该 TLB 项对应哪个虚页的页表项。

(2)块号 $4102=00\ 0001\ 0000\ 0000\ 0110\text{B}$ ，因此，所映射的 Cache 组号为 $0\ 0000\ 0110\text{B}=6$ ，对应的 H 字段内容为 $0\ 0000\ 1000\text{B}$ 。

(3)Cache 缺失带来的开销小，而处理缺页的开销大。因为缺页处理需要访问磁盘，而

Cache 缺失只要访问主存。

(4)因为采用直写策略时需要同时写快速存储器和慢速存储器，而写磁盘比写主存慢得多，所以，在 Cache-主存层次，Cache 可以采用直写策略，而在主存-外存(磁盘)层次，修改页面内容时总是采用回写策略。

4.37 某计算机的主存地址空间大小为 1GB，按字节编址。指令 Cache 和数据 Cache 分离，均有 16 个 Cache 块，每个 Cache 块大小为 64B,数据 Cache 采用直接映射方式。现有两个功能相同的程序 A 和 B,其伪代码如图 4.73 所示。

<pre>int a[256][256] ... int sum_array1() { int i,j,sum=0; for(i=0;i<256;i++) for(j=0;j<256;j++) sum += a[i][j]; return sum; }</pre>	<pre>int a[256][256] ... int sum_array2() { int i,j,sum=0; for(j=0;j<256;j++) for(i=0;i<256;i++) sum += a[i][j]; return sum; }</pre>
程序 A	程序 B

图 4.73 习题 4.37 附图

假定 int 类型数据用 32 位补码表示，程序编译时，i、j、sum 均分配在寄存器中，数组 a 按行优先方式存放，其首地址为 768(十进制数)，且程序执行之前数组 a 的任何元素均不在 Cache 中。请回答下列问题，要求说明理由或给出计算过程。

(1)若不考虑用于 Cache-致性维护和替换算法的控制位，则包括地址映射表在内，数据 Cache 的总容量为多少？

(2)数组元素 a[0][62]和 a[1][150]各自所在的主存块对应的 Cache 块号分别是多少（Cache 块号从 0 开始）？

(3)程序 A 和 B 的数据访问 Cache 命中率各是多少？哪个程序的执行时间更短？

答：(1) $1\text{GB}/(64\text{B}\times 16)=2^{20}$

主存地址格式为：

20 位	4 位	6 位
------	-----	-----

区号	块号	块内地址
----	----	------

Cache 地址格式:

4 位	6 位
块号	块内地址

若不考虑用于 Cache-致性维护和替换算法的控制位,则每个 cache 行要占用 1 位的有效位、20 位的区号和 64B (即 512 位) 的数据位, 因此数据 cache 的总容量为:

$$16 \times (1+20+512) = 8528 \text{ 位} = 1066\text{B}$$

地址映射表有 16 行, 每行存储主存的区号需要 20 位, 因此, 地址映射表的容量为 $16 \times 20 = 40\text{B}$

因此, 包括地址映射表的话, 数据 cache 的总容量为 1106B

(2) 某一元素对应的 cache 块号为: $((\text{开始地址} + (\text{行号} * \text{一行元素个数} + \text{列号}) * \text{数据种类的长度}) / \text{主存块大小}) \% (\text{cache 行个数} / \text{几路相联映射})$

数组元素 a[0][62]所在的主存块对应的 Cache 块号:

$$((768 + (0 \times 256 + 62) \times 4) / 64) \bmod (16/2) = 7$$

数组元素 a[1][150]所在的主存块对应的 Cache 块号:

$$((768 + (1 \times 256 + 150) \times 4) / 64) \bmod (16/2) = 5$$

(3) 每个 cache 块包含 16 个用 32 位补码表示的整数, 并且按行优先方式存放。

对于程序 A, 每次 cache 不命中时, 将从主存中调入一个 cache 块, 由于数组元素按行的方式访问, 则接下来对该 cache 块中的其他 15 个元素的访问均会命中, 所有 cache 块均会被依次访问 16 个元素且不重复(一次不命中), 则访问数据 cache 的命中率为 $15/16 = 93.75\%$ 。

对于程序 B, 每次 cache 不命中时, 将从主存中调入一个 cache 块, 由于数组元素按列的方式访问, 依次访问的元素均位于不同的 cache 行中, 由于 cache 空间只能存放 16 个 cache 行, 每次访问 cache 不命中时调入 cache 的块还没等到第二次访问就被其他的 cache 块所替换, 则访问数据 cache 的命中率为 0。

由于执行程序 A 的 cache 的命中率高, 因此程序 A 的执行时间更短。

4.38 某计算机主存按字节编址，虚拟（逻辑）地址空间大小为 256MB，主存（物理）地址空间大小为 16MB，页面大小为 4KB；Cache 采用直接映射方式，共 8 块；主存与 Cache 之间交换的块大小为 32B。系统运行到某一时刻时，页表的部分内容和 Cache 地址映射表的内容分别如图 4.74(a)、图 4.74(b)所示，图中实页号及标记字段的内容为十六进制形式。请回答下列问题：

- （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？哪几位表示实页号？
- （2）使用物理地址访问 Cache 时，物理地址应划分成哪几个字段？说明每个字段的位数及在物理地址中的位置。
- （3）虚拟地址 00056A8H 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？访问该物理地址时是否命中 Cache？如果命中 Cache，对应的 Cache 内部地址是什么？请写出计算过程并说明理由。
- （4）假定为该计算机配置一个 4 路组相联的 TLB，共可存放 8 个页表项，若其当前内容（十六进制）如图 4.74(c)所示，则此时虚拟地址 00356A8H 所对应的页表项是否命中 TLB？若未命中 TLB，说明理由；若命中 TLB，求该虚拟地址对应的物理地址，并写出计算过程。

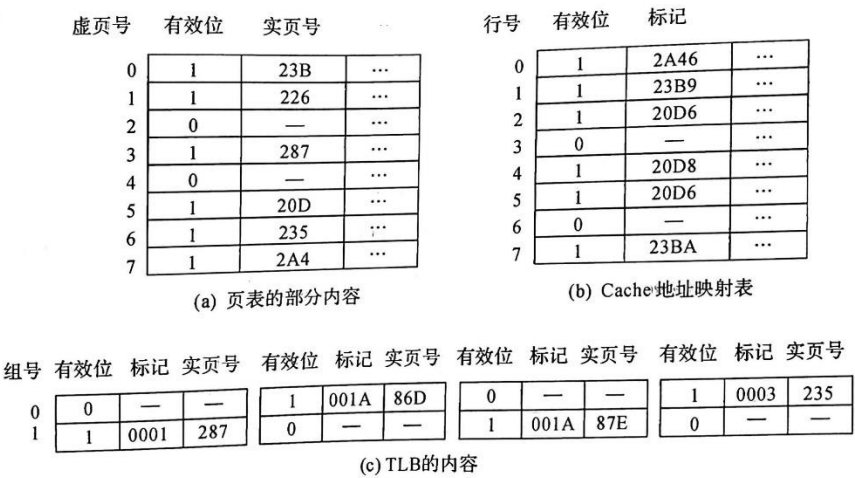


图 4.74 习题 4.38 附图

解：（1）由于虚拟地址空间大小为 256MB，且按字节编址， $2^{28}=256M$ ，所以虚拟地址的长度应为 28 位。

页面为 4KB，则页内地址为 12 位。因此，虚页号为 $28-12=16$ 位。即虚地址的高 16 位是虚页号。

主存地址大小 16MB，按字节编址， $2^{24}=16M$ ，因此，物理地址的长度为 24 位。页内地址为 12 位， $24-12=12$ ，因此实地址的高 12 位为实页号。

(2) 使用物理地址访问 Cache 时，由于采用的是直接映射，物理地址是 24 位，应划分成主存区号、块号、块内地址 3 个字段。

Cache 块大小为 32B，需要 5 位块内地址；

Cache 有 8 块，需要 3 位的块号；

则标记（区号）占了 $24-3-5=16$ 位。

(3) 虚拟地址 $00056A8H=0000\ 0000\ 0000\ 0101\ 0110\ 1010\ 1000B$ ，虚页号为 0005H，查页表发现，虚页号为 5 对应的有效位为 1，表明此页在主存中，实页号为 20DH，对应的物理地址为 $20D6A8H=0010\ 0000\ 1101\ 0110\ 1010\ 1000B$ 。该地址对应的块号为 $101B=5$ ，查 cache 地址映射表，第 5 行对应的区号为 20D6，因此命中。Cache 的地址为： $10101000B=A8H$ 。

