

组合逻辑电路

数字逻辑电路可以分为组合逻辑电路和时序逻辑电路。所谓组合逻辑电路是指电路任意时刻的输出状态只与该时刻的输入状态有关,而与该时刻之前的状态无任何关系,即组合逻辑电路不具有记忆功能,其输出与输入的关系具有即时性。而时序逻辑电路的输出不但与当前时刻的输入状态有关,而且还与电路之前的状态有关,所以时序逻辑电路具有记忆功能,后面章节将对其进行讨论。

3.1 逻辑门电路的逻辑符号及其外部特性

逻辑门电路是实现基本逻辑运算和常用逻辑运算的数字电路,本节将介绍简单逻辑门电路和复合逻辑门电路的逻辑符号及外部特性。

3.1.1 简单逻辑门电路

1. 正负逻辑

简单逻辑门电路指实现简单逻辑关系的电路,如与门、或门及非门。在逻辑电路中,输入和输出一般用高、低电平来表示两种不同的状态,应注意的是高电平和低电平所表示的是一定的电压范围,而不是某个固定的数值。如 TTL 门电路中,约定高电平和低电平的额定值为 3V 和 0.2V,所以在 2~5V 之间应算作高电平,0~0.8V 之间算作低电平。

逻辑电路电平的高低可用逻辑“0”和“1”表示,若用逻辑“1”表示高电平,逻辑“0”表示低电平,则称该体制为正逻辑体制;反之,若用逻辑“0”表示高电平,逻辑“1”表示低电平,则称为负逻辑体制。对于同一个电路,既可用正逻辑表示,也可用负逻辑表示。如“与”逻辑的功能表见表 3-1(a),H 表示高电平,L 表示低电平,分别进行正、负逻辑赋值,得到表 3-1(b)和表 3-1(c),观察发现,正逻辑下的与门,在负逻辑下却实现或逻辑运算。

表 3-1(a) “与”逻辑的功能表

A	B	Y	A	B	Y
L	L	L	H	L	L
L	H	L	H	H	H

表 3-1(b) 正逻辑下“与”逻辑的真值表

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

表 3-1(c) 负逻辑下“与”逻辑的真值表

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

同理可知,正逻辑的或门,在负逻辑中实现与运算;正逻辑的非门,在负逻辑中仍然实现非运算。以后章节若无特别说明,均采用正逻辑体制。

2. 简单门电路的逻辑符号

(1) 与门——将实现“与”逻辑关系的电路称为与门电路。二输入端的“与”门所对应的国际符号如图 3-1 所示。

(2) 或门——将实现“或”逻辑关系的电路称为或门电路。二输入端的“或”门所对应的国际符号如图 3-2 所示。

(3) 非门——将实现“非”逻辑关系的电路称为非门电路,也称为反相器。“非”门所对应的国际符号如图 3-3 所示。

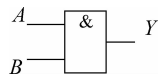


图 3-1 “与”门国际符号

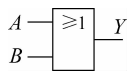


图 3-2 “或”门国际符号

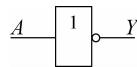
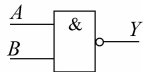


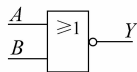
图 3-3 “非”门国际符号

3.1.2 复合逻辑门电路

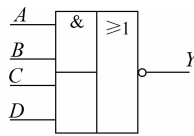
“与非”、“或非”、“与或非”、“异或”、“同或”几种复合逻辑门电路的国际符号如图 3-4 所示。



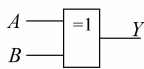
(a) “与非”门国际符号



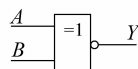
(b) “或非”门国际符号



(c) “与或非”门国际符号



(d) “异或”门国际符号



(e) “同或”门国际符号

图 3-4 复合门国际符号

以 TTL 与非门为例,给出其主要参数。

(1) 输出高电平 U_{OH} : TTL 与非门的一个或几个输入为低电平时的输出电平。产品规范值 $U_{OH} \geq 2.4V$, 标准高电平 $U_{SH} = 2.4V$ 。

