

习题解答

1.1 第 1 章习题

1. 三进制和五进制的基数分别是多少？钟表的时、分、秒分别采用什么进制？

答：三进制基数是 3, 五进制的基数是 5。钟表的时、分、秒采用的进制分别是 24、60 和 60(一天为 24 小时, 钟表的“时”应为二十四进制, 不能错把它当作十二进制)。

2. 完成下列二进制数计算。

(1) $1100.101 + 10.1$

(2) $1111.001 - 1100.101$

(3) 101.001×1.011

(4) $111.000011 \div 1.011$

解: (1)

$$\begin{array}{r} 1100.101 \\ + 10.1 \\ \hline 1111.001 \end{array}$$

(2)

$$\begin{array}{r} 1111.001 \\ - 1100.101 \\ \hline 10.100 \end{array}$$

(3)

$$\begin{array}{r} 101.001 \\ \times 1.011 \\ \hline 101001 \\ 101001 \\ + 101001 \\ \hline 111.000011 \end{array}$$

(4) 方法 1(直接计算):

$$\begin{array}{r} 101.001 \\ 1.011 \overline{) 111.000011} \\ \underline{- 101.1} \\ 1.100 \\ \underline{- 1.011} \\ 0.001011 \\ \underline{- 0.001011} \\ 0.000000 \end{array}$$

方法 2(将除数化为整数后计算):

$$\begin{array}{r} 101.001 \\ 1011. \overline{) 111000.011} \\ \underline{- 101100.} \\ 1100. \\ \underline{- 1011.} \\ 1.011 \\ \underline{- 1.011} \\ 0.000 \end{array}$$

3. 将十进制数 153.375 分别转换成二进制数、八进制数和十六进制数。

解: (1) 先利用“除二取余法”将 153 转换成二进制整数, 再利用“乘二取整法”将 0.375 转换成二进制小数。

由以下方法可得: $153 = 10011001\text{B}$ 。

商	余
2 153	1 (低位)
2 76	0
2 38	0
2 19	1
2 9	1
2 4	0
2 2	0
2 1	1 (高位)
0	

由以下方法可得: $0.375 = 0.011\text{B}$ 。

整数	×	0.375
(高位) 0	×	0.750
	×	2
1	×	1.500
	×	2
(低位) 1	×	1.000

所以, $153.375 = 10011001.011\text{B}$ 。

(2) 将二进制数 10011001.011 转换成八进制数。

010, 011, 001. 011 B (每 3 位二进制数分为一组)

2 3 1 . 3 Q (用一个八进制数字替代 3 位二进制数)

(3) 将二进制数 10011001.011 转换成十六进制数。

1001, 1001. 0110 B (每 4 位二进制数分为一组)

9 9 . 6 H (用一个十六进制数字替代 4 位二进制数)

所以, $153.375 = 10011001.011\text{B} = 231.3\text{Q} = 99.6\text{H}$ 。

4. 完成下列进制数的转换。

(1) $(10110111.1001)_2 = ()_8 = ()_{16}$ (2) $(4\text{C.E})_{16} = ()_8 = ()_2$

解: (1) $(10110111.1001)_2 = (267.44)_8 = (\text{B}7.9)_{16}$

(2) $(4\text{C.E})_{16} = (114.7)_8 = (1001100.111)_2$

5. 计算机是由哪几个基本部分组成的? 什么是总线? 什么是 CPU? CPU 是由哪两个部分组成的?

答: 计算机是由存储器、运算器、控制器、输入和输出设备五大基本部分组成的。总线是计算机各部件之间的公共信息传输线。CPU 是中央处理器 (Central Processing Unit) 的英文缩写, 它是由运算器和控制器两大部分组成的。

6. 什么是单元地址? 什么是存储容量? 存储容量的基本度量单位是什么? 它能存

放几位二进制数码？

答：存储单元的编号称为单元地址。存储器所能存放的字节总数称为存储容量。存储容量的基本计量单位是字节，一个字节可容纳 8 位二进制数。

1.2 第 2 章习题

1. 假设图 1.1 中开关闭合时为“1”，断开时为“0”；灯泡亮时为“1”，灭时为“0”。试写出图中灯泡与开关之间的逻辑关系。

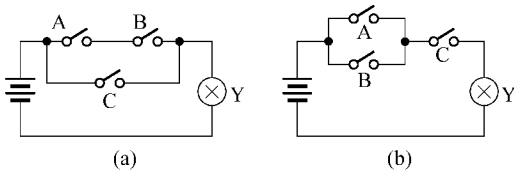


图 1.1

解：由教材例 2.1 和例 2.2 可知：两个开关串联符合逻辑“与”关系，开关并联符合逻辑“或”关系，因此

(a) $Y = A \cdot B + C$ (b) $Y = (A + B) \cdot C$

2. 试分析如图 1.2 所示逻辑门电路的工作原理，指出其输入、输出信号之间的逻辑运算关系。

解：(1) 由图 1.2(a)和表 1.1 可知：只要 A、B 输入任何一个为低电平（逻辑“0”），即：只要 V_{P1} 、 V_{P2} 管有一个导通， V_{N1} 、 V_{N2} 管有一个截止，电路输出 Y 就会变为高电平（逻辑“1”）。可见，该电路符合逻辑“与非”的运算规则。

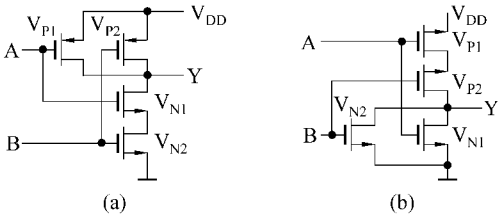


图 1.2

表 1.1 逻辑“与非”门晶体管状态

A	V_{P1}	V_{N1}	B	V_{P2}	V_{N2}	V_{DD} 、Y 间 电阻值	Y、GND 间 电阻值	Y
0	通	断	0	通	断	0	∞	1
0	通	断	1	断	通	0	∞	1
1	断	通	0	通	断	0	∞	1
1	断	通	1	断	通	∞	0	0

(2) 由图 1.2(b)和表 1.2 可知：只要 A、B 输入任何一个为高电平（逻辑“1”），即：只

要 V_{P1} 、 V_{P2} 管有一个截止, V_{N1} 、 V_{N2} 管有一个导通, 电路输出 Y 就会变为低电平(逻辑“0”)。因此, 该电路符合逻辑“或非”的运算规则。

表 1.2 逻辑“或非”门晶体管状态

A	V_{P1}	V_{N1}	B	V_{P2}	V_{N2}	V_{DD} 、Y 间 电阻值	Y、GND 间 电阻值	Y
0	通	断	0	通	断	0	∞	1
0	通	断	1	断	通	∞	0	0
1	断	通	0	通	断	∞	0	0
1	断	通	1	断	通	∞	0	0