(3) 虚拟地址 $00356A8H=0000\ 0000\ 0011\ 0101\ 0110\ 1010\ 1000B$ ，虚页号为 0035H，TLB 中存放 8 个页表项，采用 4 路组相连，即 TLB 分成 2 组，每组 4 个页表项。16 位虚页号字段中，最低位作为组索引，其余 15 位为标记位。现在最低位为 1，表明选择第 1 组，15 位的标记位为 001A，第 2 路中的第 1 组标记匹配，TLB 命中。对应的实页号为 87EH，因此，物理地址为 87E6A8H。

4.39 当 RISC-V 处理器分别工作在 Sv32、Sv39 和 Sv48 三种不同的虚拟地址管理机制下时，计算其多级页表可能占用的最大存储空间，并说明如何解决页表庞大的问题。

4.40 磁盘位密度是盘片上 B。

- A. 最外圈磁道圆周上单位长度内存储的二进制位的个数
- B. 最靠近盘心的磁道圆周上单位长度内存储的二进制位的个数
- C. 从内向外的中间磁道圆周上单位长度内存储的二进制位的个数
- D. 所有磁道圆周上单位长度内存储的二进制位个数的平均值

4.41 磁盘的平均等待时间通常是指 A。

- A. 磁盘旋转半周所用的时间
- B. 磁盘旋转 $1/3$ 周所用的时间
- C. 磁盘旋转 $2/3$ 周所用的时间
- D. 磁盘转一周所用的时间

4.42 图 4.75 所示为同一数据的磁头写入电流的三种波形，试说明每种波形对应的记录方式名称及它们的自同步能力。

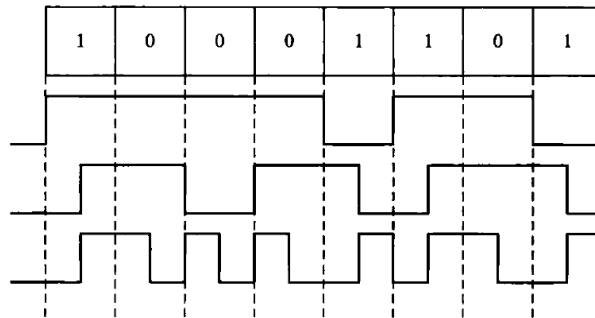


图 4.75 习题 4.42 附图

答：

- (1) 有“1”则变 NRZI 无自同步能力
- (2) MFM 有自同步能力
- (3) PM 有自同步能力

4.43 某硬盘有 5 个记录面，记录面上有效记录区域的内径为 20cm,外径为 30cm,磁道上记录的位密度为 250b/mm,道密度为 10 道 / 毫米,每一磁道上分为 16 个扇区,每个扇区记录 1KB,磁盘旋转速度为 10000r/min,则该硬盘的非格式化容量为 (1) MB; 格式化容量为 (2) MB; 该硬盘的数据传输速率约为 (3) MB/s.(计算中，1M=2²⁰.)

- (1) A.37 B.42 C.47 D.56
- (2) A.25 B.29 C.33 D.39
- (3) A.1.8 B.2.2 C.2.6 D.3.1

答：

- (1) C 非格式化容量: $3.14 \times 200 \times 250 \times (300-200) / 2 \times 10^3 \times 5 / 8 = 49062500 = 46.79 \text{ MB}$
- (2) D $(300-200) / 2 \times 10^3 \times 5 \times 16 \times 1 \text{ KB} = 40000 \text{ KB} = 39 \text{ MB}$
- (3) C $10000 \times 16 \times 1 \text{ KB} / 60 = 2666.67 \text{ KB/s} = 2.6 \text{ MB/s}$

4.44 若某磁盘有两个记录面，存储区的内径为 5cm,外径为 10cm,道密度为 500 道 / 厘米，内径上的位密度为 24000 b/cm,则该磁盘的每个记录面上有 (1) 条磁道，每条磁道上能存储的字节数约为 (2)。

- (1) A.750 B.1 250 C. 1 750 D. 2 500

(2) A.40 750 B.41 250 C. 43 750 D.47 120

答:

(1) B $500 \times (10-5)/2=1250$

(2) D $24000 \times 3.1415 \times 5/8=47122$

4.45 光盘驱动器与主机的接口总线通常采用(1)总线。只读光盘上的数据是记录在光盘表面的(2)上。

(1) A.ISA B.RS-232 C.ATA D.PCI

(2) A.集成电容 B.磁性材料 C.集成电阻 D.凹坑和平面

答:

(1) C

(2) D

4.46 某硬盘有 3 个盘片,共有 4 个记录面,转速为 7200r/min,盘面有效记录区域的外径为 30cm,内径为 10cm,记录位密度为 250b/mm,磁道密度为 8 道 / 毫米,每磁道分 16 个扇区,每扇区 512 字节,则该硬盘的非格式化容量和格式化容量分别约为(1),数据传输率约为(2)。若一个文件超出一个磁道容量,则剩下的部分(3)。(计算中, $1K=2^{10}$, $1M=2^{20}$)

(1) A.120 MB 和 100 MB B.30 MB 和 25 MB C.60 MB 和 50 MB D.22.5MB 和 25 MB

(2) A.3840 KB/s B.2880 KB/s C.1920 KB/s D.960 KB/s

(3) A.存于同一盘面的其他编号的磁道上

B.存于其他盘面的同一编号的磁道上

C.存于其他盘面的其他编号的磁道上

D.随机存放

答:

(1) B

非格式化容量: $2 \times 3.14 \times 5 \times 2500 \times [(30-10)/2] \times 8 \times 10^4 \times (1/8) = 29.9 \approx 30\text{MB}$

格式化容量: $16 \times 512 \times 8 \times (30-10)/2 \times 10^4 = 25\text{MB}$

(2) D $16 \times 512 \times 7200/60=960\text{KB/s}$

(3) A

4.47 题 4.46 给出的磁盘具有 10ms 的平均寻道时间，如果读写一个扇区，平均访问时间是多少？如果读写一个柱面，平均访问时间又是多少？

读写一个扇区：访问时间 = 寻道时间 + 等待时间 + 传输时间
=10ms+1/(2*120)s+(1/120)/16s=14.69ms

读写一个柱面：10ms+1/(2*120)s+(1/120)*4s=47.5ms

4.48 RAID 分为几级？哪些是用来提高性能的？哪些是用来提高可靠性的？

RAID级别	RAID0	RAID1	RAID3	RAID5	RAID10
别名	条带	镜像	专用奇偶位条带	分布奇偶位条带	镜像阵列条带
容错性	无	有	有	有	有
冗余类型	无	镜像	奇偶校验	奇偶校验	镜像
备盘	无	有	有	有	有
读性能	高	低	高	高	中间
随机写性能	高	低	最低	低	中间
连续写性能	高	低	低	低	中间
需要的磁盘数	2个或更多	2个或2N个	3个或更多	3个或更多	4个或2N (N≥2)
可用容量	总的磁盘容量	磁盘容量的50%	磁盘容量的 (N-1) /N	磁盘容量的 (N-1) /N	磁盘容量的50%