(2) 高电平输出电流 I_{OH} : 输出为高电平时, 提供给外接负载的最大输出电流, 超过此值会使输出高电平下降。 I_{OH} 表示电路的拉电流负载能力。

(3) 输出低电平 U_{OL} : TTL 与非门的输入全为高电平时的输出电平。产品规范值 $U_{OL} \leq 0.4V$, 标准低电平 $U_{SL} = 0.4V$ 。

(4) 低电平输出电流 I_{OL} : 输出为低电平时, 外接负载的最大输出电流, 超过此值会使输出低电平上升。 I_{OL} 表示电路的灌电流负载能力。

(5) 扇出系数 N_O : 指一个门电路能带同类门的最大数目, 它表示门电路的带负载能力。一般 TTL 门电路 $N_O \geq 8$, 功率驱动门的 N_O 可达 25。

(6) 最大工作频率 f_{max} : 超过此频率电路就不能正常工作。

(7) 输入开门电平 U_{ON} : 是在额定负载下使与非门的输出电平达到标准低电平 U_{SL} 的输入电平。它表示使与非门开通的最小输入电平。一般 TTL 门电路的 $U_{ON} \approx 1.8V$ 。

(8) 输入关门电平 U_{OFF} : 使与非门的输出电平达到标准高电平 U_{SH} 的输入电平。它表示使与非门关断所需的最大输入电平。一般 TTL 门电路的 $U_{OFF} \approx 0.8V$ 。

(9) 高电平输入电流 I_{IH} : 输入为高电平时的输入电流, 也即当前级输出为高电平时, 本级输入电路造成的前级拉电流。

(10) 低电平输入电流 I_{IL} : 输入为低电平时的输出电流, 也即当前级输出为低电平时, 本级输入电路造成的前级灌电流。

(11) 平均传输时间 t_{pd} : 信号通过与非门时所需的平均延迟时间。在工作频率较高的数字电路中, 信号经过多级传输后造成的时间延迟, 会影响电路的逻辑功能。

(12) 空载功耗: 与非门空载时电源总电流 I_{CC} 与电源电压 V_{CC} 的乘积。

3.2 组合逻辑电路的分析

组合逻辑电路可以有一个或多个输入端, 也可以有一个或多个输出端。图 3-5 给出了组合逻辑电路的示意框图。

由此可看出, 数字信号在电路中是从输入端到输出端的单向传递, 函数表达式的形式如下式:

$$\begin{cases} Y_1 = f_1(X_1, X_2, \dots, X_n) \\ Y_2 = f_2(X_1, X_2, \dots, X_n) \\ \vdots \\ Y_m = f_m(X_1, X_2, \dots, X_n) \end{cases}$$

对组合逻辑电路的研究主要从分析、设计及运用三个方面进行:

- (1) 给出具体的组合逻辑电路, 对其进行分析并确定其功能。
- (2) 依据给出的具体要求, 设计出相关的组合逻辑电路, 完成所需功能。
- (3) 掌握常用中小规模器件的逻辑功能, 并能灵活运用于工程实践中。

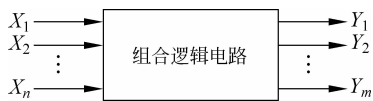


图 3-5 组合逻辑电路框图

所谓组合逻辑电路的分析,是指已知某一组合逻辑电路图,通过列出其对应的逻辑表达式、真值表,分析出该电路所完成的功能。

3.2.1 组合逻辑电路的分析步骤

给定一个组合逻辑电路图时,具体的分析步骤如下。

- (1) 根据逻辑电路图,从输入端到输出端逐级写出逻辑表达式。
- (2) 利用公式法或卡诺图法,对所得逻辑表达式进行化简,得到最简逻辑表达式。
- (3) 根据所得到的最简表达式列出真值表。
- (4) 依据真值表或最简表达式,对逻辑电路进行分析,确定其逻辑功能,并做出文字描述或是进行电路改进。

