

第3章

计算机的基本数字逻辑电路

计算机的硬件系统是由许许多多的逻辑电路组成的,如算术逻辑单元电路、触发器电路、寄存器电路、存储器电路等。本章仅对微型计算机中最常见的基本电路部件做简单介绍。

学习目标

- (1) 了解:基本逻辑运算及其运算规则。
- (2) 理解:基本逻辑电路、二进制的加/减法电路、触发器、寄存器、译码器、三态输出电路的原理和功能。

3.1 逻辑代数

逻辑代数也称为开关代数或布尔代数,和一般代数不同的是:

(1) 逻辑代数中变量只有两种可能的数值:0或1。逻辑代数变量的数值并不表示大小,只代表某种物理量的状态。例如,用于开关中,0代表关(断路)或低电位,1代表开(通路)或高电位;用于逻辑推理中,1代表正确(真),0代表错误(假)。

(2) 逻辑代数只有3种基本运算方式:“与”运算(逻辑乘)、“或”运算(逻辑加)和“取反”运算(逻辑非)。其他逻辑运算均由这3种基本运算构成,如与非运算、或非运算、异或运算、同或运算等。下面来看看这3种基本运算及其运算规则和一般代数有什么区别。

3.1.1 “与”运算

若逻辑变量A、B进行与运算,L表示其运算结果,则其逻辑表达式为:

$$L=AB \quad \text{或} \quad L=A \wedge B \quad \text{或} \quad L=A \cdot B$$

其基本运算规则为: $0 \cdot 0=0$ $0 \cdot 1=0$ $1 \cdot 0=0$ $1 \cdot 1=1$

$$A \cdot 1=A \quad A \cdot 0=0 \quad A \cdot A=A \quad A \cdot \bar{A}=0$$

注意与一般代数的区别,此处的A为逻辑变量,其取值只能是0或1。由其运算结果可归纳为:二者为真则结果必为真,有一为假则结果必为假。同样,这个结论也可推广到多个变量:各变量均为真则结果必为真,有一为假则结果必为假。

从上可知,在多输入“与”门电路中,只要其中一个输入为0,则输出必为0;只有全部输入均为1时,输出才为1。

有时也将与运算称为“逻辑乘”。当A和B为多位二进制数时,如:

$$A=A_1 A_2 A_3 \cdots A_n$$

$$B=B_1 B_2 B_3 \cdots B_n$$

则进行“逻辑乘”运算时,各对应位分别进行“与”运算:

$$Y = A \cdot B$$

$$= (A_1 \cdot B_1)(A_2 \cdot B_2)(A_3 \cdot B_3) \cdots (A_n \cdot B_n)$$

【例 3-1】 设 $A=11001010B$, $B=00001111B$, 求: $Y=A \cdot B$ 。

【解】 $Y = A \cdot B$

$$= (1 \cdot 0)(1 \cdot 0)(0 \cdot 0)(0 \cdot 0)(1 \cdot 1)(0 \cdot 1)(1 \cdot 1)(0 \cdot 1)$$

$$= 00001010$$

写成竖式则为:

$$\begin{array}{r} 11001010 \\ \wedge) 00001111 \\ \hline 00001010 \end{array}$$

由此可见,用“0”和一个数位相“与”,就是将其“抹掉”而成为“0”(即:将其置0);用“1”和一个数位相“与”,就是将此数位“保存”下来。这种方法在计算机的程序设计中经常会用到,称为“屏蔽”。上面的B数(0000 1111)称为“屏蔽字”,它将A数的高4位屏蔽起来,使其都变成0了。

3.1.2 “或”运算

若逻辑变量A、B进行或运算,L表示其运算结果,则其逻辑表达式为:

$$L = A + B \quad \text{或} \quad L = A \vee B$$

其基本运算规则为: $0+0=0$ $0+1=1$ $1+0=1$ $1+1=1$

$$A+0=A \quad A+1=1 \quad A+A=A \quad A+\bar{A}=1$$

注意与一般代数的区别,此处的A为逻辑变量,其取值只能是0或1。由其运算结果可归纳为:只要有一为真则结果必为真。这个结论也可推广到多个变量,如A、B、C、D、……,各变量全为假则结果必为假,有一为真结果必为真。

从上可知,在多输入的“或”门电路中,只要其中一个输入为1,则其输出必为1;只有全部输入均为0时,输出才为0。

有时也将或运算称为“逻辑加”。当A和B为多位二进制数时,如:

$$A = A_1 A_2 A_3 \cdots A_n$$

$$B = B_1 B_2 B_3 \cdots B_n$$

在进行“逻辑或”运算时,各对应位分别进行“或”运算:

$$Y = A + B$$

$$= (A_1 + B_1)(A_2 + B_2)(A_3 + B_3) \cdots (A_n + B_n)$$

【例 3-2】 设 $A=10101B$, $B=11011B$, 求: $Y=A+B$ 。

【解】 $Y = A + B$

$$= (1+1)(0+1)(1+0)(0+1)(1+1)$$

$$= 11111$$

写成竖式则为:

$$\begin{array}{r} 10101 \\ +) 11011 \\ \hline 11111 \end{array}$$

注意,此处不是一般的加法运算,而是逻辑或运算。1“或”1等于1,是没有进位的。

由此可见,用“0”和一个数位相“或”,就是将此数位“保存”下来;用“1”和一个数位相“或”,就是将其置1。

3.1.3 “非”运算

“非”运算又称逻辑取反或逻辑反运算。假设一件事物的性质为A,则其经过“非”运算之后,其性质必与A相反,其表达式为:

$$L = \bar{A}$$

这实际上也是反相器的性质。所以在电路实现上,反相器是非运算的基本元件。

其基本运算规则为: $\bar{1}=0$ $\bar{0}=1$ $\overline{\bar{1}}=1$ $\overline{\bar{0}}=0$ $\overline{\bar{A}}=A$