第 5 张 指令系统

5.1 解释术语：指令、指令系统、指令字长、操作码、操作数、寻址方式、CISC、RISC

答：

指令：控制计算机硬件完成指定的基本操作的命令。

指令系统：能够被一台计算机执行的全部指令的集合。

指令字长：机器指令中二进制代码的总位数。

操作码：计算机程序中所规定的要执行操作的那一部分指令或字段，其实就是指令序列号用来告诉 CPU 需要执行哪一条命令。

操作数：运算符作用于的实体，是表示式中的一个组成部分，它规定了指令中进行数字运算的量。

寻址方式：指令获取操作数的方式

CISC：复杂指令集计算机

RISC：精简指令集计算机

5.2 指令由哪几部分构成？各部分的功能是什么？

答：

指令由操作码和地址码组成；操作码指明指令要完成的功能，地址码又称操作数地址或操作数，用来提供读指令的操作对象。

5.3 操作码编码的方法有哪几种？各种方法的编码依据是什么？

答：

指令操作码的编码可以分为固定长度操作码和可变长度操作码两类。

固定长度操作码：固定长度操作码是指操作码所占的二进制位数固定不变，而且集中放在指令字的一个字段中。

可变长度操作码：可变长度操作码是操作码扩展技术的应用，即操作码的长度是可变的，且分散地放在指令的不同字段中。

操作码的编码方式有：

定长编码法：指令的操作码部分采用固定长度的编码。如：假设操作码固定为 7 位，则系统最多可表示 128 种指令。

优化编码法（huffman 编码法）：对出现频度较高的信息单元用较短的编码表示，而对频率较低的信息单元用较长的编码表示。

扩展编码法：结合定长编码法和 Huffman 编码法的思想，将操作码的编码长度分成几种固定长的格式。被大多数指令集采用。PDP-11 是典型的变长操作码机器。

5.4 采用扩展操作码设计方法的目的是什么？设计原则是什么？

答：采用扩展操作码设计方法的目的是保持指令字长不变的基础上，增加指令数量。

设计原则是

- （1） 若指令字长固定，则长地址码对应短操作码操作码长度随地址码长度缩短而增加。
- （2） 若指令字长可变，则以指令使用频度（指令在程序中出现的概率）作为设计依据，使用频度高的用短操作码，使用频度低的指令用长操作码（Huffman 编码原理）
- （3） 设计总是从短操作码开始，并要保证当前的操作码编码与未来要扩展的操作码编码能够有效进行区分。

5.5 指令系统中对操作数采用不同寻址方式的目的是什么？

答：缩短指令长度；扩展寻址空间；提高编程的灵活性。

5.6 指令长度如何确定？其设计的基本原则是什么？

答：指令长度的确定与 CPU 的复杂度、主存结构、数据传输宽度等有关。

指令长度设计的一般原则：

- 1) 短的操作码与多地址码字段配合，长的操作码与简单地址码组合；
- 2) 指令长度一般设计为总线宽度的整数倍；
- 3) 指令长度为存储器最小可寻址单位的整数倍。

5.7 依据地址码数量可以将指令格式分为哪几种？各指令格式的操作特点是什么？

答：按照地址码字段的数量，指令也可以分为四地址指令、三地址指令、两地址指令、一地址指令和零地址指令，其指令格式如图所示。

四地址指令	Op	Addr1	Addr2	Addr3	Addr4
三地址指令	Op	Addr1	Addr2	Addr3	

二地址指令	Op	Addr1	Addr2	
一地址指令	Op	Addr1		
零地址指令	Op			

四地址指令：op rd, rs1, rs2, ni; rs1 op rs2→rd, ni 提供顺序或转移地址

三地址指令：op rd, rs1, rs2; rs1 op rs2→rd, PC 提供顺序地址

二地址指令：op rd, rs1; rd op rs1→rd 或 rs1 op ACC→rd, PC 提供顺序地址

一地址指令：op rd; rd op ACC→ACC 或 rd 自身操作→rd, PC 提供顺序地址；或者 rd 提供转移地址。ACC 为累加器或操作码隐含指定的操作数

零地址指令：op; 由操作码指定操作数（隐含寻址）或无需操作数

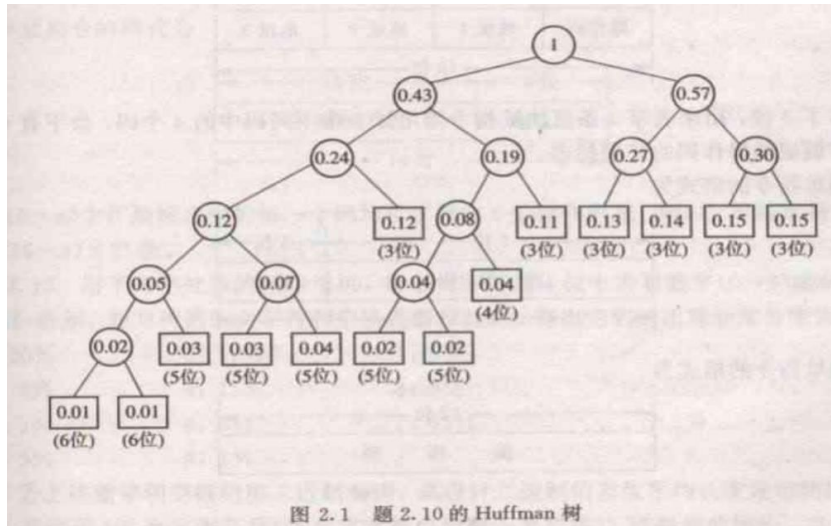
5.8 假设某计算机系统有 14 条指令，各条指令的使用频度分别是：0.01、0.15、0.12、0.03、0.02、0.04、0.02、0.04、0.01、0.13、0.15、0.14、0.11、0.03。试给出定长操作码、霍夫曼编码、只用两种码长的扩展操作码三种编码方案，并计算各种编码方案的平均操作码长度。

分析：登场操作码的意思是不管其指令系统的频度如何，都用同样长度的二进制码位数来对指令操作码编码。显然，当指令系统中的指令条数为 n 时，等长操作码的二进制码位数应当使 $\lceil \log_2 n \rceil$ 。Huffman 编码是 Huffman 算法构造出 Huffman 树来得到的。它的平均码长是用 $\sum_{i=1}^n P_i l_i$ 计算求得的。

只有两种码长的扩展操作码编码，则需要对指令使用频度进行按大小分群。将高频的质量分在同一群中，对其用短的操作码，而将低频的指令分在另一个群中，使用长的操作码编码。每一群都各自用等长操作码编码。这时，为能唯一解码和立即解码，在短操作码中还要使用某些码来作为扩展成长操作码的扩展标识码。经过综合权衡，使平均码长尽可能的短，来订好长、短码的码长组配关系。从而，再用 $\sum_{i=1}^n P_i l_i$ 来求得其平均码长。

解答：14 条指令的等长操作码的平均码长是 $\lceil \log_2 14 \rceil$ 位，即 4 位。

Huffman 编码可先用 Huffman 算法构造出 Huffman 树。如下图所示。图中，叶子上用圆括号所括起来的数字是表示该频度指令所用的二进制编码的码位数，所以 Huffman 编码的操作码平均码长为 $\sum_{i=1}^{14} P_i l_i = 3.38$ 位。



采用只有两种码长的扩展操作码，可根据 14 条指令所给出的使用频度值分成两群。让使用频度较高的 6 中指令用 3 位操作码编码表示。例如，用 000~101 分别表示使用频度为 0.15、0.15、0.14、0.13、0.12、0.11 的指令的操作码。留下 110 和 111 两个 3 位码作为长码的扩展标志，扩展出 2 位码。从而用 5 位码就可以各扩展出 4 条使用频度较低的指令，这样，共有 8 个使用频度较低的指令符合题目的要求。由此可求得操作码的平均码长为

$$\sum_{i=1}^n P_i l_i = 3 \times 0.80 + 5 \times 0.20 = 3.4$$

5.9 某指令系统共有 200 条指令。统计结果表明，传送类指令占 5%，使用频度为 50%；运算类指令占 10%，使用频度为 25%；分支跳转类指令 20%，使用频度为 15%；其余指令使用频度为 10%。试用扩展操作码编码方法为各类指令设计操作码编码，给出每类指令操作码的最短长度及相应的编码，并计算平均操作码长度。

答：

频度	类别	条数	操作码编码			
50%	传送类	200*5%=10	0000			
			↓			
			1001			
25%	运算类	200*10%=20	1010	00		
			↓			
			1110	11		
15%	分支跳转	200*20%=40	1111	00	0000	

	类		↓			
			1111	10	0111	
10%	其他指令	200-70=130	1111	10	1000	000
			↓			
			1111	11	1000	001