3.2.2 组合逻辑电路的分析实例

下面通过几个实例,给出对组合逻辑电路具体的分析过程。

【例 3-1】 试分析如图 3-6 所示的组合逻辑电路图,给出逻辑功能。

解: 由逻辑电路图逐级写出逻辑表达式:

$$\begin{cases} Y_1 = A \cdot B \\ Y_2 = A \cdot C \\ Y_3 = B \cdot C \\ Y = Y_1 + Y_2 + Y_3 \end{cases}$$

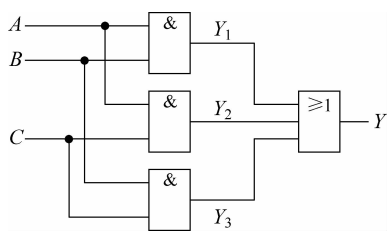


图 3-6 例 3-1 的逻辑电路图

将 Y_1 、 Y_2 、 Y_3 的表达式代入 Y 的表达式中可得: $Y = AB + AC + BC$ 。

由于表达式已经最简,所以直接列出所对应的真值表,见表 3-2。

表 3-2 例 3-1 的真值表

输 入				输出	输 入				输出
A	B	C	Y	A	B	C	Y		
0	0	0	0	1	0	0	0		
0	0	1	0	1	0	1	1		
0	1	0	0	1	1	0	1		
0	1	1	1	1	1	1	1		

由真值表可看出,只有当三个输入变量中的两个或三个取值为 1 时,输出值为 1,其他情况输出为 0,所以该电路是一个多数表决电路,即用来判断输入变量中是否多数为 1。

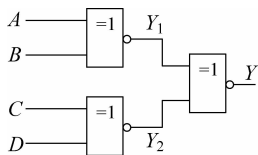


图 3-7 例 3-2 的逻辑电路图

【例 3-2】 试分析如图 3-7 所示的组合逻辑电路图,并给出逻辑功能。

解: 由电路图逐级写出逻辑表达式,并将 Y_1 和 Y_2 的表达式代入 Y 的表达式中可得:



$$\left. \begin{aligned} Y_1 &= \overline{A \oplus B} = AB + \overline{A}\overline{B} \\ Y_2 &= \overline{C \oplus D} = CD + \overline{C}\overline{D} \\ Y &= \overline{Y_1 \oplus Y_2} = Y_1Y_2 + \overline{Y_1}\overline{Y_2} \end{aligned} \right\} \rightarrow$$

$$\begin{aligned} Y &= Y_1Y_2 + \overline{Y_1}\overline{Y_2} \\ &= (AB + \overline{A}\overline{B})(CD + \overline{C}\overline{D}) + (\overline{A}\overline{B} + \overline{A}B)(\overline{C}\overline{D} + \overline{C}D) \\ &= ABCD + AB\overline{C}\overline{D} + \overline{A}\overline{B}CD + \overline{A}\overline{B}\overline{C}\overline{D} \\ &\quad + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}CD \end{aligned}$$

列出相应真值表,见表 3-3。

表 3-3 例 3-2 的真值表

输 入				输 出	输 入				输 出
A	B	C	D	Y	A	B	C	D	Y
0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	0	1	0	1
0	0	1	1	1	1	0	1	1	0
0	1	0	0	0	1	1	0	0	1
0	1	0	1	1	1	1	0	1	0
0	1	1	0	1	1	1	1	0	0
0	1	1	1	0	1	1	1	1	1

仔细观察上表,分析可发现,当输入变量中有奇数个“1”时,电路的输出为“0”;反之,为全“0”和有偶数个“1”时,电路的输出为“1”,所以该电路是一个奇偶校验器。

以上两个例子的逻辑电路都只有一个输出变量,所以为单输出组合逻辑电路,反之,若组合逻辑电路有多个输出量,则称为多输出组合逻辑电路。下面给出一个多输出组合逻辑电路的例子。