3. 证明下列等式。

$$(1) (A+B) \cdot (\bar{A}+B) = B$$

$$(2) A \cdot B + A \cdot \bar{B} + \bar{A} \cdot \bar{B} = A + \bar{B}$$

$$(3) A \cdot B + \bar{A} \cdot C + \bar{B} \cdot C = A \cdot B + C$$

$$(4) A \cdot \bar{B} + B \cdot D + \bar{A} \cdot D + C \cdot D = A \cdot \bar{B} + D$$

$$(5) A \cdot B + C + \bar{C} \cdot (\bar{A} + \bar{B}) \cdot (A + C \cdot D) = A + C$$

证明: (1) 左 = $(A+B) \cdot (\bar{A}+B)$

$$= A \cdot \bar{A} + B$$

(根据加法分配律)

$$= 0 + B$$

(根据互补律)

$$= B = \text{右}$$

(根据 01 律)

$$(2) \text{左} = A \cdot B + A \cdot \bar{B} + \bar{A} \cdot \bar{B}$$

$$= A \cdot (B + \bar{B}) + \bar{A} \cdot \bar{B}$$

(根据乘法分配律)

$$= A \cdot 1 + \bar{A} \cdot \bar{B}$$

(根据互补律)

$$= A + \bar{B} = \text{右}$$

(根据 01 律、吸收律)

$$(3) \text{左} = A \cdot B + \bar{A} \cdot C + \bar{B} \cdot C$$

$$= A \cdot B + (\bar{A} + \bar{B}) \cdot C$$

(根据乘法分配律)

$$= A \cdot B + (\overline{A \cdot B}) \cdot C$$

(根据反演律)

$$= A \cdot B + C = \text{右}$$

(根据吸收律)

$$(4) \text{左} = A \cdot \bar{B} + B \cdot D + \bar{A} \cdot D + C \cdot D$$

$$= A \cdot \bar{B} + (B + \bar{A}) \cdot D + C \cdot D$$

(根据乘法分配律)

$$= A \cdot \bar{B} + (\overline{A \cdot B}) \cdot D + C \cdot D$$

(根据交换律、反演律)

$$= A \cdot \bar{B} + D + C \cdot D$$

(根据吸收律)

$$= A \cdot \bar{B} + (1 + C) \cdot D$$

(根据乘法分配律)

$$= A \cdot \bar{B} + D = \text{右}$$

(根据 01 律)

$$(5) \text{左} = A \cdot B + C + \bar{C} \cdot (\bar{A} + \bar{B}) \cdot (A + C \cdot D)$$

$$= A \cdot B + C + (\bar{A} + \bar{B}) \cdot (A + C \cdot D)$$

(根据吸收律)

$$= A \cdot B + (\overline{A \cdot B}) \cdot (A + C \cdot D) + C$$

(根据交换律、反演律)

$$= A \cdot B + (A + C \cdot D) + C$$

(根据吸收律)

$$= A \cdot (B+1) + C \cdot (D+1)$$

(根据结合律、乘法分配律)

$$= A + C = \text{右}$$

(根据 01 律)

4. 化简下列逻辑表达式。

$$(1) Y = A \cdot B \cdot (A+B \cdot C)$$

$$(2) Y = \overline{A \cdot B \cdot C \cdot (B+\overline{C})}$$

$$(3) Y = \overline{A} \cdot \overline{B} + A \cdot B + A \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C$$

$$(4) Y = A + A \cdot B \cdot C + A \cdot \overline{B} \cdot \overline{C} + B \cdot C + \overline{B} \cdot C$$

$$(5) Y = A \cdot \overline{B} + \overline{A} \cdot C \cdot D + B + \overline{C} + \overline{D}$$

解: (1) $Y = A \cdot B \cdot (A+B \cdot C)$

$$= A \cdot B \cdot A + A \cdot B \cdot B \cdot C$$

(根据乘法分配律)

$$= A \cdot B + A \cdot B \cdot C$$

(根据重叠律)

$$= A \cdot B \cdot (1+C)$$

(根据乘法分配律)

$$= A \cdot B$$

(根据 01 律)

$$(2) Y = \overline{A \cdot B \cdot C \cdot (B+\overline{C})}$$

$$= (A+\overline{B}+\overline{C}) \cdot (B+\overline{C})$$

(根据反演律、还原律)

$$= (A+\overline{B}) \cdot B + \overline{C}$$

(根据加法分配律)

$$= A \cdot B + \overline{B} \cdot B + \overline{C}$$

(根据乘法分配律)

$$= A \cdot B + 0 + \overline{C}$$

(根据互补律)

$$= A \cdot B + \overline{C}$$

(根据 01 律)

$$(3) Y = \overline{A} \cdot \overline{B} + A \cdot B + A \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C$$

$$= \overline{A} \cdot \overline{B} + \overline{A} \cdot B \cdot C + A \cdot B + A \cdot \overline{B} \cdot C$$

(根据交换律)

$$= \overline{A} \cdot (\overline{B} + B \cdot C) + A \cdot (B + \overline{B} \cdot C)$$

(根据乘法分配律)

$$= \overline{A} \cdot (\overline{B} + C) + A \cdot (B + C)$$

(根据吸收律)

$$= \overline{A} \cdot \overline{B} + A \cdot B + \overline{A} \cdot C + A \cdot C$$

(根据乘法分配律、交换律)

$$= \overline{A} \cdot \overline{B} + A \cdot B + (\overline{A} + A) \cdot C$$

(根据乘法分配律)

$$= \overline{A} \cdot \overline{B} + A \cdot B + C$$

(根据互补律、01 律)

$$(4) Y = A + A \cdot B \cdot C + A \cdot \overline{B} \cdot \overline{C} + B \cdot C + \overline{B} \cdot C$$

$$= A \cdot (1 + B \cdot C + \overline{B} \cdot \overline{C}) + (B + \overline{B}) \cdot C$$

(根据乘法分配律)

$$= A \cdot 1 + 1 \cdot C$$

(根据 01 律、互补律)

$$= A + C$$

(根据 01 律)

$$(5) Y = A \cdot \overline{B} + \overline{A} \cdot C \cdot D + B + \overline{C} + \overline{D}$$

$$= A \cdot \overline{B} + B + \overline{A} \cdot C \cdot D + \overline{C} + \overline{D}$$

(根据交换律)

$$= A + B + \overline{A} \cdot C \cdot D + \overline{C} + \overline{D}$$

(根据吸收律)

$$= A + B + \overline{A} \cdot C \cdot D + \overline{C \cdot D}$$

(根据反演律)

$$= A + B + \overline{A} + \overline{C \cdot D}$$

(根据吸收律)

$$= A + \overline{A} + B + \overline{C \cdot D}$$

(根据交换律)

$$= 1 + B + \overline{C \cdot D}$$

(根据互补律)

$$= 1$$

(根据 01 律)

5. 写出图 1.3 中的逻辑表达式。

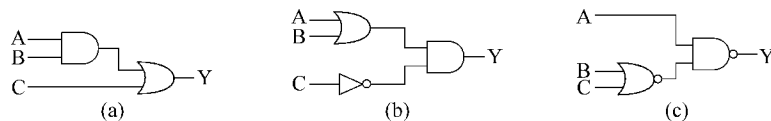


图 1.3

解: (a) $Y = A \cdot B + C$ (b) $Y = (A + B) \cdot \bar{C}$ (c) $Y = A \cdot (B + C)$

6. 根据如图 1.4 所示的逻辑图输入信号 A、B、C 的波形,画出其输出信号 Y 的波形。

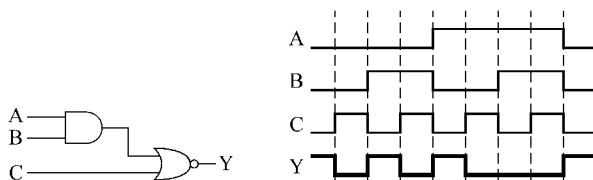


图 1.4

解: 根据题中逻辑图 1.4 可知输出信号的逻辑关系是: $Y = \overline{A \cdot B + C}$, 可见, 只有当 A、B 都为“1”时, 或者 C 为“1”时, 输出信号 Y 才为“0”。由此可以得到图中输出信号 Y 的波形。

7. 三态门有哪三个输出状态? 在数字电路中,它的主要用途是什么?