平均操作码长度=4×50%+ 6×25%+ 10×15%+10%×13=6.3 (位)

5.10 若某机器指令系统要求有如下形式的指令：三地址指令 4 条，单地址指令 255 条，零地址指令 16 条。设指令字长固定为 12 位，每个地址码长 3 位，那么能否以扩展操作码进行编码？如果单地址指令为 254 条，其他不变，是否可以进行扩展操作码编码？说明理由。

答：三地址 4 条， $12-3*3=3$ ，3 位编码，单地址指令 255 条， $12-3=9$ ， $(2^3-4)*2^6=256$ ，够用。 $(256-255)*2^3=8$ ，即只能编码 8 个零地址指令。

如果单地址指令为 254 条，则 $(256-254)*2^3=16$ ，可以对 16 条的零地址指令进行编码。

5.11 某计算机指令字长 16 位，设有单地址指令和双地址指令两类，若每个地址字段均为 6 位，且双地址指令有 m 条，那么单地址指令最多可以有多少条？

答： $(2^4-m)*2^6$

5.12 某指令系统指令字长 16 位，有零地址、一地址、二地址、三地址指令格式，每个地址字段均为 4 位，且三地址指令有 L 条，二地址指令有 M 条，零地址有 N 条。若采用定长操作码设计方法，一地址指令最多可以有多少条？若采用扩展操作码设计方法，一地址指令最多可以有多少条？

答： $16-3*4=4$ 位，操作码有 4 位可用，因此可以编码的指令条数： $2^4=16$ ，因此，若采用定长操作码设计方法，一地址指令最多可以有 $(16-L-M-N)$ 条。

若采用扩展操作码设计方法，一地址指令最多可以有：

$$\{[(2^4-L)*2^4-M]*2^4-x\}*2^4=N, \quad x=[(2^4-L)*2^4-M]*2^4-N/2^4$$

5.13 在一个 36 位字长的指令系统中，用扩展操作码表示下列指令：7 条具有两个 15 位地址

和一个 3 位地址的指令，500 条具有一个 15 位地址和一个 3 位地址的指令，50 条无地址指令。试为每类指令设计操作码长度及编码。

答：三地址指令 $7 < 2^3$, 000~110

二地址指令: $(2^3-7)*2^{15} > 500$, $512=2^9$, 因此低 9 位足够编码

111 000 0000 0000 0000 ~ 111 000 0001 1111 0011

5.14 对下列数据结构，给出数据在以字节编址的主存中以大端和小端方式存储的位置情况。

(1) struct{

double i;//0x1112131415161718

} s1;

(2) struct{

int i;//0x11121314

int j;//0x15161718

} s2;

(3) struct{

short i;//0x1112

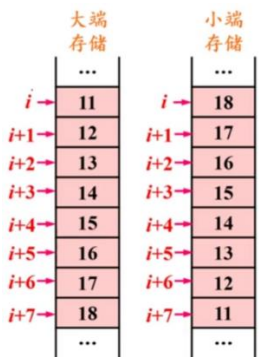
short j;//0x1314

short k;//0x1516

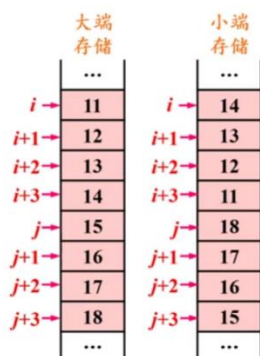
short l;//0x1718

} s3;

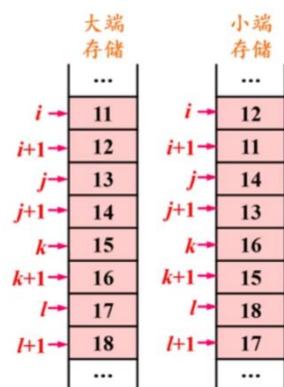
(1)



(2)



(3)



5.15 根据操作数所在位置，指出寻址方式：

- (1) 操作数在寄存器中，为（ ）寻址方式。
- (2) 操作数地址在寄存器中，为（ ）寻址方式。
- (3) 操作数在指令中，为（ ）寻址方式。
- (4) 操作数地址在指令中，为（ ）寻址方式。
- (5) 操作数的地址为某一寄存器内容与位移量之和，可以是（ ）寻址方式。

答：

- (1) 寄存器寻址
- (2) 寄存器间接寻址
- (3) 立即寻址
- (4) 直接寻址
- (5) 相对寻址

5.16 某单地址指令格式如图 5.23 所示：

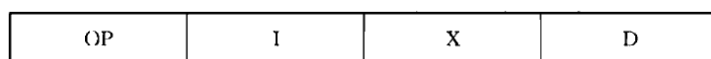


图 5.23 习题 5.16 附图

图 5.23 中，I 为间接特征，X 为寻址模式，D 为形式地址。I、X、D 组成该指令的操作数有效地址 EA。设 R 为变址寄存器，R1 为基址寄存器，PC 为程序计数器，具体寻址方式特征如表 5.11 所示，请在表中填入寻址方式名称。

表 5.11 习题 5.16 附表

I	X	EA	寻址方式
0	00	EA=D	
0	01	EA=(PC)+D	
0	10	EA=(R)+D	
0	11	EA=(R1)+D	
1	00	EA=(D)	
1	11	EA=((R1)+D)	

答：

直接寻址； 相对寻址；变址寻址；基址寻址；间接寻址；基址的间接寻址；

5.17 某计算机字长为 32 位，采用单字长单地址指令格式，共有 40 条指令。若采用直接、立即、变址、相对四种寻址方式获取操作数，试设计指令格式。该指令格式可直接寻址的地址范围是多大？可间接寻址地址范围又是多大？

答：2⁶=64>40,40 条指令需要 6 位进行编码，寻址方式 4 种需要 2 位进行编码，因此地址码为 24 位。

31	26	25	24	23	0
Opcode		X		D	

X=00 直接寻址，直接寻址的地址范围是 2²⁴=16MB

X=01 立即寻址，

X=10 变址寻址,有效地址 E = (IR) +D（可寻址 4GB 个存储单元）

X=11 相对寻址，有效地址 E = (PC) +D（可寻址 4GB 个存储单元）

注：IR 和 PC 的位数与机器的字长相等，均为 32 位

5.18 某采用 PC 相对寻址的转移指令存放于地址 0810H 处, 且占用 2 字节, 转移目标地址位于 07A0H 处。若该指令的地址字段是 10 位长度, 计算指令中提供的二进制偏移量(补码形式)。

答: PC 读出的内容 0810H, 一条指令占两个字节, 未跳转时下一条指令应该是 0812H, 要跳转的地址为 07A0H, 则位移字段 $07A0-0812H=-72H$, 对应的 10 位长度的二进制原码为 1001110010, 补码形式为: 1110001110

5.19 计算机指令系统中基本的指令类型有哪些? 它们的基本功能是什么? 移位指令有哪几种操作? 各自的操作特点是什么?

答:

指令类型分为:

(1) 数据传送类: 这是一种常用的指令, 用以实现寄存器与寄存器, 寄存器与存储单元以及存储器单元与存储器单元之间的数据传送, 对于存储器来说, 数据传送包括对数据的读(相当于取数指令)和写(相当于存数指令)操作。

(2) 算术运算类: 计算机指令系统一般都设有二进制数加\减\比较和求补等最基本的指令, 此外还设置了乘\除法运算指令\浮点运算指令以有十进制动算指令等。

(3) 逻辑运算类: 提供处理用户希望使用的其他类型数据的能力。一般计算机中都设有与、或、非、异或逻辑运算指令。有些计算机还设有位操作指令, 如位测试、位清除、位设置等。

(4) 系统控制类(通常是特权指令): 包括空操作、等待及某些特权指令。特权指令是指具有特殊权限的指令, 它们只供操作系统或其他系统软件使用, 一般不直接提供给用户使用。