【例 3-3】 试分析下面两输出电路(图 3-8)的逻辑功能。

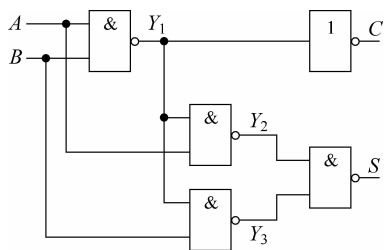


图 3-8 例 3-3 的逻辑电路图

解: 由逻辑电路图分别写出两个输出的逻辑表达式,并进行化简:

$$C = \overline{Y_1} = \overline{AB} = \overline{A}\overline{B}$$

$$\begin{aligned} S &= \overline{Y_2 \cdot Y_3} = \overline{Y_1A \cdot Y_1B} = \overline{AB \cdot A \cdot AB \cdot B} \\ &= \overline{AB \cdot \overline{A}\overline{B}} = \overline{AB} + \overline{A}\overline{B} = A \oplus B \end{aligned}$$

列出真值表,见表 3-4。

表 3-4 例 3-3 的真值表

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

对以上真值表进行分析可以看出,若 A, B 为两个二进制的加数,则 S 为这两个数的和, C 为两数相加向高位的进位,所以该电路可作为运算器中的基本单元电路——半加器。后面章节中将对加法器进行详细介绍。

3.3 组合逻辑电路的设计

3.3.1 组合逻辑电路的设计步骤

组合逻辑电路的设计过程与分析过程互逆,是根据所给逻辑问题,设计出相关的逻辑电路以满足逻辑功能要求的过程。一般来说,组合逻辑电路的设计按照以下步骤进行。

- (1) 分析逻辑问题,抽象出逻辑输入变量和输出变量。
- (2) 依据逻辑要求列出真值表。
- (3) 根据真值表列出逻辑函数表达式。
- (4) 对表达式进行化简或变形,使之与给定器件相匹配。
- (5) 根据逻辑表达式做出逻辑电路图。

3.3.2 组合逻辑电路的设计实例

下面通过两个实例分别给出单输出组合逻辑电路和多输出组合逻辑电路的设计过程。

【例 3-4】 试设计一个裁判表决器。假设在某舞蹈考试的考场有三位考官,其中有一位主考官,两位副考官。当考生完成表演后,由考官按下自己面前的按钮来决定该考生是否通过考试。若有两位或两位以上考官通过(其中一位必须是主考官),则表明考生通过考试。现试用与非门实现电路完成该表决器。

解: (1) 抽象输入变量与输出变量。

根据逻辑问题,有三位考官进行评分,所以设定三个输入变量 A, B, C 分别代表这三位考官,其中 A 表示主考官, B 和 C 表示副考官,取值为“1”时表示考官认为合格,取值为“0”时表示考官认为不合格;每位考生的考试结果只有两种结果,所以用 1 个输出变量 Y 表示,取值为“1”时表示考试通过,取值为“0”时表示考试不通过。

(2) 由给出的条件列出真值表,见表 3-5。

表 3-5 例 3-4 的真值表

输 入			输出	输 入			输出
A	B	C	Y	A	B	C	Y
0	0	0	0	1	0	0	0
0	0	1	0	1	0	1	1
0	1	0	0	1	1	0	1
0	1	1	0	1	1	1	1

(3) 由以上真值表写出逻辑函数表达式,并化为最简与非表达式:

$$\begin{aligned}
 Y &= \overline{A}BC + A\overline{B}C + ABC = (\overline{A}BC + ABC) + (A\overline{B}C + ABC) \\
 &= AC + AB = \overline{\overline{AC} + \overline{AB}} = \overline{\overline{AC} \cdot \overline{AB}}
 \end{aligned}$$

(4) 画出逻辑电路图,如图 3-9 所示。

【例 3-5】 现有两个水泵 L 和 S 从某个深井抽水,其中水泵 L 的功率大于水泵 S 的功率,示意图如图 3-10 所示。当水位低于 A 点时,需要两个水泵一起工作进行抽水;当水位位于 A 点和 B 点之间时,只要水泵 L 工作即可;当水位位于 B 点和 C 点之间时,只需要水泵 S 工作;若水位位于 C 点或 C 以上时,水泵不需要工作。试用与非门设计一个控制电路来控制这两个水泵的工作(输入仅提供原变量)。