答: 三态门有 0 态(低电平)、1 态(高电平)和高阻态(断开状态)三种输出状态。在数字电路中,三态门主要用途是进行二进制数据信号的传送与隔离。

8. 根据二进制减法规则,参照“全加器”真值表,写出“全减器”的真值表。

答: 根据减法运算规则,可以得到如表 1.3 所示的“全减器”的真值表。

表 1.3 “全减器”的真值表

输 入			输 出	
A	B	I	S	C
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

其中, A、B 分别是二进制“被减数”和“减数”的某位数字, S 是该位的“差”, C 是该位向高位的借位输出信号, I 是来自低位的借位输入信号。

9. 什么是译码器? 对照 2-4 译码器的真值表, 写出 3-8 译码器的真值表。

答: 译码器是一种信号转换电路, 它将一个二进制编码信号转换成一个对应的逻辑

输出信号。3-8 译码器的真值表如表 1.4 所示。

表 1.4 3-8 译码器真值表

允许 E	编 码 输 入			译 码 输 出								说 明
	C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	
0	×	×	×	0	0	0	0	0	0	0	0	无译出信号
1	0	0	0	1	0	0	0	0	0	0	0	编码 0 译出信号
1	0	0	1	0	1	0	0	0	0	0	0	编码 1 译出信号
1	0	1	0	0	0	1	0	0	0	0	0	编码 2 译出信号
1	0	1	1	0	0	0	1	0	0	0	0	编码 3 译出信号
1	1	0	0	0	0	0	0	1	0	0	0	编码 4 译出信号
1	1	0	1	0	0	0	0	0	1	0	0	编码 5 译出信号
1	1	1	0	0	0	0	0	0	0	1	0	编码 6 译出信号
1	1	1	1	0	0	0	0	0	0	0	1	编码 7 译出信号

10. 什么是优先权编码器？对照 4-2 优先权编码器的真值表，写出 8-3 优先权编码器的真值表。

答：优先权编码器事先安排好了各输入信号产生编码的优先顺序，这个顺序决定了输入信号的优先权。优先权高的信号享有优先产生其编码的权力。当多个输入信号同时有效时，编码器将输出当前优先权最高的那个信号的编码。8-3 优先权编码器的真值表如表 1.5 所示。

表 1.5 8-3 优先权编码真值表

输 入 信 号								编码有效	编 码 输 出			说 明
A ₀	B	C	D	E	F	G	H	CA	Y ₂	Y ₁	Y ₀	
0	0	0	0	0	0	0	0	0	0	0	0	无效输出
1	×	×	×	×	×	×	×	1	0	0	0	输出 A 编码
0	1	×	×	×	×	×	×	1	0	0	1	输出 B 编码
0	0	1	×	×	×	×	×	1	0	1	0	输出 C 编码
0	0	0	1	×	×	×	×	1	0	1	1	输出 D 编码
0	0	0	0	1	×	×	×	1	1	0	0	输出 E 编码
0	0	0	0	0	1	×	×	1	1	0	1	输出 F 编码
0	0	0	0	0	0	1	×	1	1	1	0	输出 G 编码
0	0	0	0	0	0	0	1	1	1	1	1	输出 H 编码

11. 如图 1.5 中的触发器初始状态为 Q=0，试根据输入信号的波形，画出其输出信号 OUT 的相应波形。

解：同步 D 型触发器(图 1.5(a))的写入控制信号 E 是一个高电平有效的信号，只有当 E(=CLK)为高电平时，数据输入 D(=DAT)信号才能写到触发器中，使输出 Q 变成与输入 D 一致的状态，因此输出信号 OUT(=Q)的波形如图 1.5(a)中的粗实线所示。

主从 D 型触发器(图 1.5(b))的写入控制信号 CLK 是一个上升沿有效的信号，只有

当 CLK 由低到高跳变时,数据输入 $D(=\bar{Q})$ 信号才能写到触发器中,使输出 Q 变成与输入 D 一致的状态,因此输出信号 $OUT(=Q)$ 的波形如图 1.5(b)中的粗实线所示。

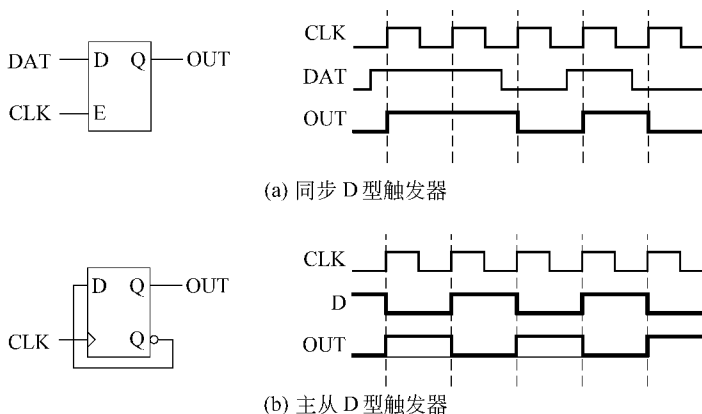


图 1.5

1.3 第 3 章习题

1. 写出下列机器数所能表示的最大真值和最小真值。

(1) 字长为 6 的原码 (2) 字长为 10 的补码

答: (1) 根据原码定义可知,字长为 n 位的原码表示范围是: $-(2^{n-1}-1) \sim 2^{n-1}-1$ 。因此字长为 6 位的原码,最大真值是 $31(=2^5-1)$,最小真值是 $-31(=-2^5+1)$ 。

(2) 根据补码定义可知,字长为 n 位的补码表示范围是: $-2^{n-1} \sim 2^{n-1}-1$ 。因此字长为 10 位的补码,最大真值是 $511(=2^9-1)$,最小真值是 $-512(=-2^9)$ 。

2. 设字长为 8,分别写出下列二进制数的原码和补码。

(1) 1000001 (2) -1000001

(3) 1101 (4) -1101

解: (1) $[1000001]_{\text{原}} = 01000001$ (2) $[-1000001]_{\text{原}} = 11000001$
 $[1000001]_{\text{补}} = 01000001$ $[-1000001]_{\text{补}} = 10111111$
 (3) $[1101]_{\text{原}} = 00001101$ (4) $[-1101]_{\text{原}} = 10001101$
 $[1101]_{\text{补}} = 00001101$ $[-1101]_{\text{补}} = 11110011$

3. 设字长为 8,分别写出下列二进制数的补码和反码。

(1) -1000100 (2) -1110011 (3) -1111000

解: (1) $[-1000100]_{\text{补}} = 10111100$ (2) $[-1110011]_{\text{补}} = 10001101$
 $[-1000100]_{\text{反}} = 10111011$ $[-1110011]_{\text{反}} = 10001100$
 (3) $[-1111000]_{\text{补}} = 10001000$
 $[-1111000]_{\text{反}} = 10000111$

4. 若将机器数 11101110 分别解释成原码、补码和反码,那么它所对应的真值分别是多少?