当A为多位数时,如:

$$A = A_1 A_2 A_3 \cdots A_n$$

则其“逻辑非”为: $Y = \bar{A}_1 \bar{A}_2 \bar{A}_3 \cdots \bar{A}_n$

【例 3-3】 设 $A = 10100000B$, 求 $Y = \bar{A}$ 。

【解】 $Y = 01011111B$

3.1.4 逻辑代数的基本运算法则

与一般代数一样,逻辑代数也有类似的运算法则,如交换律、结合律、分配律,而且它们与普通代数的规律完全相同。其具体法则如下。

(1) 交换律: $A \cdot B = B \cdot A$

$$A + B = B + A$$

(2) 结合律: $(AB)C = A(BC) = ABC$

$$(A+B)+C = A+(B+C) = A+B+C$$

(3) 分配律: $A(B+C) = AB+AC$

$$(A+B)(C+D) = AC+AD+BC+BD$$

(4) 吸收律: $A+AB = A(1+B) = A$

$$A \cdot (A+B) = A \cdot A + AB$$

$$= A + AB$$

$$= A$$

$$(A+B)(A+C) = A \cdot A + AC + BA + BC$$

$$= A + AC + AB + BC$$

$$= A + AB + BC$$

$$= A + BC$$

(5) 消去律: $A + \bar{A}B = A(1+B) + \bar{A}B = A + (A + \bar{A})B = A + B$

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

(6) 反演律: $\overline{A+B} = \bar{A} \cdot \bar{B}$

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

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

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

【例 3-4】 化简逻辑代数式： $Y=AB+\bar{A}C+BC$ 。

【解】 此处需要配方，具体步骤如下：

$$\begin{aligned} Y &= AB + \bar{A}C + BC \\ &= AB + \bar{A}C + BC(A + \bar{A}) \\ &= AB + \bar{A}C + ABC + \bar{A}BC \\ &= AB(1 + C) + \bar{A}C(1 + B) \\ &= AB + \bar{A}C \end{aligned}$$

【例 3-5】 化简逻辑代数式： $Y=A\bar{B}C+\bar{A}+B+\bar{C}$ 。

【解】 化简的具体步骤如下：

$$\begin{aligned} Y &= A\bar{B}C + \bar{A} + B + \bar{C} \\ &= A\bar{B}C + \overline{\bar{A}BC} \\ &= 1 \end{aligned}$$

3.2 基本逻辑电路

逻辑代数中的各种逻辑运算均可以通过对各种基本逻辑门的组合实现。“门”是这样的一种电路：它规定各个输入信号之间满足某种逻辑关系时，才有信号输出。从逻辑关系看，门电路的输入端或输出端只有两种状态，低电平为“0”，高电平为“1”，称为正逻辑；反之，如果规定高电平为“0”，低电平为“1”，称为负逻辑。然而，高与低是相对的。本书均采用正逻辑。

很多复杂的逻辑运算都可以通过基本的逻辑运算“与”、“或”、“非”来实现。实现这 3 种逻辑运算的电路是最基本的 3 种逻辑门电路：与门电路、或门电路和非门电路。通过组合这 3 个基本门电路，可实现更复杂的逻辑电路，如与非门电路、或非门电路、异或门电路和同或门电路等，分别用于完成与非、或非、异或和同或等逻辑运算功能。

3.2.1 与门电路

实现逻辑运算“与”的电路称为与门电路。与门电路的逻辑符号如图 3-1 所示，其中 A、B 是输入信号，Y 是输出信号。

与门电路的逻辑表达式为：

$$Y=A \cdot B$$

与门电路的逻辑真值表如表 3-1 所示。

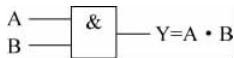


图 3-1 与门的逻辑符号

表 3-1 与门的逻辑真值表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

3.2.2 或门电路

实现逻辑运算“或”的电路称为或门电路。或门电路的逻辑符号如图 3-2 所示,其中 A、B 是输入信号,Y 是输出信号。

或门电路的逻辑表达式为:

$$Y = A + B$$

或门电路的逻辑真值表如表 3-2 所示。

表 3-2 或门的逻辑真值表



图 3-2 或门的逻辑符号

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

3.2.3 非门电路

实现逻辑运算“非”的电路称为非门电路。非门电路的逻辑符号如图 3-3 所示,其中 A 是输入信号,Y 是输出信号。

非门电路的逻辑表达式为:

$$Y = \bar{A}$$

由于输入和输出总是相反,故非门电路又称为反相器。

非门电路的逻辑真值表如表 3-3 所示。

表 3-3 非门的逻辑真值表

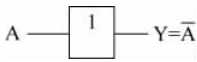


图 3-3 非门的逻辑符号

A	Y
0	1
1	0

3.2.4 与非门电路

实现逻辑运算“与非”的电路称为与非门电路。与非门电路是与门和非门相结合形成。与非门电路的逻辑符号如图 3-4 所示,其中 A、B 是输入信号,Y 是输出信号。

与非门电路的逻辑表达式为:

$$Y = \overline{A \cdot B}$$

其运算规则为:先对 A 和 B 进行与运算,再对与运算后的结果进行非运算。

与非门电路的逻辑真值表如表 3-4 所示。

表 3-4 与非门的逻辑真值表



图 3-4 与非门的逻辑符号

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

3.2.5 或非门电路

实现逻辑运算“或非”的电路称为或非门电路。或非门电路是或门和非门相结合形成的。或非门电路的逻辑符号如图 3-5 所示,其中 A、B 是输入信号,Y 是输出信号。

或非门电路的逻辑表达式为:

$$Y = \overline{A+B}$$

其运算规则为:先对 A 和 B 进行或运算,再对或运算后的结果进行非运算。

或非门电路的逻辑真值表如表 3-5 所示。

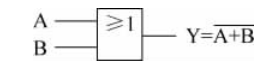


图 3-5 或非门的逻辑符号

表 3-5 或非门的逻辑真值表

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

3.2.6 异或门电路

实现逻辑运算“异或”的电路称为异或门电路。异或门电路的逻辑符号如图 3-6 所示,其中 A、B 是输入信号,Y 是输出信号。

异或门电路的逻辑表达式为:

$$Y = A \oplus B \quad \text{或} \quad Y = \bar{A}B + A\bar{B}$$

其运算规则为:两个逻辑变量取值不相同,它们“异或”的结果为 1;两个逻辑变量取值相同时,它们“异或”的结果为 0。其运算规则可总结为:相同为 0,相异为 1。

异或门电路的逻辑真值表如表 3-6 所示。

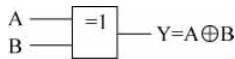


图 3-6 异或门的逻辑符号

表 3-6 异或门的逻辑真值表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

3.2.7 同或门电路

实现逻辑运算“同或”的电路称为同或门电路。同或门电路的逻辑符号如图 3-7 所示,其中 A、B 是输入信号,Y 是输出信号。

同或门电路的逻辑表达式为:

$$Y = A \odot B \quad \text{或} \quad Y = \bar{A}\bar{B} + AB$$

其运算规则为:两个逻辑变量取值相同时,它们“同或”的结果为 1;两个逻辑变量取值不同时,它们“同或”的结果为 0。其运算规则可总结为:相同为 1,相异为 0。

同或门电路的逻辑真值表如表 3-7 所示。

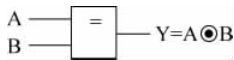


图 3-7 同或门的逻辑符号

表 3-7 同或门的逻辑真值表

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

3.3 二进制数的加、减法电路

常见的算术运算只有加、减、乘、除 4 种。为了使微型计算机中硬件结构简单、成本较低，在计算机中常常只采用加法电路来实现计算机的运算。

3.3.1 二进制数的加法运算

由于补码的引入，在计算机中采用同一个逻辑部件来完成加法和减法运算。为了更好地理解采用补码运算的二进制加法/减法器，首先应理解半加和全加的概念。二进制具体运算如下所示。