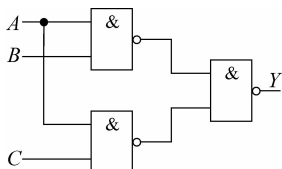


图 3-9 例 3-4 的逻辑电路图

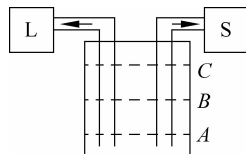


图 3-10 例 3-5 的示意图

解: 设输入逻辑变量为 A, B, C , 输出变量为 L, S , 当水高于某一水位时, 取值为“1”, 否则为“0”; 水泵工作时, 取值为“1”, 否则为“0”。两个水泵工作时的逻辑取值如下:

当水位低于 A 点时, 有 $A=0, B=0, C=0$ 时, $L=1, S=1$;

当水位位于 A 点和 B 点之间时, 有 $A=1, B=0, C=0$ 时, $L=1, S=0$;

当水位位于 B 点和 C 点之间时, 有 $A=1, B=1, C=0$ 时, $L=0, S=1$;

若水位位于 C 点或 C 以上时, 有 $A=1, B=1, C=1$ 时, $L=0, S=0$ 。

除了以上确定情况外, 其他情况均不会出现, 可视为无关项。列出真值表, 见表 3-6。

表 3-6 例 3-5 的真值表

输入			输出		输入			输出	
A	B	C	L	S	A	B	C	L	S
0	0	0	1	1	1	0	0	1	0
0	0	1	×	×	1	0	1	×	×
0	1	0	×	×	1	1	0	0	1
0	1	1	×	×	1	1	1	0	0

利用卡诺图(图 3-11)进行化简, 可得:

$$\begin{cases} L = \bar{B} \\ S = \bar{A} + B\bar{C} \end{cases}$$

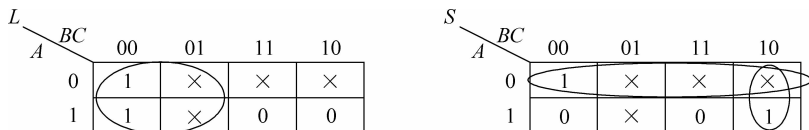


图 3-11 例 3-5 的卡诺图

画出逻辑电路图,如图 3-12 所示。

若要求用与非门实现, 则逻辑表达式变为:

$$\begin{cases} L = \bar{B} = \overline{B \cdot 1} \\ S = \bar{A} + B\bar{C} = \overline{\overline{\bar{A} + B\bar{C}}} = \overline{\overline{\bar{A}} \cdot \overline{B\bar{C}}} = \overline{\overline{\bar{A}} \cdot \overline{B \cdot \bar{C}}} = \overline{\overline{\bar{A}} \cdot \overline{B \cdot \bar{C}} \cdot 1} \end{cases}$$

根据逻辑函数表达式, 画出相应逻辑电路图, 如图 3-13 所示。

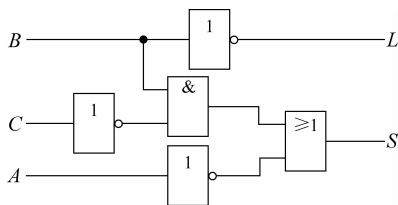


图 3-12 例 3-5 的逻辑电路图

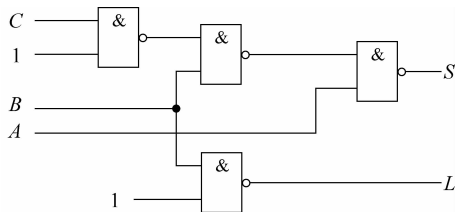


图 3-13 例 3-5 用与非门构造的逻辑电路图

3.4 中规模通用集成电路的逻辑设计

为了使用方便,可以将常用组合逻辑电路的设计标准化,制造成各类中、小规模集成电路芯片,具有通用性强、扩展性好、兼容性好、功耗小、可靠性强、输入负载小等优点。采用中小规模集成电路构造数字系统,可以使数字系统的装配密度增大、结构简化、体积缩小、重量减轻、功耗降低、可靠性提高、设计实现和维护较容易,而且使用方便。本节将介绍几种常用的中规模通用集成电路。