解: 若将机器数解释成原码 $[x]_{\text{原}} = 11101110$, 因为 $\text{MSB}=1$, 所以它的真值是 $x = -1101110\text{B} = -110$ 。

若将机器数解释成补码 $[x]_{\text{补}} = 11101110$, 由于 $\text{MSB}=1$, 无法直接读出其真值。但对机器数取补后有: $00010010 = [-x]_{\text{补}}$, $-x = 10010$, 所以它的真值是 $x = -10010\text{B} = -18$ 。

若将机器数解释成反码 $[x]_{\text{反}} = 11101110$, 由于 $\text{MSB}=1$, 无法直接读出其真值。但对机器数取反后有: $00010001 = [-x]_{\text{反}}$, $-x = 10001$, 所以它的真值是 $x = -10001\text{B} = -17$ 。

5. 设字长为 8, 利用补码加法公式计算二进制数 $x+y$ 的值。

$$(1) x = -10110, y = -1010 \quad (2) x = -10110, y = 1010$$

$$(3) x = -1010, y = 10110 \quad (4) x = 10110, y = 1010$$

解: (1) $[x]_{\text{补}} = 11101010, [y]_{\text{补}} = 11110110$

$$\begin{aligned} [x+y]_{\text{补}} &= 11101010 + 11110110 \quad (\text{mod } 2^8) \\ &= 11000000 \end{aligned}$$

$$\begin{array}{r} 11101010 \\ + 11110110 \\ \hline 1 \quad 11000000 \end{array}$$

由于符号位为 1, 结果为负, 其相反数的补码为 $[-(x+y)]_{\text{补}} = 00100000$, 所以 $-(x+y) = 100000$, $x+y = -100000$ 。

$$(2) [x]_{\text{补}} = 11101010, [y]_{\text{补}} = 00001010$$

$$\begin{aligned} [x+y]_{\text{补}} &= 11101010 + 00001010 \\ &= 11110100 \end{aligned}$$

$$\begin{array}{r} 11101010 \\ + 00001010 \\ \hline 11110100 \end{array}$$

由于符号位为 1, 结果为负, 其相反数的补码为 $[-(x+y)]_{\text{补}} = 00001100$, 所以 $-(x+y) = 1100$, $x+y = -1100$ 。

$$(3) [x]_{\text{补}} = 11110110, [y]_{\text{补}} = 00010110$$

$$\begin{aligned} [x+y]_{\text{补}} &= 11110110 + 00010110 \quad (\text{mod } 2^8) \\ &= 00001100 \end{aligned}$$

$$\begin{array}{r} 11110110 \\ + 00010110 \\ \hline 1 \quad 00001100 \end{array}$$

由于符号位为 0, 结果为正, 所以 $x+y = 1100$ 。

$$(4) [x]_{\text{补}} = 00010110, [y]_{\text{补}} = 00001010$$

$$\begin{aligned} [x+y]_{\text{补}} &= 00010110 + 00001010 \\ &= 00001100 \end{aligned}$$

$$\begin{array}{r}
 00010110 \\
 + 00001010 \\
 \hline
 00100000
 \end{array}$$

由于符号位为 0, 结果为正, 所以 $x+y=100000$ 。

6. 设字长为 8, 利用补码减法公式计算二进制数 $x-y$ 的值。

$$(1) x=-10110, y=-1010 \quad (2) x=-10110, y=1010$$

$$(3) x=1010, y=-10110 \quad (4) x=10110, y=1010$$

解: (1) $[x]_{\text{补}}=11101010$, $[y]_{\text{补}}=11110110$

$$\begin{aligned}
 [x-y]_{\text{补}} &= 100000000 + 11101010 - 11110110 \\
 &= 1110100
 \end{aligned}$$

$$\begin{array}{r}
 1 \quad 11101010 \\
 - \quad 11110110 \\
 \hline
 1110100
 \end{array}$$

由于符号位为 1, 结果为负, 其相反数的补码为 $[-(x-y)]_{\text{补}}=00001100$, 所以 $-(x-y)=1100$, $x-y=-1100$ 。

$$(2) [x]_{\text{补}}=11101010, [y]_{\text{补}}=00001010$$

$$\begin{aligned}
 [x-y]_{\text{补}} &= 100000000 + 11101010 - 00001010 \quad (\text{mod } 2^8) \\
 &= 11100000
 \end{aligned}$$

$$\begin{array}{r}
 1 \quad 11101010 \\
 - \quad 00001010 \\
 \hline
 11100000 \quad (\text{mod } 2^8)
 \end{array}$$

由于符号位为 1, 结果为负, 其相反数的补码为 $[-(x-y)]_{\text{补}}=00100000$, 所以 $-(x-y)=100000$, $x-y=-100000$ 。

$$(3) [x]_{\text{补}}=00001010, [y]_{\text{补}}=11101010$$

$$\begin{aligned}
 [x-y]_{\text{补}} &= 100000000 + 00001010 - 11101010 \\
 &= 00100000
 \end{aligned}$$

$$\begin{array}{r}
 1 \quad 00001010 \\
 - \quad 11101010 \\
 \hline
 00100000
 \end{array}$$

由于符号位为 0, 结果为正, 所以 $x-y=100000$ 。

$$(4) [x]_{\text{补}}=00010110, [y]_{\text{补}}=00001010$$

$$\begin{aligned}
 [x-y]_{\text{补}} &= 100000000 + 00010110 - 00001010 \quad (\text{mod } 2^8) \\
 &= 00001100
 \end{aligned}$$

$$\begin{array}{r}
 1 \quad 00010110 \\
 - \quad 00001010 \\
 \hline
 00001100 \quad (\text{mod } 2^8)
 \end{array}$$

由于符号位为 0, 结果为正, 所以 $x-y=1100$ 。

7. 下面两个补码相加后是否会出现溢出? 为什么?