【例 3-6】

(1)

$$\begin{array}{r} 1 \quad A \\ +) 1 \quad B \\ \hline 10 \quad S \end{array}$$

(2)

$$\begin{array}{r} 01 \quad A \\ +) 10 \quad B \\ \hline 11 \quad S \end{array}$$

(3)

$$\begin{array}{r} \overline{11} \quad C \\ 11 \quad A \\ +) 11 \quad B \\ \hline 110 \quad S \end{array}$$

(4)

$$\begin{array}{r} \overline{11} \quad \overline{11} \quad C \\ 011 \quad A \\ +) 011 \quad B \\ \hline 110 \quad S \end{array}$$

在例 3-6(1)中，加数 A 和被加数 B 都是 1 位数，其和 S 变成 2 位数，这是因为相加结果产生进位。

在例 3-6(2)中，A 和 B 都是 2 位数，相加结果 S 也是 2 位数，这是因为相加结果没有产生进位。

在例 3-6(3)中，A 和 B 都是 2 位数，相加结果 S 是 3 位数，这也是因为相加结果产生了进位。

在例 3-6(4)中，A 和 B 都是 3 位数，C 为低位向高位的进位。

由上可得：

① 两个二进制数相加时，可以逐位相加。如二进制数可以写成：

$$A = A_3 A_2 A_1 A_0$$

$$B = B_3 B_2 B_1 B_0$$

则从最右边第 1 位开始，逐位相加，其结果可以写成：

$$S = S_3 S_2 S_1 S_0$$

其中各位是分别求出的：

$$\begin{aligned} S_0 &= A_0 + B_0 \rightarrow \text{进位 } C_1 \\ S_1 &= A_1 + B_1 + C_1 \rightarrow \text{进位 } C_2 \\ S_2 &= A_2 + B_2 + C_2 \rightarrow \text{进位 } C_3 \\ S_3 &= A_3 + B_3 + C_3 \rightarrow \text{进位 } C_4 \end{aligned}$$

最后所得的和是：

$$A+B=C_4S_3S_2S_1S_0$$

② 右边第 1 位相加的电路要求：

输入量为两个，即 A_0 及 B_0 ；

输出量为两个，即 S_0 及 C_1 。

这样的二进制位相加的电路称为半加器（不考虑进位输入的相加）。

③ 从右边第 2 位开始，各位可以对应相加。各位对应相加时的电路要求：

输入量为 3 个，即 A_i 、 B_i 、 C_i ；

输出量为两个，即 S_i 、 C_{i+1} 。

其中 $i=1, 2, 3, \dots, n$ 。这样的二进制位相加的电路称为全加器（考虑低位的进位）。

3.3.2 半加器

半加器是用于逻辑变量相加的逻辑电路，它可以实现两个变量相加操作，是加法器的一种。半加器只有两个输入端，用以代表两个数字（ A_0 、 B_0 ）的电位输入；有两个输出端，用以输出总和 S_0 及进位 C_1 。

半加器的真值表如图 3-8(a) 所示。

考察一下 C_1 与 A_0 及 B_0 的关系，即可看出这是“与”的关系，即：

$$C_1 = A_0 \cdot B_0$$

再看一下 S_0 与 A_0 及 B_0 的关系，也可看出这是“异或”的关系，即：

$$S_0 = A_0 \oplus B_0 \quad \text{或} \quad S_0 = \bar{A}_0 B_0 + A_0 \bar{B}_0$$

即只有当 A_0 及 B_0 二者相异时，其结果为 1；二者相同时，其结果为 0。因此，可以用“与门”及“异或门”来实现真值表的要求。图 3-8(a) 和 3-8(b) 就是这个真值表及半加器的电路图。其符号如图 3-8(c) 所示。从上可以看出：半加器可以实现二进制数最低位的相加操作。

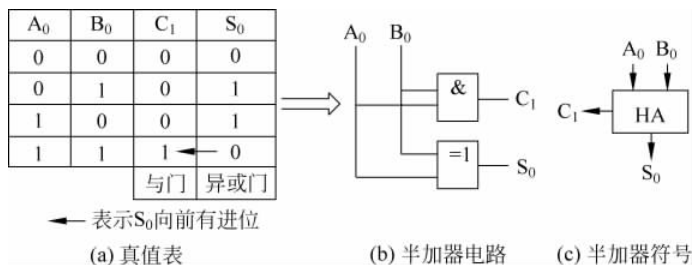


图 3-8 半加器的真值表、电路及符号

3.3.3 全加器

全加器也是用于逻辑变量相加的逻辑电路，它可以实现 3 个变量相加操作，也是加法器的

一种。全加器有 3 个输入端 A_i 、 B_i 和 C_i ；有两个输出端 S_i 和 C_{i+1} 。其真值表如图 3-9 所示，符号如图 3-10 所示。由此真值表可知，其总和 S_i 可用“异或门”来实现，即 $S_i = A_i \oplus B_i \oplus C_i$ ；而其进位 C_{i+1} 则可以用 3 个“与门”和一个“或门”来实现，即 $C_{i+1} = A_i B_i + A_i C_i + B_i C_i$ ；其电路图如图 3-9 所示。从上可以看出：全加器可以实现二进制数任何一位的相加操作。

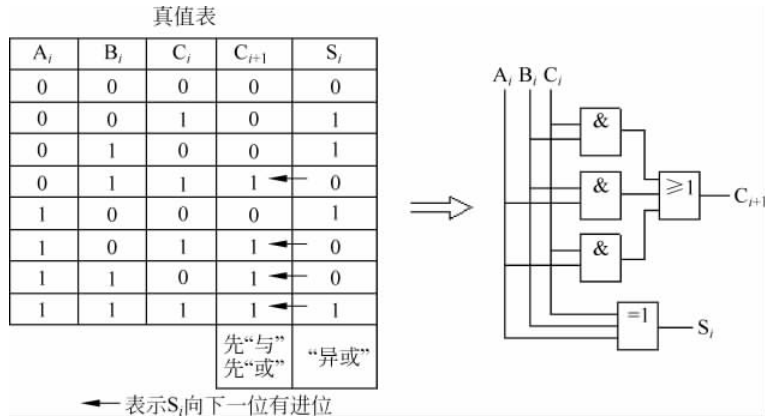


图 3-9 全加器的真值表及电路图

3.3.4 二进制数的加法电路

设 $A = 1010B = 10$, $B = 1011B = 11$, A 与 B 相加，写成竖式算法如下：

$$\begin{array}{r}
 1010 \quad A \\
 +) 1011 \quad B \\
 \hline
 10101 \quad S
 \end{array}$$