3.4.1 加法器

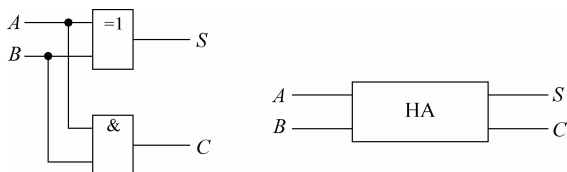
所谓加法器,是指能够实现加法运算的电路。在计算机及其他数字系统中,采用的是二进制的表示方法,不管是加、减运算还是乘、除运算,均要转换为二进制的加法进行运算,所以加法器是算术逻辑运算单元的基本逻辑电路。

1. 半加器

两个 1 位的二进制数相加时,只考虑本位的相加,而不考虑低位来的进位,这种相加被称为半加,能够实现半加功能的逻辑电路被称为半加器。第 3.2 节中的例 3-3 所给的是一个实现半加器的逻辑电路图,对图 3-8 进行改进,可得到用异或门和与门构造的半加器,逻辑函数表达式如下:

$$\begin{cases} C = AB \\ S = A \oplus B \end{cases}$$

图 3-14(a)、图 3-14(b)分别给出了改进后半加器的逻辑电路图和逻辑符号。



(a) 逻辑电路图

(b) 逻辑符号

图 3-14 半加器的逻辑电路图与逻辑符号

2. 全加器

两个 1 位的二进制数相加时,除了考虑本位的相加外,还要考虑低位来的进位,这种相加被称为全加,能够实现全加功能的逻辑电路被称为全加器。全加器的逻辑符号如图 3-15 所示,真值表见表 3-7,其中, C_{n-1} 代表低位来的进

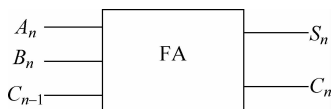


图 3-15 全加器的逻辑符号

位, A_n 和 B_n 代表本位的两个加数, S_n 代表本位和, C_n 代表向高位的进位。

表 3-7 全加器真值表

输 入			输 出		输 入			输 出	
A_n	B_n	C_{n-1}	C_n	S_n	A_n	B_n	C_{n-1}	C_n	S_n
0	0	0	0	0	1	0	0	0	1
0	0	1	0	1	1	0	1	1	0
0	1	0	0	1	1	1	0	1	0
0	1	1	1	0	1	1	1	1	1

由真值表写出全加器本位和与进位信号的表达式, 并进行变形可得:

$$\begin{aligned}
 C_n &= \overline{A_n}B_nC_{n-1} + A_n\overline{B_n}C_{n-1} + A_nB_n\overline{C_{n-1}} + A_nB_nC_{n-1} \\
 &= (\overline{A_n}B_nC_{n-1} + A_n\overline{B_n}C_{n-1}) + (A_nB_n\overline{C_{n-1}} + A_nB_nC_{n-1}) \\
 &= (\overline{A_n}B_n + A_n\overline{B_n})C_{n-1} + A_nB_n(\overline{C_{n-1}} + C_{n-1}) \\
 &= (A_n \oplus B_n)C_{n-1} + A_nB_n \\
 S_n &= \overline{A_n}\overline{B_n}C_{n-1} + \overline{A_n}B_n\overline{C_{n-1}} + A_n\overline{B_n}\overline{C_{n-1}} + A_nB_nC_{n-1} \\
 &= (\overline{A_n}\overline{B_n}C_{n-1} + A_n\overline{B_n}\overline{C_{n-1}}) + (\overline{A_n}B_n\overline{C_{n-1}} + A_nB_nC_{n-1}) \\
 &= (\overline{A_n}\overline{B_n} + A_n\overline{B_n})C_{n-1} + (\overline{A_n}B_n + A_nB_n)\overline{C_{n-1}} \\
 &= \overline{(A_n \oplus B_n)}C_{n-1} + (A_n \oplus B_n)\overline{C_{n-1}} \\
 &= A_n \oplus B_n \oplus C_{n-1}
 \end{aligned}$$