(1) 01000000 和 01000011 (2) 10001000 和 11101110

解: (1) 因为 $[x]_{\text{补}} = 01000000$, $[y]_{\text{补}} = 01000011$

所以 $[x+y]_{\text{补}} = 01000000 + 01000011$

$$= 10000011$$

由此得到 $x+y < 0$ 的错误结果, 出错的原因是由于真值运算结果:

$$x+y = 1000000 + 1000011 = 1000011$$

已超出了 8 位补码的上限 +1111111。

(2) 因为 $[x]_{\text{补}} = 10001000$, $[y]_{\text{补}} = 11101110$

所以 $[x+y]_{\text{补}} = 10001000 + 11101110 \pmod{2^8}$

$$= 01110110$$

由此得到 $x+y > 0$ 的错误结果, 出错的原因是由于真值运算结果:

$$x+y = -1111000 + (-10010) = -10001010$$

已超出了 8 位补码的下限 -10000000。

8. 下面两个补码相减后是否会出现溢出? 为什么?

(1) 01100100 和 11100000 (2) 10001000 和 00010010

解: (1) 因为 $[x]_{\text{补}} = 01100100$, $[y]_{\text{补}} = 11100000$

所以 $[x-y]_{\text{补}} = 10000000 + 01100100 - 11100000$

$$= 10000100$$

由此得到 $x+y < 0$ 的错误结果, 出错的原因是由于真值运算结果:

$$x-y = 1100100 - (-100000) = 10000100$$

已超出了 8 位补码的上限 +1111111。

(2) 因为 $[x]_{\text{补}} = 10001000$, $[y]_{\text{补}} = 00010010$

所以 $[x-y]_{\text{补}} = 10000000 + 10001000 - 00010010 \pmod{2^8}$

$$= 01110110$$

由此得到 $x-y > 0$ 的错误结果, 出错的原因是由于真值运算结果:

$$x-y = -1111000 - 10010 = -10001010$$

已超出了 8 位补码的下限 -10000000。

9. 分别写出下列二进制数的补码和移码。其中, 哪些机器数可通过减法借位的状态来判断它们的大小? 请举例说明。

(1) $x = 1100100$, $y = 110010$

(2) $x = -1011111$, $y = -1010101$

(3) $x = 1001011$, $y = -1010101$

(4) $x = -1011111$, $y = 1100100$

解: (1) $[x]_{\text{补}} = 01100100$, $[y]_{\text{补}} = 00110010$

$$[x]_{\text{移}} = 10000000 + [x]_{\text{补}} \pmod{2^8} = 11100100$$

$$[y]_{\text{移}} = 10000000 + [y]_{\text{补}} \pmod{2^8} = 10110010$$

(2) $[x]_{\text{补}} = 10100001$, $[y]_{\text{补}} = 10101011$

$$[x]_{\text{移}} = 10000000 + [x]_{\text{补}} \pmod{2^8} = 00100001$$

$$[y]_{\text{移}} = 10000000 + [y]_{\text{补}} \pmod{2^8} = 00101011$$

$$(3) [x]_{\text{补}} = 01001011, [y]_{\text{补}} = 10101011$$

$$[x]_{\text{移}} = 10000000 + [x]_{\text{补}} \pmod{2^8} = 11001011$$

$$[y]_{\text{移}} = 10000000 + [y]_{\text{补}} \pmod{2^8} = 00101011$$

$$(4) [x]_{\text{补}} = 10100001, [y]_{\text{补}} = 01100100$$

$$[x]_{\text{移}} = 10000000 + [x]_{\text{补}} \pmod{2^8} = 00100001$$

$$[y]_{\text{移}} = 10000000 + [y]_{\text{补}} \pmod{2^8} = 11100100$$

其中,对于补码而言,当两个同号数参与比较时,通过它们相减的借位状态可以了解它们之间的大小关系,例如从(1)、(2)两题的补码减法中,可以看出同号数的大小关系。

题(1)补码减法:

$$\begin{array}{r} \text{借位 } 0 \quad 01100100 \\ - \quad 00110010 \\ \hline 00110010 \end{array}$$

题(2)补码减法:

$$\begin{array}{r} \text{借位 } 1 \quad 10100001 \\ - \quad 10101011 \\ \hline 11110110 \end{array}$$

在题(1)中由于没有借位,所以 $x > y$;而在题(2)中由于有借位,所以 $x < y$ 。

对于两个异号数相减,由于实质上执行的是相加运算,这时借位信号已没有意义,不能通过有无借位来判定 x 与 y 的大小关系。但如果相减结果没有溢出,还可以通过符号位来断定 x 与 y 的大小关系。

移码表示的机器数,可通过减法借位的状态来判断它们的大小。因而避免了因补码相减结果溢出而无法判断它们大小的问题,例如从题(3)、题(4)中的补码减法中,

题(3)补码减法:

$$\begin{array}{r} \text{借位 } 1 \quad 01001011 \\ - \quad 10101011 \\ \hline 10100000 \end{array}$$

题(4)补码减法:

$$\begin{array}{r} \text{借位 } 0 \quad 10100001 \\ - \quad 01100100 \\ \hline 00111101 \end{array}$$

可以看出:两个异号数相减实质上是相加运算,这里借位信号已没有意义,不能通过有无借位来判定 x 、 y 的大小。由于结果已经溢出,因此也不能通过符号位的状态来判定 x 、 y 的大小。但从(3)、(4)两题中的移码减法中,

题(3)移码减法:

$$\begin{array}{r} \text{借位 } 0 \quad 11001011 \\ - \quad 00101011 \\ \hline 10100000 \end{array}$$

题(4)移码减法:

$$\begin{array}{r} \text{借位 } 1 \quad 00100001 \\ - \quad 11100100 \\ \hline 00111101 \end{array}$$

可以看出它们的大小。因为,移码已将符号数转换成无符号数,两个无符号数相减,借位信号是有意义的。这里,可以通过无借位来断定 $x > y$,通过有借位来断定 $x < y$ 。

10. 分别写出 25.625 和 -0.375 的 IEEE 短浮点数表示形式。

解:分别将十进制 25.625 和 -0.375 数转换成二进制数:

$$11001.101\text{B} \quad \text{和} \quad -0.011\text{B}$$

然后分别将对应的二进制数写成规格化数:

$$1.1001101 \times 10^{100}\text{B} \quad \text{和} \quad -1.1 \times 10^{-10}\text{B}$$

接着将规格化二进制数中的指数 $e(=100 \text{ 和 } -10)$ 表示成 8 位阶码 $(1111111+e)$ ：

10000011 和 01111101

最后按照 1 位尾数符号、8 位阶码、23 位尾数(不含整数“1”)的顺序,写出相应的 32 位 IEEE 短浮点数:

0,100 0001 1,100 1101 0000 0000 0000 0000B=41CD0000H

和

1,011 1110 1,100 0000 0000 0000 0000 0000B=BEC00000H

11. 分别写出 156.375 的二进制数和 BCD 码。二进制数与 BCD 码是一回事吗?为什么?

答: 156.375 的二进制数和 BCD 码分别为:

10011100.011B 和 0001 0101 0110.0011 0111 0101BCD

可见,二进制数与 BCD 码不是一回事。因为 BCD 码是十进制数字的二进制编码,它不是数而是一个编码,它的计数规则不是二进制而是十进制。

12. 写出数字 0~9 的 ASCII 码,如何将这些 ASCII 码转换成相应的 BCD 码? 写出大写字母 A、B、C、X、Y、Z 的 ASCII 码,如何将大写字母的 ASCII 码转换成相应的小写字母的 ASCII 码?