(5) 移位类: 移位操作指令分为算术移位\逻辑移位和循环移位三种, 可以实现对操作数左移或右移一位或若干位。

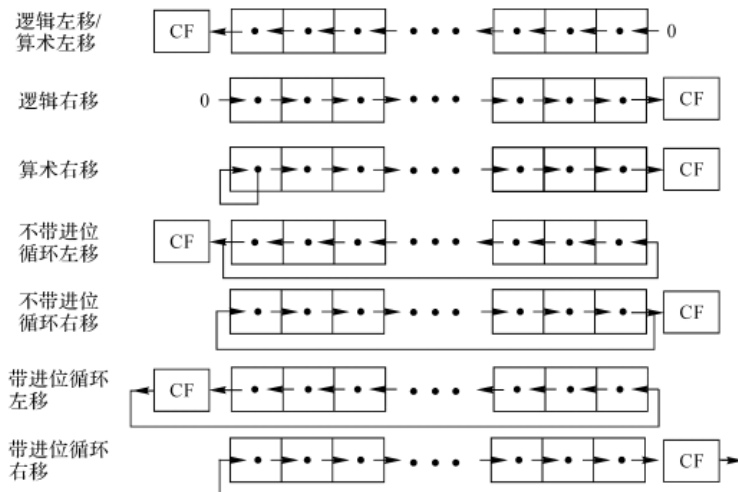
(6) 字符串操作类: 字符串处理指令就是一种非数值处理指令, 一般包括字符串传送, 字符串转换(把一种编码的字符串转换成另一种编码的字符串), 字符串比较, 字符串查找(查找字符串中某一子串), 字符串匹配, 字符串的抽取(提取某一子串)和替换(把某一字符串用另一字符串替换)等。

(7) 数据转换类: 完成主机与外围设备之间的信息传送, 包括输入/输出数据, 主机向外设发送控制命令或读取外设的工作状态等。

(8) 输入/输出类: 完成主机与外围设备之间的信息传送, 包括输入/输出数据, 主机向外设发送控制命令或读取外设的工作状态等。

(9) 程序控制类：用来控制程序的走向,具有改变指令执行顺序的功能。卜

移位类指令：在计算机中,常见的移位运算包括逻辑移位、算术移位、不带进位循环移位和带进位 循环移位



5.20 MOV operand1,operand2 为传送指令, operand1 为目的操作数, operand2 为源操作数,指令功能为将源操作数传送到目的操作数。现有基址寄存器(RB)=0100H, 变址寄存器(RI)=0002H, **R1 为 16 位寄存器**, () 表示寄存器或存储单元中的内容, 主存中存储的信息如图 5.24 所示 (采用小数端存储)。

试说明下列指令执行后 R1 中的值。

- (1) MOV R1, #1200H
- (2) MOV R1, RB
- (3) MOV R1, (1200H)
- (4) MOV R1, (RB)
- (5) MOV R1, 1100H(RB)
- (6) MOV R1, (RB)(RI)
- (7) MOV R1, 1100H(RB)(RI); 源操作数 EA=(RB)+(RI)+1100H

主存地址	内容
⋮	
0100H	12H
0101H	34H
0102H	56H
0103H	78H
⋮	
1200H	2AH
1201H	4CH
1202H	B7H
1203H	65H
⋮	

图 5.24 习题 5.20 附图

答: (1) **MOV R1, #1200H**

源操作数#1200H 是立即寻址, 存的是操作数本身。目的操作数 R1 是寄存器寻址, 因此, R1=1200H

(2) **MOV R1, RB**

把基址寄存器 RB 的内容传送给 R1 寄存器, 所以两个操作数都是寄存器寻址

R1=0100H

(3) **MOV R1, (1200H)**

把主存地址 1200H 的内容传送到 R1 寄存器中。注意是目的寄存器是 16 位的数据, 小端存储。

R1= 4C2AH

(4) **MOV R1, (RB)**

R1= 3412H

(5) **MOV R1, 1100H(RB)**

寄存器相对寻址, 基址寄存器 RB 中的内容和偏移量 1100H 相加, 即 0100+1100=1200H, 因此, 从主存的 1200H 地址开始取 16 位数据存放到 R1 寄存器中。

R1= 4C2AH

(6) **MOV R1, (RB)(RI)**

基址变址寻址方式, 基址寄存器 RB 中的内容和变址寄存器 RI 中的内容相加, 即 0100+0002=0102H, 相加的结果地址作为取数据的内存地址。

R1= 7856H

(7) MOV R1,1100H(RB)(RI); 源操作数 EA=(RB)+(RI)+1100H

基址变址相对寻址方式,源操作数 EA=(RB)+(RI)+1100H,即 0100+0002+1100=1202H,

R1=65B7H

5.21 写出实现如下要求的指令(指令操作码可用英文单词或缩写表示,R1 为 16 位寄存器)

- (1) 将寄存器 R1 低 4 位清零;
- (2) 将寄存器 R1 低 4 位置 1;
- (3) 将寄存器 R1 低 4 位求反;
- (4) 将寄存器 R1 低 4 位移动到高 4 位;

答:

(1) AND R1, FFF0H

(2) OR R1, 000FH

(3) XOR R1, 000FH

(4) MOV CL,4

ROL AL,CL

5.22 简述 CISC 和 RISC 的特点。

见 5.5 节,指令数量、寻址方式、指令长度、指令执行周期、访问存储指令、寄存器数量、控制器结构等区别重点关注。

第 6 章 中央处理器

6.1 中央处理器 (CPU)由哪些部分组成？它们应具有哪些基本功能？

答：CPU 是由控制器、运算器、寄存器组组成，CPU 的主要功能是执行存储在主存中的指令序列，也即执行程序。

6.2 控制器的功能是什么？它由哪些部件组成？

答：控制器的作用是为 CPU 内外的所有部件提供指令执行时所需的控制信号。

6.3 什么是硬布线控制器？有什么特点？

答：

硬布线控制器是将控制部件做成产生专门固定时序控制信号的逻辑电路，产生各种控制信号，因而又称为组合逻辑控制器。这种逻辑电路以使用最少元件和取得最高操作速度为设计目标，因为该逻辑电路由门电路和触发器构成的复杂树形网络，所以称为硬布线控制器。

硬布线控制器的特点：硬布线控制器的优点是控制器的速度取决于电路延迟所以速度快，缺点是由于将控制部件看作专门产生固定时序控制信号的逻辑电路，所以把用最少元件和取得最高速度作为目标，一旦设计完成，不可能通过其他额外修改添加新功能。

6.4 硬布线控制器如何产生微命令？产生微命令的主要条件有哪些？

答：根据每个微命令的逻辑表达式，使用与或逻辑电路来实现微命令的生成。

条件包括了：节拍周期信息、指令信息、状态条件

6.5 什么是指令周期和微指令周期？二者之间存在什么关系？

答：

指令周期是执行一条指令所用的时间。或者定义为：在执行一条指令的过程中，由 CPU 完成的一组操作构成一个指令周期。

微指令周期：一条微指令执行的时间（包括从控制存储器中取得微指令和执行微指令所用时间）。

一条指令周期中，包含了多个微指令周期。

6.6 微程序控制的基本思想是什么？微程序控制器的特点是什么？

答：微程序控制的基本思想是把机器指令的每一操作控制步编成一条微指令。每条机器指令对应一段微程序。执行机器指令时，从控制存储器中顺序取出这些微指令，就可按所要求的次序产生相应的操作控制信号。

微程序控制器的特点：优点是同组合逻辑控制器相比，微程序控制器具有规整性、灵活性、可维护性等优点；缺点是由于微程序控制器采用了存储程序原理所以，每条指令都要从控制存储器中取一次，影响了速度。

6.7 微程序控制器设计中微指令由哪两部分组成？各部分的作用是什么？

答：

微指令可分为微命令字段（或微操作控制字段）和微地址字段（或顺序控制字段）

微命令字段：提供一步操作所需的微命令。

微地址字段：指明后续微地址的形成方式，提供微地址的给定部分。

6.8 微指令产生控制信号（微命令）的方式有哪些？各方式是如何产生控制信号的？

答：水平型微指令(Horizontal Microinstruction)和垂直型微指令(Vertical Microinstruction)。

水平型微指令

一次能定义并执行多个并行操作微命令的微指令，叫做水平型微指令。水平型微指令的一般格式如下：控制字段，判别测试字段和下地址字段。按照控制字段的编码方法不同，水平型微指令又分为三种：一种是全水平型（不译法）微指令，第二种是字段译码法水平型微指令，第三种是直接和译码相混合的水平型微指令。