根据逻辑表达式画出逻辑电路图, 如图 3-16 所示。

仔细观察图 3-16 可发现, 全加器可由两个半加器和一个或门构成, 如图 3-17 所示。

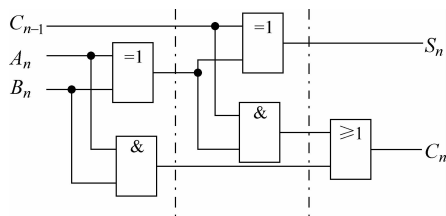


图 3-16 全加器逻辑电路图

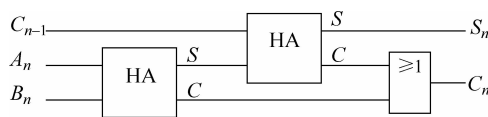


图 3-17 半加器构成全加器

3. n 位加法器

若要实现 n 位二进制的加法运算, 可以采用串行结构或是并行结构。下面以两个 4 位二进制数 $A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$ 相加为例, 给出这两种不同的结构。

1) 串行进位加法器

原理图如图 3-18 所示, 由 4 个全加器级联构成, 低位全加器的进位输出与相邻的高位全加器的进位输入相连, 各全加器的进位按照由低位向高位逐级串行传递, 并形成一条进位链。

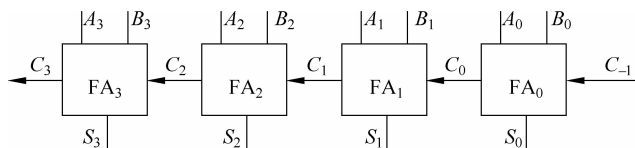


图 3-18 4 位串行进位加法器

串行进位加法器具有电路简单的特点。又由于每一位相加的和都与本位进位输入有关,最高位只有在其他各低位全部相加并产生进位信号之后才能产生最后的运算结果,所以运算速度较慢,而且位数越多,运算速度越低。

2) 超前进位加法器

超前进位加法器不必逐级传递进位信号,解决了串行进位加法器速度慢的问题。由于超前进位加法器可以根据输入信号同时形成各位向高位的进位,所以又被称为先行进位加法器、并行进位加法器。

4 位二进制数 $A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$ 相加,令 $P_i = A_i \oplus B_i$, $G_i = A_i B_i$, 则第 i 位的进位信号为 $C_i = (A_i \oplus B_i)C_{i-1} + A_i B_i = P_i C_{i-1} + G_i$, 各位相加产生的进位表达式如下:

$$C_0 = P_0 C_{-1} + G_0$$

$$C_1 = P_1 C_0 + G_1 = P_1 (P_0 C_{-1} + G_0) + G_1 = P_1 P_0 C_{-1} + P_1 G_0 + G_1$$

$$C_2 = P_2 C_1 + G_2 = P_2 (P_1 P_0 C_{-1} + P_1 G_0 + G_1) + G_2 \\ = P_2 P_1 P_0 C_{-1} + P_2 P_1 G_0 + P_2 G_1 + G_2$$

$$C_3 = P_3 C_2 + G_3 = P_3 (P_2 P_1 P_0 C_{-1} + P_2 P_1 G_0 + P_2 G_1 + G_2) + G_3 \\ = P_3 P_2 P_1 P_0 C_{-1} + P_3 P_2 P_1 G_0 + P_3 P_2 G_1 + P_3 G_2 + G_3$$