答: 数字 0~9 的 ASCII 码是: 011 0000B~011 1001B (30H~39H),而它们所对应的 BCD 码是: 0000B~1001B(0H~9H)。可见,只要将 ASCII 码的高三位去掉,即可得到相应的 BCD 码。

大写字母 A、B、C、X、Y、Z 的 ASCII 码分别是: 41H、42H、43H、58H、59H、5AH。

小写字母 a、b、c、x、y、z 的 ASCII 码分别是: 61H、62H、63H、78H、79H、7AH。

可见,只要将大写字母的 ASCII 码加上 20H,即可得到相应的小写字母的 ASCII 码。

1.4 第4章习题

1. 在模型计算机中,CPU、存储器和 I/O 接口电路是通过哪三组总线相互连在一起的? 这些总线各有什么用途?

答: 它们是通过地址总线、数据总线和控制总线相互连在一起的,这些总线的用途说明如下。

- 地址总线: 用来传送 CPU 发出的地址信息,以便选定 CPU 将要访问的存储单元或 I/O 接口中的电路。
- 数据总线: 用来实现 CPU 与存储器以及 CPU 与 I/O 接口等电路之间的数据传送。
- 控制总线: 用来传送系统各电路的控制信号,如 CPU 发给外电路的读、写控制信号,外电路发给 CPU 的复位、同步时钟等信号。

2. 堆栈的存取原则是什么? 当堆栈中无数据时, 堆栈指针 SP 指向哪里? 当堆栈有数据时, SP 指向哪里?

答: 堆栈的存取原则是后进先出(List In First Out, LIFO), 即后存进堆栈的数据将先取出来。

当堆栈中无任何数据时, SP 所指定的位置称为栈底(Bottom), 栈底地址就是 SP 的初值。当堆栈有数据时, SP 将始终指向栈顶(Top), 位于栈顶的数据是最后一个被推入堆栈的数据。

3. 写出下列机器数经 ALU 运算后的结果和标志 ZF、CF 的状态。若将这些机器数当作有符号数(补码), 哪些运算会出现溢出?

- (1) $A8H + 58H$ (2) $23H + 52H$
 (3) $54H + 47H$ (4) $35H - 4EH$
 (5) $9DH - 4BH$ (6) $4CH - 89H$

解: (1)

十六进制加法:

$$\begin{array}{r} A8H \\ + 58H \\ \hline 1 \quad 00H \\ \text{进位} \quad \underbrace{\hspace{1cm}}_{8 \text{ 位二进制结果}} \end{array}$$

ALU 加法:

$$\begin{array}{r} 10101000B \\ + 01011000B \\ \hline 1 \quad 00000000B \\ \text{进位} \quad \underbrace{\hspace{1cm}}_{8 \text{ 位结果}} \end{array}$$

8 位运算结果为 0, 标志 $ZF=1$ 、 $CF=1$ 。两个异号数相加不会出现溢出。

(2)

十六进制加法:

$$\begin{array}{r} 23H \\ + 52H \\ \hline 0 \quad 75H \\ \text{进位} \quad \underbrace{\hspace{1cm}}_{8 \text{ 位二进制结果}} \end{array}$$

ALU 加法:

$$\begin{array}{r} 00100011B \\ + 01010010B \\ \hline 0 \quad 01110101B \\ \text{进位} \quad \underbrace{\hspace{1cm}}_{8 \text{ 位结果}} \end{array}$$

8 位运算结果为 75H(01110101B), 标志 $ZF=0$ 、 $CF=0$ 。两个正数相加结果仍为正数, 没有出现溢出。

(3)

十六进制加法:

$$\begin{array}{r} 54H \\ + 47H \\ \hline 0 \quad 9BH \\ \text{进位} \quad \underbrace{\hspace{1cm}}_{8 \text{ 位二进制结果}} \end{array}$$

ALU 加法:

$$\begin{array}{r} 01010100B \\ + 01000111B \\ \hline 0 \quad 10011011B \\ \text{进位} \quad \underbrace{\hspace{1cm}}_{8 \text{ 位结果}} \end{array}$$

8 位运算结果为 9BH(10011011B), 标志 $ZF=0$ 、 $CF=0$ 。两个正数相加结果变为负数, 出现溢出。

(4)

十六进制减法:

$$\begin{array}{r}
 \text{借位 } 1 \quad 35\text{H} \\
 - \quad 4\text{EH} \\
 \hline
 \text{E7H} \\
 \text{8位二进制结果}
 \end{array}$$

ALU 减法(补码加法):

$$\begin{array}{r}
 \quad \quad 00110101\text{B} \\
 \quad \quad 10110001\text{B} \\
 + \quad \quad \quad 1 \\
 \hline
 \text{进位 } 0 \quad 11100111\text{B} \\
 \text{8位结果}
 \end{array}$$

01001110B 取补

8 位运算结果为 E7H(11100111B), 标志 ZF=0、CF= $\overline{\text{Cm}}$ =1。两个同号数相减结果不会出现溢出。

(5)

十六进制减法:

$$\begin{array}{r}
 \text{借位 } 0 \quad 9\text{DH} \\
 - \quad 4\text{BH} \\
 \hline
 52\text{H} \\
 \text{8位二进制结果}
 \end{array}$$

ALU 减法(补码加法):

$$\begin{array}{r}
 \quad \quad 10011101\text{B} \\
 \quad \quad 10110100\text{B} \\
 + \quad \quad \quad 1 \\
 \hline
 \text{进位 } 1 \quad 01010010\text{B} \\
 \text{8位结果}
 \end{array}$$

01001011B 取补

8 位运算结果为 52H(01010010B), 标志 ZF=0、CF= $\overline{\text{Cm}}$ =0。一个负数减去一个正数结果应为负数, 但这里的结果却为正数, 因此出现溢出。

(6)

十六进制减法:

$$\begin{array}{r}
 \text{借位 } 1 \quad 4\text{CH} \\
 - \quad 89\text{H} \\
 \hline
 \text{C3H} \\
 \text{8位二进制结果}
 \end{array}$$

ALU 减法(补码加法):

$$\begin{array}{r}
 \quad \quad 01001100\text{B} \\
 \quad \quad 01110110\text{B} \\
 + \quad \quad \quad 1 \\
 \hline
 \text{进位 } 0 \quad 11000011\text{B} \\
 \text{8位结果}
 \end{array}$$

10001001B 取补

8 位运算结果为 C3H(11000011B), 标志 ZF=0、CF= $\overline{\text{Cm}}$ =1。一个正数减去一个负数结果应为正数, 但这里的结果却为负数, 因此出现溢出。

4. 指出下列模型计算机指令中操作数的寻址方式。

(1) ROR BR

(2) ADC AR, #25

(3) SBB AR, 50H

(4) ST BR, 3CH[IX]