垂直型微指令

微指令中设置微操作码字段，采用微操作码编译法，由微操作码规定微指令的功能，称为垂直型微指令。垂直型微指令的结构类似于机器指令的结构。它有操作码，在一条微指令中只有 1-2 个微操作命令，每条微指令的功能简单，因此，实现一条机器指令的微程序要比水平型微指令编写的微程序长得多。它是采用较长的微程序结构去换取较短的微指令结构。

6.9 对微指令进行字段直接编码的基本原则是什么？

答：子字段划分的主要原则

- ①把互斥的微命令划分在同一字段内，如存储器的读和写。相容的微命令划分在不同字段内，如 ALU 和存储器之间的微命令。
- ②一般每个子字段应留出一个编码状态，表示本字段不发任何微命令。
- ③每个子字段所定义微命令数不宜太多，否则将使微命令译码复杂。

6.10 主存与控存有什么异同？

答：控制存储器中存放微程序。与主存的区别：

- ①控制存储器在 CPU 中、而主存不是；
- ②控制存储器是一个 ROM，而主存是 ROM 和 RAM
- ③控制存储器容量比主存小
- ④控制存储器字长比主存长
- ⑤控制存储器速度比主存快

6.11 与硬布线控制器相比，微程序控制器有哪些优缺点？

答 微程序控制器的特点：优点是同组合逻辑控制器相比，微程序控制器具有规整性、灵活性、可维护性等优点；缺点是由于微程序控制器采用了存储程序原理所以，每条指令都要从控制存储器中取一次，影响了速度。

6.12 在微程序控制器设计中，若控制器需要处理 100 条指令，每条指令需要最多 8 条微指令构成的微程序来实现其功能；控制器需要产生 100 个微命令（控制信号），且被分为 6 个相容类，如表 6.14 所示。

表 6.14 习题 6.12 附表

类型	0	1	2	3	4	5
微命令数	14	29	8	23	11	15

- (1) 请合理设计微指令的控制域（微命令生成字段），使其具有尽可能短的长度。
- (2) 若微指令的次地址（即下一条微指令地址）仅是顺序地址和无条件跳转地址，请合理设计微指令的地址域（下一条微指令地址生成字段）。
- (3) 请根据以上设计确定，微指令分布采用单地址格式和可变地址格式时，控制存储器

容量多大为宜。

答:

(1) 14+1:4 位; 29+1:5 位; 8+1:4 位; 23+1:5 位; 11+1:4 位; 15+1:4 位

因此控制域的长度为: $4+5+4+5+4+4=26$ 位

(2) 地址域:

用 1 位表示: $AC=0$, 下一条顺序微指令地址由 μPC 提供; $AC=1$, 则根据指令操作码和寻址信息跳转到微程序首地址。

(3) $100 \times 8 = 800$ 条微指令, 需要 10 位地址编码。

单地址微指令的长度: $26+1+10=37$ 位, 需要存储微指令的容量为 $800 \times 37b$

双地址微指令的长度: $26+1+20=47$ 位, 需要存储微指令的容量为 $800 \times 47b$

6.13 某微程序控制器采用的微指令字长为 24 位。微命令生成部分由 4 个字段构成, 各字段所包括的互斥微命令分别为 5 个、8 个、14 个和 3 个。控制产生次地址的条件有 3 种。试说明该微控制器最多可用几位来表示次地址, 控制存储器的容量为多少?

答:

互斥微命令所需的最少位数分别是 3 位, 4 位, 4 位, 2 位, 根据二进制表示的最大值, 其次, 控制产生次地址的条件有 3 种, 则至少需要 2 位, 则最多的空余地址位为 $24-3-4-4-2-2=9$ 位, 控制存储器的容量为 $2^9=512$ 位

6.14 某计算机系统简化的 CPU 结构如图 6.3 所示, 实现指令 ADD R0, R1 (功能为 $(R_0) + (R_1) \rightarrow R_0$) 时采用以下微操作流程:

取指令:

$AR \leftarrow PC$

$AB \leftarrow AR$

$DB \leftarrow \text{Memory}[AR], PC \leftarrow PC+2$

$DR \leftarrow DB$

$IR \leftarrow DR$

执行指令:

$Y \leftarrow R1$

$Z \leftarrow R0 + Y$

$R0 \leftarrow Z$

(1) 依据此流程，写出实现加法指令“ADD R0, (R1)”的**微流程**。说明:该指令中(R1)为寄存器间接寻址，指令功能为 $(R0) + ((R1)) \rightarrow R0$ ，微操作 $DB \leftarrow \text{Memory}[AB]$ 表示存储器做读操作。

(2) 试拟出加法指令“ADD R1,B(R2)”的微流程。其中 B(R2)表示基址寻址，B 是偏移量，R2 是基址寄存器。

(3) 拟出减法指令“SUB R1,100(PC)”的指令微流程。

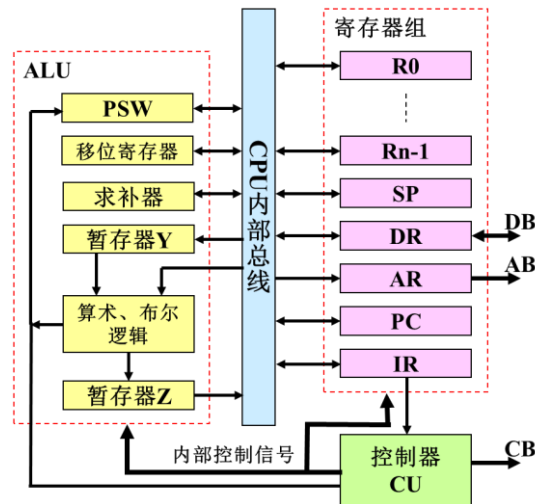


图6.3 单总线数据通路CPU内部结构图

答：

(1) 加法指令“ADD R0, (R1)”的微流程

取指令：

$AR \leftarrow PC$

$AB \leftarrow AR$

$DB \leftarrow \text{Memory}[AR], PC \leftarrow PC + 2$

$DR \leftarrow DB$

$IR \leftarrow DR$

执行指令：

$AR \leftarrow R1$

$AB \leftarrow AR$

$DB \leftarrow \text{Memory}[AR]$

$DR \leftarrow DB$

$Y \leftarrow DR$

$Z \leftarrow R0 + Y$

$R0 \leftarrow Z$

(2) 加法指令“ADD R1,B(R2)”的微流程:

取指令:

$AR \leftarrow PC$

$AB \leftarrow AR$

$DB \leftarrow \text{Memory}[AR], PC \leftarrow PC + 2$

$DR \leftarrow DB$

$IR \leftarrow DR$

执行指令:

$AR \leftarrow R2$

$AB \leftarrow AR$

$DB \leftarrow \text{Memory}[AR]$

$DR \leftarrow DB$

$Y \leftarrow DR$

$Z \leftarrow B + Y$

$Y \leftarrow Z$

$Z \leftarrow R1 + Y$

$R1 \leftarrow Z$

(3) 减法指令“SUB R1,100(PC)”的指令微流程

取指令:

$AR \leftarrow PC$

$AB \leftarrow AR$

$DB \leftarrow \text{Memory}[AR]$

$DR \leftarrow DB$

$IR \leftarrow DR$

执行指令:

$AR \leftarrow PC$

$AB \leftarrow AR$

$DB \leftarrow \text{Memory}[AR], PC \leftarrow PC+2$

$DR \leftarrow DB$

$Y \leftarrow DR$

$Z \leftarrow 100+Y$

$Y \leftarrow Z$

$Z \leftarrow R1+Y$

$R1 \leftarrow Z$

6.15 图 6.30 为某一运算器的简化框图，其中 A、B 具有寄存器和多路选择器的功能，M 具有移位和多路选择器的功能，ALU 为算术逻辑单元，R0 和 R1 为通用寄存器。图中带箭头的线段为数据通路，是一种单总线结构。带下标 in 者为数据存入该寄存器的微命令，带下标 out 者为将该寄存器的内容输出的微命令，XX→ALU、ADD、+1、V、L 等均为微命令。