即其相加结果为 $S = 10101B = 21$ 。 A 与 B 相加的加法电路如图 3-11 所示。

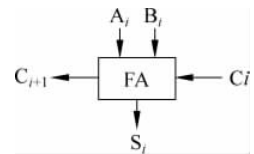


图 3-10 全加器符号

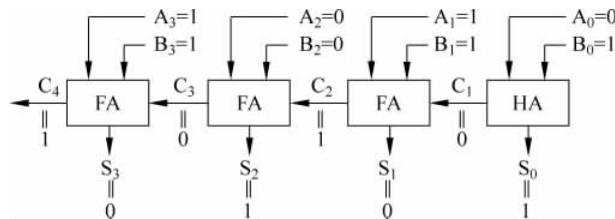


图 3-11 4 位二进制数加法电路

从加法电路，可看到同样的结果：

$$S = C_4 S_3 S_2 S_1 S_0 = 10101B = 21$$

3.3.5 二进制数的减法运算

在计算机中，没有专用的减法器，而是将减法运算转换为加法运算。其原理是：

$$[A - B]_{补} = [A]_{补} + [-B]_{补}$$

其中，已知 $[B]_{补}$ 求 $[-B]_{补}$ 的方法是将 $[B]_{补}$ 各位按位取反(包括符号位)末位加 1。

这个公式说明：要计算 $A - B$ ，可以先计算 A 的补码与 $-B$ 的补码(如有进位，则舍去进

位),这个和数就是 $A-B$ 的补码,只需将这个和转换为原码,即可得到 A 与 B 两数之差。

关于补码的概念,前面已作介绍,此处就不再介绍了。

其具体的运算过程如下例所示。

【例 3-7】 已知 $A=7, B=4$ (假设机器字长为 4 位),求: $Y=A-B$ 。

【解】 因为 A 和 B 均为正数,所以

$$[A]_{\text{补}}=[A]_{\text{原}}=0111\text{B} \quad [B]_{\text{补}}=[B]_{\text{原}}=0100\text{B}$$

于是

$$\begin{aligned} [-B]_{\text{补}} &= 1100\text{B} \\ [A]_{\text{补}} + [-B]_{\text{补}} &= 0111\text{B} + 1100\text{B} \\ &= 1\ 0011\text{B} \\ &\quad \uparrow \text{——进位,舍去} \\ &= 0011\text{B} = [A-B]_{\text{补}} \end{aligned}$$

因为 0011B 为正数,所以 $[A-B]_{\text{补}}=[A-B]_{\text{原}}$,可得 $A-B=3$ 。

【例 3-8】 已知 $A=-5\text{H}, B=-2\text{H}$,求: $Y=A-B$ 。

【解】 因为

$$\begin{aligned} [A]_{\text{原}} &= 1101\text{B} & [A]_{\text{补}} &= 1011\text{B} \\ [B]_{\text{原}} &= 1010\text{B} & [B]_{\text{补}} &= 1110\text{B} & [-B]_{\text{补}} &= 0010\text{B} \end{aligned}$$

所以

$$\begin{aligned} [A]_{\text{补}} + [-B]_{\text{补}} &= 1011\text{B} + 0010\text{B} \\ &= 1101\text{B} = [A-B]_{\text{补}} \end{aligned}$$

因为 1101B 为负数,所以由 $[A-B]_{\text{补}}=1101\text{B}$ 得 $[A-B]_{\text{原}}=1011\text{B}$,可得 $A-B=-3$ 。

3.3.6 可控反相器及加、减法电路

由于二进制补码运算可将减法运算转换为加法运算,因此需要用一电路来实现 $[B]_{\text{补}}$ 转换为 $[-B]_{\text{补}}$,即一个二进制补码各位按位取反(包括符号位)末位加 1。

图 3-12 为可控反相器,它能有控制地按位取反。这实际上是一个异或门,两输入端的异或门的特点是:二者相同则输出为 0,二者不同则输出为 1。

若将 SUB 端看作控制端,当在 SUB 端加上低电位时, Y 端的电平和 B_0 端的电平相同;在 SUB 端加上高电平时, Y 端的电平和 B_0 端的电平相反。

即:当 $\text{SUB}=0$ 时, $Y=B_0$; 当 $\text{SUB}=1$ 时, $Y=\bar{B}_0$ 。

利用这个特点,在图 3-11 的 4 位二进制数加法电路上增加 4 个可控反相器并将最低位的半加器也改用全加器,就可以得到如图 3-13 所示的 4 位二进制数加法器/减法器电路了,因为这个电路既可以作为加法器电路(当 $\text{SUB}=0$),又可以作为减法器电路(当 $\text{SUB}=1$)。

设有下面两个二进制数:

$$A = A_3 A_2 A_1 A_0$$

$$B = B_3 B_2 B_1 B_0$$

则可将这两个数的各位分别送入该电路的对应端,于是:

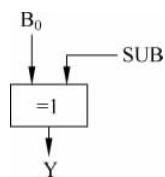


图 3-12 可控反相器

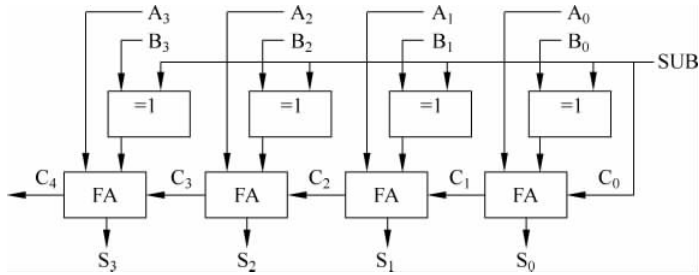


图 3-13 二进制补码加法器/减法器

当 SUB=0 时,电路作加法运算: $A+B$ 。

当 SUB=1 时,电路作减法运算: $A-B$ 。

当 SUB=0 时,各位的可控反相器的输出与 B 的各位同相,所以图 3-13 和图 3-11 的原理完全一样,各位均按位相加。结果 $S=S_3S_2S_1S_0$,而其和为 $C_4S_3S_2S_1S_0$ 。

当 SUB=1 时,各位的反相器的输出与 B 的各位反相。注意,最右边第一位(即 S_0 位)也是用全加器,其进位输入端与 SUB 端相连,实现加 1 操作。即 $C_0=SUB=1$ 。所以此位相加即为:

$$A_0 + \bar{B}_0 + 1$$

其他各位为:

$$A_1 + \bar{B}_1 + C_1$$

$$A_2 + \bar{B}_2 + C_2$$

$$A_3 + \bar{B}_3 + C_3$$

因此其总和输出 $S=S_3S_2S_1S_0$,即:

$$\begin{aligned} S &= A_3A_2A_1A_0 + \bar{B}_3\bar{B}_2\bar{B}_1\bar{B}_0 + 1 \\ &= A + \bar{B} + 1 \\ &= [A]_{补} + [-B]_{补} \\ &= [A - B]_{补} \end{aligned}$$