(5) LD AR, [SP].2

(6) JMP 30H

答: (1) 寄存器寻址

(2) 寄存器寻址、立即数寻址

(3) 寄存器寻址、直接寻址

(4) 寄存器寻址、变址寻址

(5) 寄存器寻址、基址寻址

(6) 直接寻址

5. 试编写一个程序段, 将单元(10H)和(11H)的内容相互交换。

解:

```

LD    AR, 10H      ;将(10H)内容放在 AR 中
LD    BR, 11H      ;将(11H)内容放在 BR 中

```

```

ST    AR,11H      ;将 AR 内容放在 (11H) 中
ST    BR,10H      ;将 BR 内容放在 (10H) 中

```

6. 试编写一个程序段,将 AR 中的两位 BCD 拆开,低位 BCD 保留在 AR 的低 4 位中,高位 BCD 放到 BR 的低 4 位中。

解: 设 $AR = D_1 D_0$, 其中 D_1 和 D_0 分别为高位 BCD 和低位 BCD。

```

MOV   BR,AR        ;将  $D_1 D_0$  放在 BR 中
AND   AR,#0FH       ; $AR = 0D_0$ 
AND   BR,#F0H       ;将  $D_1 0$  放在 BR 中
ROR   BR            ;BR 循环右移 4 位
ROR   BR
ROR   BR
ROR   BR            ; $BR = 0 D_1$ 

```

7. 试编写一个程序段,将 AR 的低 2 位清 0,中间 4 位取反,保留高 2 位。

解: 设 $AR = A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$ 。

```

MOV   BR,AR        ;将 AR 内容暂存在 BR 中
AND   AR,#COH       ; $AR = A_7 A_6 000000$ 
NOT   BR            ; $BR = \overline{A_7} \overline{A_6} \overline{A_5} \overline{A_4} \overline{A_3} \overline{A_2} \overline{A_1} \overline{A_0}$ 
AND   BR,#3CH       ; $BR = 00 \overline{A_5} \overline{A_4} \overline{A_3} \overline{A_2} 00$ 
OR    AR,BR         ; $AR = A_7 A_6 \overline{A_5} \overline{A_4} \overline{A_3} \overline{A_2} 00$ 

```

8. 已知单元(10H)、(11H)中存放着两个无符号数,试编写一个程序段,将其中较小的数放到单元(12H)中。

解: 设单元(10H)、(11H)中分别存放两个无符号数 x 和 y 。

```

LD     AR,10H       ;取 x
CMP    AR,11H       ;比较:  $x - y$ 
JC     XLTY          ;若  $x < y$  转至地址 XLTY 处执行
LD     AR,11H       ;取 y
XLTY:  ST    AR,12H  ;存小数

```

9. 已知单元(10H)中存放着一个有符号数 x (补码),试编写一个程序段,确定以下函数值:

$$y = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

并将函数值 y 存放到单元(11H)中。

解: 方法一:

按照教材例 4.14 中判断符号数 x (补码)正负的方法,可以编写出如下程序段:

```

LD     AR,10H       ;取 x 到 AR 中
CMP    AR,#0        ;AR 与 0 比较
JZ     WFUN         ;若  $AR = 0$  转至地址 WFUN

```



```

CMP    AR, #128      ;AR 与 128 比较
LD     AR, #1
JC     WFUN           ;若 AR<128 (即 x>0) 转至地址 WFUN
LD     AR, #-1        ;
WFUN:  ST     AR, 11H   ;存放函数值

```

方法二:

利用循环左移指令 ROL, 在将 AR 的 MSB 位移到 CF 中的同时检测 AR 的零状态, 然后通过 CF 和 ZF 来完成对符号数 x 正、负和 0 的判断。

```

LD     AR, 10H        ;取 x 到 AR 中
ROL    AR              ;CF←MSB、ZF←AR 零状态
JZ     WFUN           ;若 AR=0 转至地址 WFUN
LD     AR, #1
JNC    WFUN           ;若 MSB=0 (即 x>0) 转至地址 WFUN
LD     AR, #-1        ;
WFUN:  ST     AR, 11H   ;存放函数值

```

10. 试编写一个程序段, 完成 $1+2+3+\dots+20$ 的运算, 并将运算结果放在 AR 中。

解:

```

LD     AR, #0          ;累加和清 0
LD     BR, #20         ;置计数初值
LOOP:  CLRC            ;清 CF
      ADC     AR, BR    ;累加: AR←AR+BR
      DEC     BR        ;循环计数
      JNZ     LOOP     ;若 BR≠0 转 LOOP 继续

```

11. 已知在首地址为 10H 和 14H 的存储区中, 分别存放着两个字长为 4 字节的数据, 试编写一个程序段完成它们的相加运算, 并将运算结果存放到首地址为 18H 的存储区中。

解:

```

LD     BR, #4          ;将循环计数初值 4 装入 BR
LD     IX, #0          ;将偏移量 0 装入 IX
CLRC
LOOP:  LD     AR, 10H[IX] ;取“被加数”的一字节到 AR 中
      ADC     AR, 14H[IX] ;加上“加数”的对应字节 (包括低字节的进位)
      ST     AR, 18H[IX] ;保存“和”到结果的对应字节中
      INC     IX        ;计算下一字节地址的偏移量
      DEC     BR        ;循环控制计数 BR←BR-1
      JNZ     LOOP     ;若 BR≠0 则转至标号 LOOP 处

```

12. 已知在首地址为 10H 和 14H 的存储区中, 分别存放着两个字长为 4 字节的数据, 试编写一个程序段完成它们的相减运算, 并将运算结果存放到首地址为 18H 的存储

区中(假设首地址为 10H 的 4 字节数据为被减数)。

解:

```

LD    BR, #4      ;将循环计数初值 4 装入 BR
LD    IX, #0      ;将偏移量 0 装入 IX
CLRC                ;清借位 CF←0
LOOP: LD    AR, 10H[IX] ;取“被减数”的一个字节到 AR 中
      SBB    AR, 14H[IX] ;减去“减数”的对应字节 (包括低字节的借位)
      ST     AR, 18H[IX] ;将“差”保存到结果的对应字节中
      INC    IX          ;计算下一字节地址的偏移量
      DEC    BR          ;循环控制计数 BR←BR-1
      JNZ    LOOP        ;若 BR≠0 则转至标号 LOOP 处

```

13. 试编写一个子程序,完成交换寄存器 AR 和 BR 内容的操作(要求:子程序执行不能影响其他寄存器)。

解:方法一:

```

XCHR:  PUSH    IX      ;暂存 IX 内容
      MOV     IX, AR    ;AR 内容暂存到 IX 中
      MOV     AR, BR    ;BR 内容放在 AR 中
      MOV     BR, IX    ;IX 中的 AR 内容放在 BR 中
      POP     IX        ;恢复 IX 内
      RET

```

方法二:

```

XCHR:  PUSH    AR      ;AR 内容推入堆栈
      PUSH    BR      ;BR 内容推入堆栈
      POP     AR      ;将堆栈中 BR 内容弹出放在 AR 中
      POP     BR      ;将堆栈中 AR 内容弹出放在 BR 中
      RET