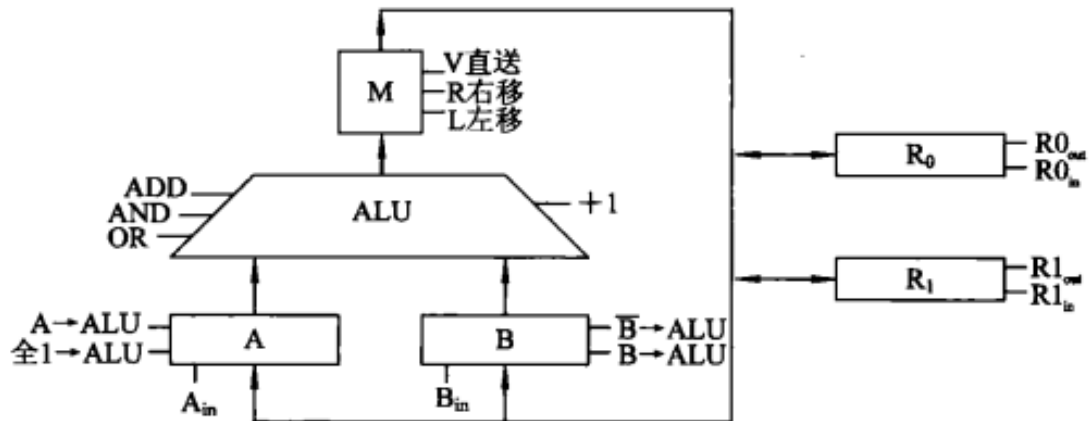


图 6.30 题 6.15 附图

(1) 根据图中所示，选择括号中的正确答案。

- ① $R0_{out}$, A_{in} (相容/互斥)
- ② V, R, L (相容/互斥)
- ③ $R0_{out}$, $R1_{out}$ (相容/互斥)
- ④ $ALU \leftarrow A$, ADD, V (相容/互斥)
- ⑤ ADD, AND (相容/互斥)

(2) 若 $2(R0-1) \rightarrow R0$ 操作的微命令序列如下所示:

①R0out, Bin ; R0→B

②ALU←B, ALU←全 1, ADD, L, R0in ; (B-1)×2→R0

试写出执行 R0VR1→R0 操作所需的微命令序列(v为或运算)。

答:

(1) 根据图中所示, 选择括号中的正确答案。

①R0out, Ain (相容)

②V, R, L (互斥)

③R0out, R1out (互斥)

④ALU←A, ADD, V (相容)

⑤ADD, AND (互斥)

(2) R0VR1→R0 操作所需的微命令序列:

① R0out, Ain; R0→A

② R1out, Bin; R1→B

③ ALU←A, ALU←B, OR, V, R0in; (A OR B → R0)

6.16 某假想主机的主要部件如图 6.31 所示。其中 R0~R3 为通用寄存器, A、B 为暂存器, 其他部件如图中所标识。

(1)在图 6.31 的基础上, 画出其内部的数据通路并标明数据流向。

(2)写出传送指令“MOV R0, R1” (即将 R1 的内容传送到 R0)的指令微操作流程。

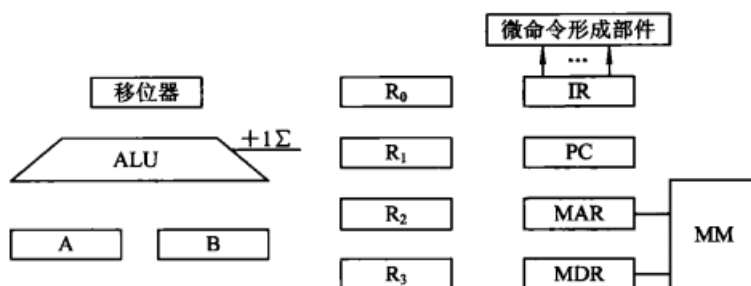


图 6.31 习题 6.16 附图

答:

(2)

① R1→MDR

② MDR→R0

6.17 某 CPU 结构如图 6.32 中虚框内所示，其中包括一个累加寄存器 AC、一个状态寄存器和其他四个寄存器，各部分之间的连线表示数据通路，箭头表示信息传送方向。

- (1) 图中的四个寄存器 A、B、C、D 分别是什么功能的寄存器？
- (2) 写出 LDA X 指令执行阶段的微操作流程(X 为内存地址，LDA X 功能为 $(X) \rightarrow AC$)。
- (3) 写出 STA Y 指令执行阶段的微操作流程(Y 为内存地址，STA Y 功能为 $(AC) \rightarrow Y$)。

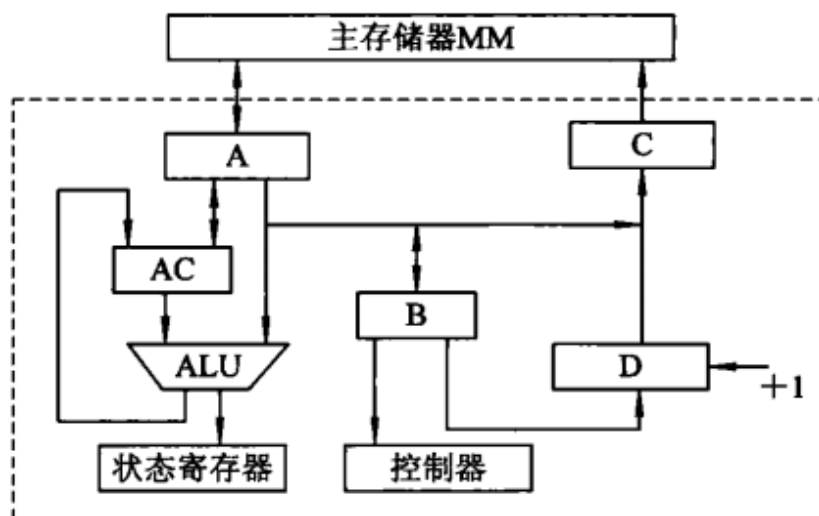
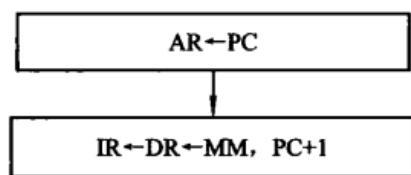


图 6.32 习题 6.17 附图

提示:取指令阶段的微操作流程可描述为



答:

- (1) A: 数据缓冲寄存器 DR; B: 指令寄存器 IR; C: 主存地址寄存器 AR; D: 程序计数器 PC
- (2)

6.18 已知某假想机的运算器逻辑如图 6.33 所示，它的核心部分是一个并行加法器，S 为加法器的输出，加法器的输入为寄存器 A 和 B，最低进位信号为 “+1” (“+1”为高电平时，进

位为 1)。图中带有下标 in 的控制信号分别为寄存器(A, B, C,D)的信息输入锁存脉冲信号，其他为节拍控制电平信号，节拍控制电平与输入锁存脉冲的相位关系如图中右下方所示。写出实现运算 $(D) - 1 \rightarrow D$ 的微命令序列。