当然,此时若 C_4 不等于 0,则要被舍去。

注意: 在图 3-13 中,最低位的相加不能用半加器,只能用全加器。读者可以思考其缘由。

3.4 算术逻辑单元

算术逻辑单元简称 ALU,它是功能较强的组合逻辑电路,既能进行算术运算,又能进行逻辑运算,是计算机运算器的核心部件。算术逻辑单元的基本逻辑结构是超前进位加法器。算术逻辑单元的逻辑符号如图 3-14 所示。其中,ALU 的功能受功能控制信号的控制。

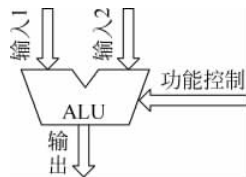


图 3-14 ALU 的逻辑符

3.5 触 发 器

触发器是一种能记忆机器前一输入状态的存放二进制代码的单元电路,是计算机中各种逻辑电路的基础。它具有两个稳定状态,但在任何时刻触发器只处于一种稳定状态,稳定状态的翻转只发生在脉冲到来的时候。触发器可以组成寄存器、计数器、运算器、分配器、译码器等,寄存器又可以组成存储器。

触发器一般用晶体管元件构成。简单触发器可以由两个晶体管组成的对称电路来构成,在复杂的触发电路中则有单稳态触发电路和双稳态触发电路,这里就不对这些电路的原理图和工作特点作一一介绍了。

触发器的种类很多,在计算机中最常见的触发器有 RS 触发器、D 触发器和 JK 触发器。下面就对其作一简单介绍。

3.5.1 R-S 触发器

R-S 触发器是最基本、最简单的触发器,可以用两个与非门来组成,也可以用两个或非门来组成,常见的 R-S 触发器是用两个与非门来组成。R-S 触发器还分为低电平触发和高电平触发,常见的 R-S 触发器是低电平触发。下面就以两个与非门组成的低电平触发的 R-S 触发器为例,介绍一下 R-S 触发器的功能。

两个与非门组成的低电平触发的 R-S 触发器的逻辑电路图如图 3-15 所示,其逻辑符号如图 3-16 所示。其中,S 端为置位端(置 1 端);R 端为复位端(置 0 端);Q 端为状态输出端; \bar{Q} 端为与状态相反的输出端。其工作原理如下所示:

- 当 $S=1$ 而 $R=0$ 时, $Q=0(\bar{Q}=1)$ 称为复位;
- 当 $S=0$ 而 $R=1$ 时, $Q=1(\bar{Q}=0)$ 称为置位;
- 当 $S=1$ 而 $R=1$ 时,Q 将保持前一状态不变;
- 当 $S=0$ 而 $R=0$ 时,Q 的状态不定。

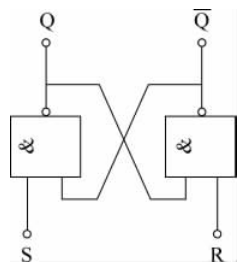


图 3-15 R-S 触发器

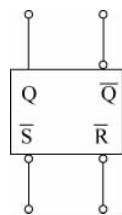


图 3-16 R-S 触发器的符号

其真值表如表 3-8 所示。

表 3-8 R-S 触发器的真值表

R	S	Q
1	0	1
0	1	0
1	1	保持前一状态不变
0	0	不定

3.5.2 D 触发器

D 触发器又称数据触发器,在计算机中使用最为广泛,主要用于存放数据。D 触发器分为上升沿触发和下降沿触发,下面就以上升沿触发为例介绍 D 触发器的工作原理。D 触发器的逻辑符号如图 3-17 所示,真值表如表 3-9 所示。

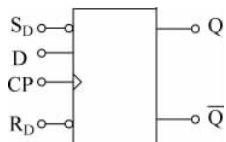


图 3-17 D 触发器的符号

表 3-9 D 触发器的真值表

CP	Q
0→1(上升沿)	Q=D
其他情况	保持前一状态不变

图 3-17 中, S_D为置 1 端, R_D为置 0 端, D 为同步输入端, CP 为时钟脉冲输入端, Q 为 D 触发器的状态输出端。D 触发器的状态由时钟脉冲上升沿到来时 D 端的状态决定, 当 D 端为 1 (高电位) 时, 触发器的状态为 1 (Q=1); 当 D 端为 0 (低电位) 时, 触发器的状态为 0 (Q=0)。时钟脉冲上升沿到来之前, 触发器保持前一状态不变。

3.5.3 JK 触发器

JK 触发器是组成计数器的理想记忆元件。JK 触发器只是在 RS 触发器前面增加两个与门, 并从输出端到输入端(与门的输入端)作交叉反馈即可, JK 触发器克服了 RS 触发器中存在的状态不稳定的缺点。其逻辑图如图 3-18 所示, 真值表如表 3-10 所示。

其中, S_D端为置位端, R_D为复位端, Q 为触发器状态的输出端。

当 J=0, K=0 时, CP 脉冲不会改变触发器的状态;

当 J=0, K=1 时, CP 脉冲使触发器为 0 状态;

当 J=1, K=0 时, CP 脉冲使触发器为 1 状态;

当 J=1, K=1 时, CP 脉冲使触发器的状态翻转。

后面章节的计数器就是利用 JK 触发器的翻转特性构成的。

表 3-10 JK 触发器的真值表

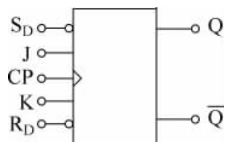


图 3-18 JK 触发器的符号

CP	J	K	Q	动作
×	0	0	保持原状态	自锁状态
0→1	0	1	0	复位
0→1	1	0	1	置位
0→1	1	1	原状态的反码	翻转

3.6 寄存器

所谓寄存器,是指寄存(存放)一个数字或指令(用二进制表示)的逻辑部件。寄存器也是由触发器构成的。每一个触发器都有两个相反的 0 或 1 状态,故一个触发器可以表示一个二进制数位。因此,一个触发器就是一个一位寄存器。一个 n 位寄存器可以由 n 个触发器组成。寄存器由于其在计算机中的作用不同而具有不同的功能,按照功能可将寄存器分为缓冲寄存

器、移位寄存器、计数器、累加器等。

下面就分别介绍这些寄存器的工作原理及其电路图。

3.6.1 缓冲寄存器

所谓缓冲寄存器,就是用以暂存某个数据,以便在适当的时间节拍和给定的计算步骤将数据输入或输出到其他记忆元件中去的部件。图 3-19 是一个 4 位缓冲寄存器的电路原理图。