```

14. 试编写一个子程序,判断寄存器 AR 中有符号数(补码)的正负,并按以下要求设置标志 CF、ZF 的状态(要求:子程序执行不能破坏 AR 的内容)。

$$AR \begin{cases} >0 & CF=0, ZF=0 \\ =0 & CF=0, ZF=1 \\ <0 & CF=1, ZF=0 \end{cases}$$

解:方法一:

```

SNGAR:  CMP     AR, #128
      JNC     XNEG    ;若 AR>128 (真值为负),转至 XNEG
      CMP     AR, #0
      RET      ;若 AR=0,返回 (CF=0, ZF=1)
      RET      ;若 AR>0,返回 (CF=0, ZF=0)
XNEG:   SETC
      RET      ;返回 (CF=1, ZF=0)

```

方法二:

NGAR:	ROL	AR	;循环左移 AR
	ROR	AR	;循环右移 AR, CF $\leftarrow$ MSB, ZF $\leftarrow$ AR 零状态
	RET		

1.5 第5章习题

1. 只读存储器 ROM 与随机存储器 RAM 的主要区别是什么? MOS 型存储芯片的种类有哪些? 存储芯片有哪些主要的技术指标?

答: ROM 中存放的内容只能读出,不能被修改,断电后 ROM 中存放的信息不会消失;RAM 中存放的内容可以任意读写,但断电后 RAM 中存放的信息将全部消失。

MOS 型存储芯片的种类主要有:只读存储器 ROM 和随机存储器 RAM 两大类。其中,只读存储器包含掩膜 ROM、紫外线擦除可编程 EPROM、电擦除可编程 EEPROM 等;随机存储器包含静态 SRAM、动态 DRAM 等。

存储芯片主要的技术指标有:芯片容量和存取时间。其中,芯片容量用芯片中的单元总数与单元位数的乘积来表示,存取时间是指数据存入存储芯片或从存储芯片中取出所需的最短时间。

2. 在容量为 $8K \times 8$ 的存储芯片中,共有多少个存储单元? 每个单元的位数是多少? 该芯片分别有多少个地址和数据引脚?

答:该芯片中共有 $8192(8K = 8 \times 2^{10} = 2^{13})$ 个存储单元,每个单元的位数为 8 位,通常该芯片具有 13 条地址引脚、8 条数据引脚。

3. 什么是基本存储单元? 在 ROM、SRAM 和 DRAM 的基本存储单元中,分别是利用什么方法来记忆二进制 0、1 信息的?

答:用来记忆一位二进制信息的存储电路称为基本存储单元,它是构成存储器的基础。

ROM 基本存储单元中是利用有、无晶体管来记忆二进制 0、1 信息的;SRAM 基本存储单元是利用 R-S 触发器来记忆二进制 0、1 信息的,该触发器是由两个交叉连接的反相器构成的;DRAM 基本存储单元是利用存储电容是否充有电荷来记忆二进制 0、1 信息的。

4. 在 SRAM 芯片中,有哪些主要的功能电路? 在存储芯片中,为什么通常将基本单元阵列排列成方阵?

答:SRAM 芯片中的主要功能电路有:静态基本单元阵列、行/列地址译码器、列选电路和读写控制等电路。

众所周知,在矩形面积一定的情况下,正方形的边长最短。将这一结论用到由基本单元阵列构成的方阵时就是:在基本单元总数一定的情况下,将其排列成方阵可使行/列选择线的总数最少,从而可以减少行/列地址译码器输出引脚的总数,简化译码电路。

5. 用容量为 $1K \times 4$ 、 $2K \times 8$ 或 $8K \times 8$ 的存储芯片组成一个容量为 $64K \times 8$ 的存储

器,分别需要多少片芯片? 其中,片内地址分别是多少位? 芯片地址分别是多少位?

答: 如果用容量为 $1K \times 4$ 的存储芯片构成存储器,共需要 128 片芯片,其中片内地址 10 位,芯片地址 6 位。

如果用容量为 $2K \times 8$ 的存储芯片构成存储器,共需要 32 片芯片,其中片内地址 11 位,芯片地址 5 位。

如果用容量为 $8K \times 8$ 的存储芯片构成存储器,共需要 8 片芯片,其中片内地址 13 位,芯片地址 3 位。

6. 已知某存储芯片的容量为 $1K \times 4$ 。试画出用这种芯片构成的容量为 $4K \times 8$ 的存储器线路图,并要求该存储器的地址空间为 $1000H \sim 1FFFH$ (假定地址总线为 16 位)。

解: $4K \times 8$ 的存储器线路图如图 1.6 所示。

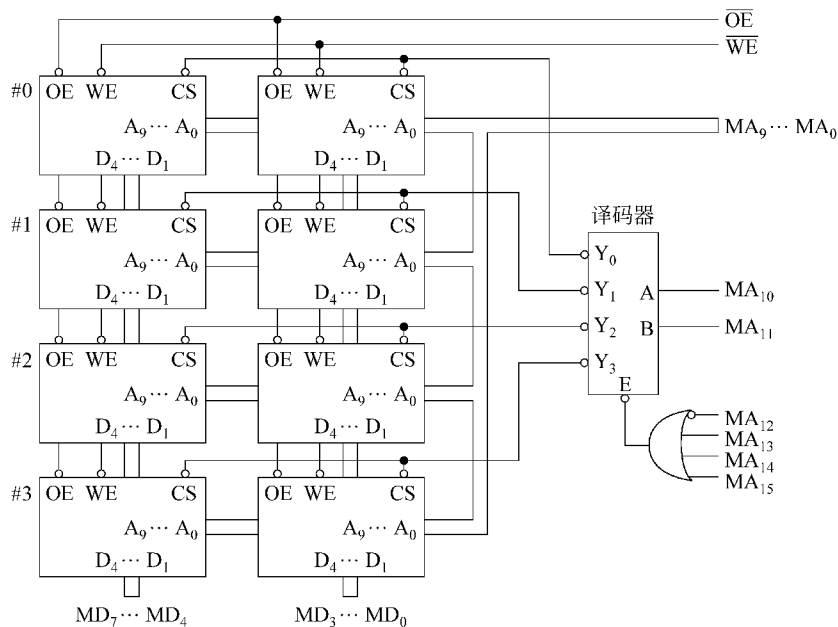


图 1.6

根据该存储器的地址空间可知,当多余地址 $MA_{15} \sim MA_{12}$ 的状态为 0001 时,将选中该存储器。因此译码器的允许控制端 E 必须满足如下逻辑关系:

$$E = \overline{MA}_{15} \cdot \overline{MA}_{14} \cdot \overline{MA}_{13} \cdot MA_{12}$$

$$\overline{E} = MA_{15} + MA_{14} + MA_{13} + \overline{MA}_{12}$$

7. 某存储器的线路图如图 1.7 所示。

(1) 分别写出该存储器中 ROM 和 RAM 的地址空间。

(2) 该存储器的总容量和字长分别是多少?

已知:

CS 片选

OE 输出允许 WE 写允许