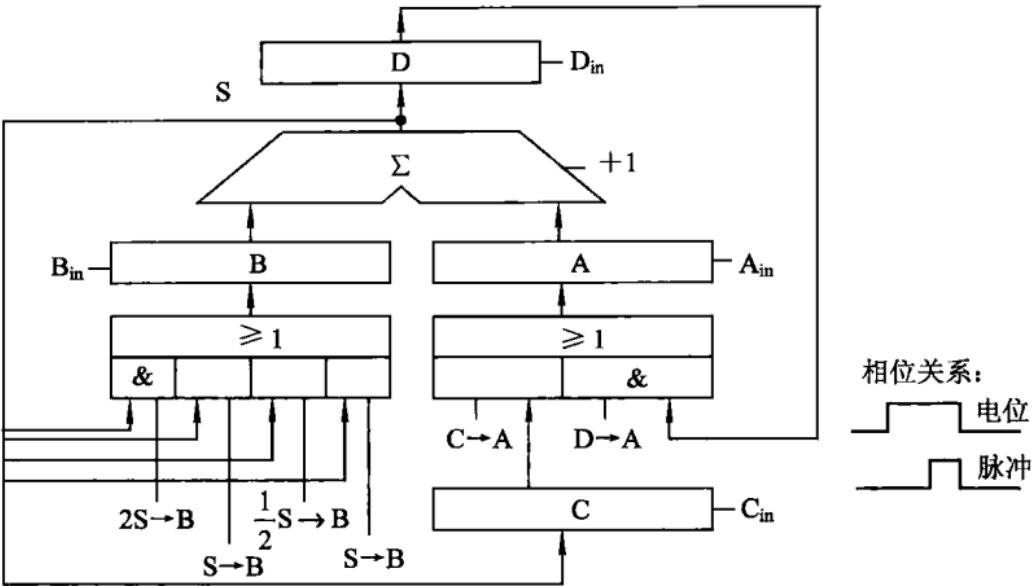


图 6.33 习题 6.18 附图

6.19 在计算机 A 和 B 上运行两个程序，得到测试数据如下：

程序	计算机 A			计算机 B		
	执行时间/s	指令数/条	成本	执行时间/s	指令数/条	成本
1	2.0	5×10^9	\$ 500	1.5	6×10^9	\$ 800
2	5.0			10.0		

- 对于程序 1 和 2，哪个计算机更快？
- 当运行程序 1 时，计算机 A 和 B 的指令执行速率(即每秒指令数)分别为多少？
- 如果需要很多次地运行程序 1，你愿意大量购买哪种计算机?为什么？

答：

第 7 章 流水线技术

6.20 某 500MHz 计算机，执行基准测试程序，程序中的指令类型、数量及指令执行的平均周期数如下表所示：

指令类型	指令数量/条	指令执行周期数
整数	50 000	1
数据传送	80 000	2
浮点	10 000	4
控制传送	5000	2

求该计算机的有效 CPI、MIPS 及程序的执行时间。

(1) $CPI = (50000 \times 1 + 80000 \times 2 + 10000 \times 4 + 5000 \times 2) / (50000 + 80000 + 10000 + 5000) = 1.79$

(2) 执行时间 $= (50000 \times 1 + 80000 \times 2 + 10000 \times 4 + 5000 \times 2) / (500 \times 10^6) = 520 \mu s$

(3) $MIPS = (50000 + 80000 + 10000 + 5000) / 520 = 278.8$

6.21 某计算机系统具有 L1 Cache 和 L2Cache，在 Cache 完全命中的情况下，系统的 CPI 为

1.2。假设每条指令有平均 1.1 次的主存访问，两级 Cache 的命中情况见下表：

Cache 级别	命中率	未命中的时间损失
L1	95 %	8 个时钟周期
L2	80 %	60 个时钟周期

(1) 在 Cache 未命中的情况下，系统的有效 CPI 是多少？

(2) 如果将 L1 Cache 和 L2 Cache 看作单一的 Cache，那么整体的有效命中率和未命中时间损失是多少？

6.22 假设条件分支指令的两种不同设计方法如下：

CPU1: 通过比较指令设置条件码，然后测试条件码进行分支；

CPU2: 在分支指令中包括比较过程。

在两种 CPU 中，条件分支指令都占用 2 个时钟周期，而其他指令占用 1 个时钟周期。

现有一段程序分别在 CPU1 和 CPU2 上运行，程序中分支指令占 20%。

(1) 假设 CPU1 的时钟频率比 CPU2 快 1.25 倍，程序在哪一个 CPU 上运行得更快？

(2) 如果 CPU1 的时钟频率比 CPU2 快 1.1 倍，程序在哪一个 CPU 上运行得更快？

(1) CPU1 的分支使用 2 条指令完成。

$$\frac{\text{CPU1 执行时间}}{\text{CPU2 执行时间}} = \frac{0.8*1+0.2*1+0.2*2}{0.8*1+0.2*2} \times \frac{1}{1.25} = 0.93, \text{ CPU1 快。}$$

(2) $\frac{\text{CPU1 执行时间}}{\text{CPU2 执行时间}} = \frac{0.8*1+0.2*1+0.2*1}{0.8*1+0.2*2} \times \frac{1}{1.1} = 1.06, \text{ CPU2 快。}$

6.23 已知 4 个程序在三台计算机上的执行时间如下:

程序	执行时间/s		
	计算机 A	计算机 B	计算机 C
程序 1	1	10	20
程序 2	1000	100	20
程序 3	500	1000	50
程序 4	100	800	100

假设 4 个程序中每一个都有 100 000 000 条指令要执行,

(1) 计算这三台计算机中每台机器上每个程序的 MIPS 速率。根据这些速率值, 你能否得出有关三台计算机相对性能的明确结论?

(2) 给出一种统计的方法(比如求均值)来估计三台计算机的相对性能, 说明理由。

7.2 一个浮点流水线有 5 级, 其延迟分别为 110ns、90ns、120ns、80ns、100ns。该流水线的最大吞吐量是多少 MFLOPS?

解: $TP_{\max} = 1/(120\text{ns}) = 8.33\text{MFlops}$

7.7 某 CPU 内有 5 级指令流水线, 每级的处理时间(包括级间流水线寄存器延迟)为 10ns、5ns、5ns、10ns、5ns。

(1) 当执行 1000 条指令时, 该流水线的吞吐率和加速比为多少?

(2) 若要改进该流水线的性能, 可对流水线做何改造? 改造后的流水线吞吐率可达到多少?

解: (1) 流水执行时间 $T_n(m) = (10+5+5+10+5) + (1000-1)*10 = 10025\text{ns}$

$TP = 1000/10025\text{ns} = 99.75\text{MIPS}$

非流水时间 $T_n(1) = (10+5+5+10+5)*1000 = 35000\text{ns}$

加速比 $S = 35000/10025 = 3.49$

(2) 改为 7 级流水, 每级延迟为 5ns。

最大吞吐率为 200MIPS。

7.12 一台非流水线处理机 X，其时钟速率为 25MHz 且平均 CPI 为 4。对 X 进行改进后的处理机是 Y，它被设计成 5 段的线性指令流水线，但是由于流水线寄存器延迟及及时钟偏移等影响，Y 的时钟速率仅为 20MHz。

(1) 如果有一个 100 条指令的程序在这两台处理机上执行，那么处理机 Y 与处理机 X 相比较的加速比是多少？

(2) 当这一程序在两台机器上执行时，试计算每台处理机的 MIPS 速率。

解：(1) 非流水时间 $T_n(1)=100*4/25\text{Mhz}=16\mu\text{s}$

流水时间 $T_n(m)=(5+100-1)*(1/20\text{MHz})=5.2\mu\text{s}$

加速比 $S=16/5.2=3.08$

(2) 非流水处理机为 $100/16\mu\text{s}=6.25\text{MIPS}$

流水处理机为 $100/5.2=19.23\text{MIPS}$

7.14 图 7.37 是某 4 级指令流水线的时-空图，求该流水线在不断流时的实际吞吐率(假设 $n=100$)。

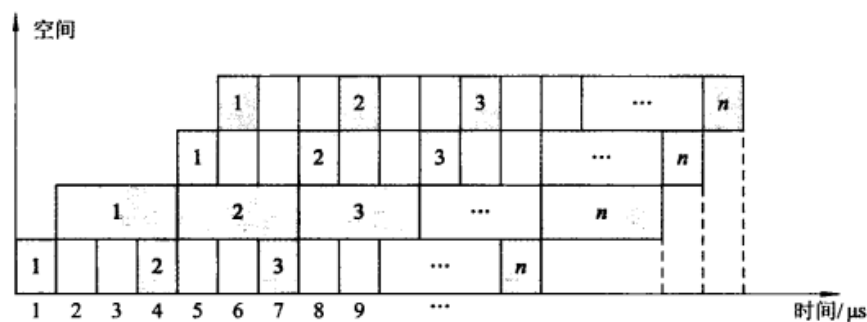


图 3.37 习题 7.14 附图

解：完成 100 条指令所用时间为 $6+(100-1)*3=303\mu\text{s}$

$TP=100/303\mu\text{s}=0.33\text{MIPS}$