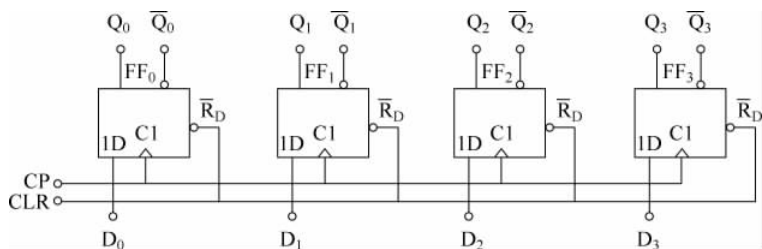


图 3-19 4 位缓冲寄存器电路原理图

此缓冲寄存器是由 4 个上升沿触发的 D 触发器组成。设有一个 4 位二进制数 $D = D_3D_2D_1D_0$ 要存到这个缓冲寄存器中去,其基本工作原理为:将 D_0 、 D_1 、 D_2 、 D_3 分别送到各个触发器的 D 端去,若 CP 脉冲的上升沿还未到来,则各个 D 触发器保持其原状态不变;只有当 CP 脉冲的上升沿到来时,各个 D 触发器的输入端(D 端)才接受 D_0 、 D_1 、 D_2 、 D_3 的影响,而变成: $Q_0 = D_0$ 、 $Q_1 = D_1$ 、 $Q_2 = D_2$ 、 $Q_3 = D_3$,结果就是: $Q = Q_3Q_2Q_1Q_0 = D_3D_2D_1D_0 = D$,这就将数据 D 装到缓冲寄存器中去了;如要将此数据送至其他记忆元件中,则可由 Q_0 、 Q_1 、 Q_2 、 Q_3 各条引线引出去。

3.6.2 移位寄存器

移位寄存器能将所存储的数据逐位向左或向右移动,以实现计算机在运行过程中所需的算术运算、逻辑运算等功能,例如加减运算前的移位、最左边位的 0 或 1 判断等。根据其功能可将移位寄存器分为左移寄存器和右移寄存器,其电路原理图如图 3-20 所示。

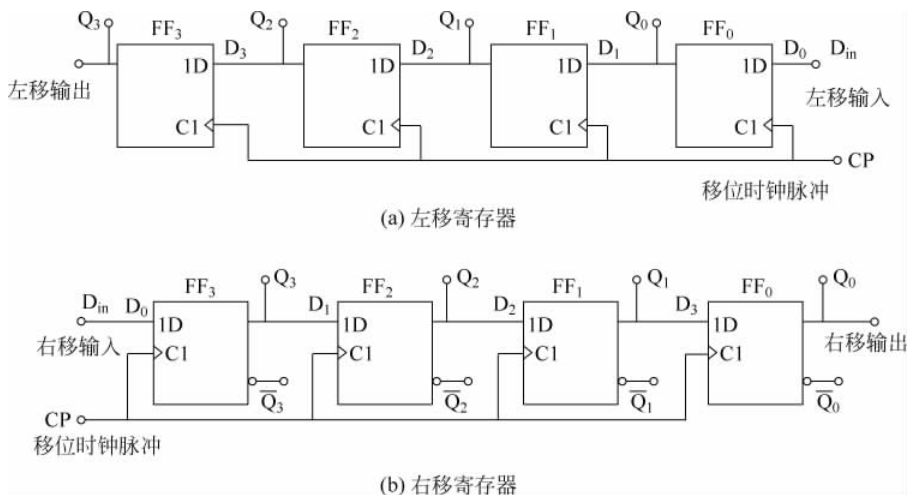


图 3-20 移位寄存器简化原理

左移寄存器如图 3-20(a)所示,当 $D_{in}=1$ 送至最右边的第 1 位时, $D_0=1$,当 CP 的上升沿到达时, $Q_0=1$ 。同时第 2 位的 $D_1=1$ 。当 CP 第 2 个上升沿到达时, $Q_1=1$ 。其左移过程为:

CP 上升沿未到 $Q=Q_3Q_2Q_1Q_0=0000$

第 1 个上升沿来到 $Q=0001$

第 2 个上升沿来到 $Q=0011$

第 3 个上升沿来到 $Q=0111$

第 4 个上升沿来到 $Q=1111$

第 5 个上升沿来到,如此时 D_{in} 仍为 1,则 Q 不变,仍为 1111。当 $Q=1111$ 之后,改变 D_{in} ,使 $D_{in}=0$,则结果将是把 0 逐位左移。其具体过程如下:

第 1 个上升沿来到 $Q=Q_3Q_2Q_1Q_0=1110$

第 2 个上升沿来到 $Q=1100$

第 3 个上升沿来到 $Q=1000$

第 4 个上升沿来到 $Q=0000$

由此可见,在左移寄存器中,每个时钟脉冲都要把所存储的各位向左移动一个数位。

右移寄存器如图 3-20(b)所示。图 3-20(b)与图 3-20(a)的差别仅在于各位的接法不同,而且输入数据 D_{in} 是加到左边第 1 位的输入端 D_3 。根据上面的分析,当 $D_{in}=1$ 时,随着时钟脉冲而逐步位移过程如下:

CP 上升沿未到 $Q=Q_3Q_2Q_1Q_0=0000$

第 1 个上升沿来到 $Q=1000$

第 2 个上升沿来到 $Q=1100$

第 3 个上升沿来到 $Q=1110$

第 4 个上升沿来到 $Q=1111$

由此可见,在右移寄存器中,每个时钟脉冲都要把所存储的各位向右移动一个数位。

3.6.3 计数器

计数器是指能对输入信号(脉冲)进行加或减运算的装置,是由若干个触发器和一些控制门组成的逻辑部件。它的特点是能够把存储在其中的数字加 1 或减 1。

计数器的种类很多,按构成计数器的触发器的翻转次序分类,可将计数器分为同步计数器和异步计数器;按计数过程中计数器中数的增减变化分类,可将计数器分为加法计数器、减法计数器和加减计数器;按计数器中数的编码方式分类,可将计数器分为二进制计数器和十进制计数器;还有行波计数器、环形计数器和程序计数器等。

① 当计数脉冲到达时,若组成计数器的所有触发器同时发生翻转,则为同步计数器;若各级触发器不是同时发生翻转,则为异步计数器。

② 行波计数器的工作原理是:在时钟边缘到来时开始计数,由右边第一位开始,如有进位则要一位一位地推进。图 3-21 就是由 JK 触发器组成的行波计数器的工作原理图。其特点是:第 1 个时钟脉冲促使其最低有效位加 1,由 0 变 1;第 2 个时钟脉冲促使其最低有效位由 1 变 0,同时推动第 2 位,使其由 0 变 1;同理,第 2 位由 1 变 0 时又去推动第 3 位,使其由 0 变 1,这样有如水波前进一样逐位进位下去。

图 3-21 中的各位的 J、K 输入端都是悬浮的,这相当于 J、K 端都是置 1 的状态。只要时钟脉冲边缘一到,最右边的触发器就会翻转,即 Q 由 0 转为 1 或由 1 转为 0。各位的 JK 触发器

的时钟脉冲输入端都带有一个“o”，这表示是串有一个反相门(非门)，这样，只有时钟脉冲的后沿才能为其所接受。其计数过程如下：

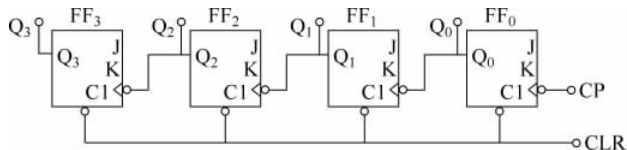


图 3-21 行波计数器的工作原理图

开始时使 CLR 由高电位变至低电位，计数器全部清除，即 $Q = Q_3 Q_2 Q_1 Q_0 = 0000$ ，当第 1 个时钟后沿到达时， $Q = 0001$ 。 Q_0 由低电位(0)升高至高电位(1)，产生的是电位上升的变化，由于有“0”在第 2 位的时钟脉冲输入端，所以第 2 个触发器不会翻转，必须在 Q_0 由 1 降为 0 时才会翻转。接着：

第 2 个时钟后沿到	$Q = 0010$
第 3 个时钟后沿到	$Q = 0011$
第 4 个时钟后沿到	$Q = 0100$
第 5 个时钟后沿到	$Q = 0101$
第 6 个时钟后沿到	$Q = 0110$
第 7 个时钟后沿到	$Q = 0111$
第 8 个时钟后沿到	$Q = 1000$
.....
第 15 个时钟后沿到	$Q = 1111$
第 16 个时钟后沿到	$Q = 0000$
第 17 个时钟后沿到	$Q = 0001$

在第 16 个时钟脉冲到达时，计数器复位至 0，因此这个计数器可以计 0~15 的数。如果要计的数更多，就需要更多的位，即由更多的 JK 触发器来组成计数器（一个触发器表示一个二进制数位）。如 8 位计数器可由 8 个触发器构成，可计 0~255 的数；12 位计数器可由 12 个触发器构成，可计 0~4095 的数。

③ 环形计数器也是由若干个触发器组成的。不过，环形计数器与上述计数器不一样，它仅有唯一的一个位为高电位，即只有一位为 1，其他各位为 0。图 3-22 就是由 D 触发器组成的环形计数器的电路原理图。

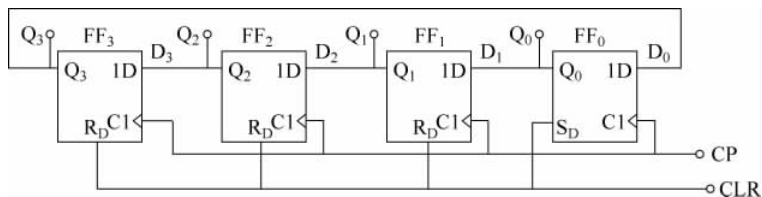


图 3-22 环形计数器的电路原理图

当 CLR 端有高电位输入时，除右边第 1 位外，其他各位全被置 0（因清除电位 CLR 都接至它们的 R_D 端），而右边第 1 位则被置 1（因清除电位 CLR 被引至其 S_D 端）。这就是说，开始时 $Q_0 = 1$ ，而 Q_1, Q_2, Q_3 全为 0。因此， $D_1 = 1$ ，而 $D_0 = Q_3 = 0$ 。在时钟脉冲正边缘到来时， $Q_0 = 0$ ，

$Q_1=1$,其他各位仍为0。第2个时钟脉冲前沿到来时, $Q_0=0,Q_1=0$,而 $Q_2=1,Q_3=0$ 。这样,随着时钟脉冲各位轮流置1,并且是在最后一位(左边第1位)置1之后又回到右边第1位,这就形成环形置位,所以称为环形计数器。

环形计数器不是用来计数,而是用来发出顺序控制信号的,它是计算机控制器中的一个很重要的部件。

3.6.4 累加器

累加器是一个由多个触发器组成的多位寄存器,也可以说是一种暂存器,用来存储参加运算的操作数或计算所产生的中间结果。若没有像累加器这样的暂存器,那么在每次计算(如加法、乘法、移位等)后就必须把结果写回到内存,然后再读回来。然而从内存中读取的速度比从累加器读取更慢。这种特殊的寄存器在微型计算机的数据处理中担负着重要的角色。

3.7 二进制译码器

译码器是将一种代码转换成另一种代码的逻辑电路。译码器的应用非常广泛,如地址译码器、用二进制译码器实现码制变换等。译码器又分为“1”选中和“0”选中。下面就以“1”选中为例,对译码器作一介绍。其工作原理为:设二进制译码器的输入端为 n 个,则输出端为 2^n 个,且对应于输入代码的每一种状态, 2^n 个输出中只有一个为1,其余全为0。下面以3-8译码器为例,其电路图如图3-23所示。3-8译码器的逻辑表达式可写为:

$$\begin{cases} Y_0 = \bar{A}_2 \bar{A}_1 \bar{A}_0 \\ Y_1 = \bar{A}_2 \bar{A}_1 A_0 \\ Y_2 = \bar{A}_2 A_1 \bar{A}_0 \\ Y_3 = \bar{A}_2 A_1 A_0 \\ Y_4 = A_2 \bar{A}_1 \bar{A}_0 \\ Y_5 = A_2 \bar{A}_1 A_0 \\ Y_6 = A_2 A_1 \bar{A}_0 \\ Y_7 = A_2 A_1 A_0 \end{cases}$$

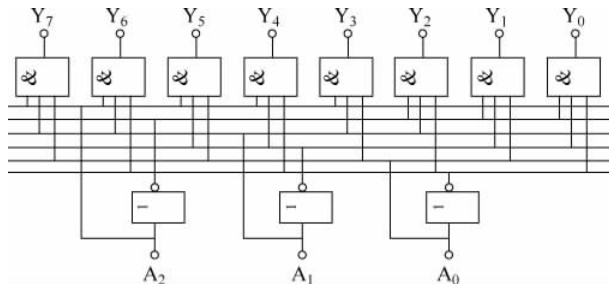


图 3-23 3-8 译码器电路图

其中, $A_2、A_1、A_0$ 为二进制译码输入端, $Y_0 \sim Y_7$ 为译码器的输出端。其真值表如表3-11所示。

表 3-11 3-8 译码器的真值表

A ₂	A ₁	A ₀	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

3.8 三态输出电路

在计算机的逻辑电路中还有一种特殊的门电路,此门电路可以有 3 种不同的输出状态,即高电平、低电平、高阻。此门电路就是三态输出电路,又称三态门。

其中,高阻状态是指悬空、悬浮状态,又称为禁止状态。测其电阻为∞,电压为 0V,但不是接地,测其电流为 0A。

三态门既可以用与非门构成,也可以用或非门等其他门电路构成。下面以两端输入三态与非输出电路为例,其逻辑符号如图 3-24 所示。

其中,A、B 为输入端,EN 为三态门的控制端,Y 为输出端。

对图 3-24 中的(a)图,其真值表如表 3-12 所示,逻辑表达式可写为:

$$\text{当 } EN=0 \text{ 时, } Y = \overline{AB}$$

$$\text{当 } EN=1 \text{ 时, } Y = \text{高阻}$$

对图 3-24 中的(b)图,其真值表如表 3-13 所示,逻辑表达式可写为:

$$\text{当 } EN=0 \text{ 时, } Y = \text{高阻}$$

$$\text{当 } EN=1 \text{ 时, } Y = \overline{AB}$$

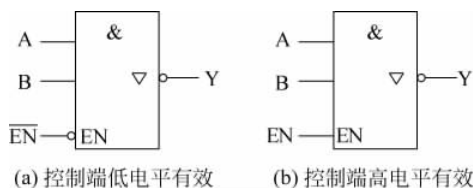


图 3-24 两端输入三态与非门逻辑符号

表 3-12 三态输出电路的逻辑表

EN	A	B	Y
0	0	0	1
	0	1	1
	1	0	1
	1	1	0
1	0	0	高阻
	0	1	
	1	0	
	1	1	

表 3-13 三态输出电路的逻辑表

EN	A	B	Y
0	0	0	高阻
	0	1	
	1	0	
	1	1	
1	0	0	1
	0	1	1
	1	0	1
	1	1	0

习 题 3

一、选择题

- 下列逻辑运算中,正确的是()。
 - $1 \cdot 1=1$
 - $0 \cdot 1=1$
 - $1 \cdot 0=1$
 - $1+1=2$
- 设与非门的输入变量为 A、B,输出变量为 Y,则当 A、B 分别为()时, $Y=0$ 。
 - 0、0
 - 1、1
 - 0、1
 - 1、0
- 设异或门的输入变量为 A、B,输出变量为 Y,则当 A、B 分别为()时, $Y=1$ 。
 - 0、0
 - 1、1
 - 0、1
 - 以上都不对
- 设或非门的输入变量为 A、B,输出变量为 Y,则当 A、B 分别为()时, $Y=1$ 。
 - 0、0
 - 1、1
 - 0、1
 - 1、0
- D 触发器只有在()到来时,才能接收信息。
 - D 端信号
 - CP 脉冲
 - S 端信号
 - R 端信号
- 将一个 8 位寄存器 R 的低 4 位全部置 1,高 4 位保持不变,则下列()项逻辑运算可以实现。
 - $(R) \vee 80H \rightarrow R$
 - $(R) \vee 08H \rightarrow R$
 - $(R) \vee F0H \rightarrow R$
 - $(R) \vee 0FH \rightarrow R$
- 若 R-S 触发器当前的输出 $Q=0$,要使触发器翻转成 $Q=1$,应使 R、S 分别为()。
 - 0、0
 - 0、1
 - 1、0
 - 1、1
- 半加器有()个输入端,()个输出端。
 - 1
 - 2
 - 3
 - 4
- 全加器有()个输入端,()个输出端。
 - 1
 - 2
 - 3
 - 4
- 设四位二进制基本寄存器的初始状态为 0000,输入总是 0001,在两个脉冲到来后,该寄存器的状态为()。
 - 0000
 - 0001
 - 0010
 - 0011
- 有一个用于表示无符号整数的 8 位二进制寄存器,每位的数字均为 1,其十进制数为()。
 - 2^7-1
 - 2^8-1
 - 2^8
 - 2^8+1

二、填空题

1. 在逻辑运算中, $A+0=$ _____, $A \cdot 1=$ _____。
2. 二进制数 $A=10110110B$, $B=00001111B$, 那么执行逻辑与运算 $A \wedge B=$ _____。
3. 设有一个逻辑变量 A , 它与 0 作异或运算后, 结果为 _____; 它与 1 作异或运算后, 结果为 _____。
4. 设控制端高电平有效的两输入三态与非门的两个输入变量分别为 A 和 B , 当控制端 EN 为高电平时, 输出为 _____。
5. 为使 $R-S$ 触发器置 1 , 则输入端 R 应为 1 , S 应为 _____, 输出端 Q 为 _____。
6. 设半加器的输入变量分别为 A 和 B , 输出变量分别为 H 和 J , 其中 H 表示“和”, J 表示“进位”, 若要求 $H=0, J=1$, 则 A 应为 _____, B 应为 _____。
7. 设全加器的输入变量分别为 A 和 B , 低位进位为 J , 若要求 $A=1, B=0, J=1$, 则输出“和” H 应为 _____, 进位 J 应为 _____。
8. 八位二进制移位寄存器可由 _____ 个 D 触发器组成。
9. 某译码器的输入数码为四位二进制数, 则它应有 _____ 条相应的输出线。
10. 若某译码器的输出端有 256 个, 则它的输入端是 _____ 位的二进制数。
11. 逻辑运算 $1101 \vee 1011 \wedge 1001=$ _____。
12. 逻辑运算 $1110 \wedge (1011 + 1100)=$ _____。
13. 逻辑运算 $111 \oplus 101 \oplus 010=$ _____。
14. 逻辑运算 $101 \odot 111 \oplus 001=$ _____。
15. 将 8 位寄存器 A 的 $D_6、D_1$ 置 1 , 其他位保持不变, 可采用的逻辑运算为 _____。
16. 将 8 位寄存器 B 的 $D_6、D_1$ 置 0 , 其他位保持不变, 可采用的逻辑运算为 _____。
17. 将 8 位寄存器 C 的 $D_6、D_1$ 取反, 其他位保持不变, 可采用的逻辑运算为 _____。

三、计算题

1. 用真值表证明 $\overline{A+B} = \overline{A} \cdot \overline{B}$ 。
2. 化简逻辑函数 $Y=(A+B)(A+C)$ 。
3. 化简逻辑函数 $Y=\overline{AB} + \overline{AB} + \overline{AB}$ 。
4. 化简逻辑函数 $Y=\overline{AB} + \overline{AB} + \overline{AB}$ 。
5. 化简逻辑函数 $Y=ABC + \overline{ABC} + \overline{ABC}$ 。
6. 化简逻辑函数 $Y=\overline{ABC} + ABC$ 。

四、问答题

1. 逻辑代数与一般代数的本质区别是什么?
2. 逻辑代数有哪 3 种基本运算?
3. 试述逻辑加 $A+A=A$ 的理由。
4. 请画出与非门的逻辑符号, 并写出其逻辑表达式、真值表。
5. 请画出异或门的逻辑符号, 并写出其逻辑表达式、真值表。
6. 请画出同或门的逻辑符号, 并写出其逻辑表达式、真值表。
7. 请画出 $R-S$ 触发器的电路, 并写出其真值表。
8. 请画出 D 触发器的逻辑符号, 并写出其真值表。