式中 P_i 被称为进位传递函数, G_i 为进位产生函数。由以上式子可以看出,各全加器的进位信号只与最低位的进位信号有关,所以在输入两个加数及 C_{-1} 之后,可同时并行产生 $C_0 \sim C_3$, 而不必像串行进位加法器需逐级传递进位信号。由基本门电路构成的逻辑电路图如图 3-19 所示。

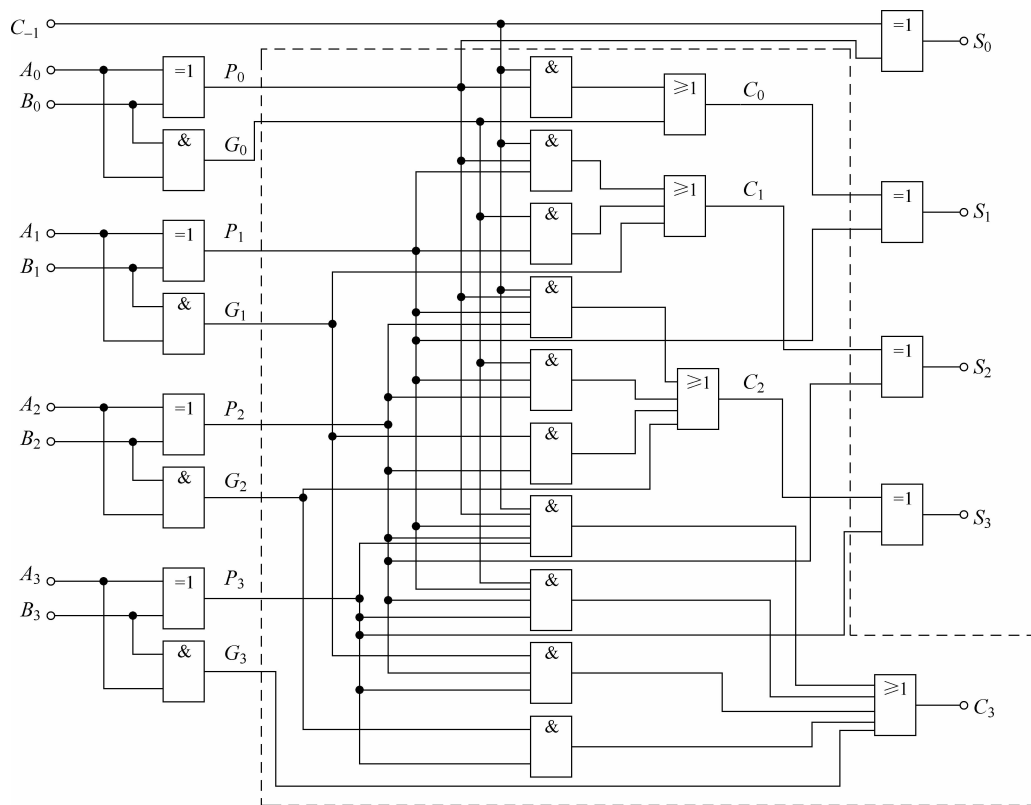


图 3-19 基本门构造超前进位加法器

中规模集成电路 74LS283 是 4 位超前进位全加器,其芯片引脚图如图 3-20 所示,图中 V_{CC} 接电源, GND 接地, $A_3 \sim A_0$ 与 $B_3 \sim B_0$ 分别输入参加运算的 4 位二进制数, C_3 与 $S_3 \sim S_0$ 输出最后运算结果。若要完成更多位数的运算,可以将芯片级联扩展。

【例 3-6】 试用 74LS283 芯片实现两个 8 位数的相加运算。

解: 由于每片 74LS283 可实现 4 位二进制的运算, 8 位数的运算可用两片 74LS283 实现。将两个 8 位数的低 4 位送入低位芯片相加, 高 4 位送入高位芯片相加, 并将低位芯片的 C_{-1} 接地, 进位信号 C_3 接入高位芯片的 C_{-1} 。进行运算后, 由高位芯片的 C_3 和两片芯片的 S 端输出结果。构造的电路图如图 3-21 所示。

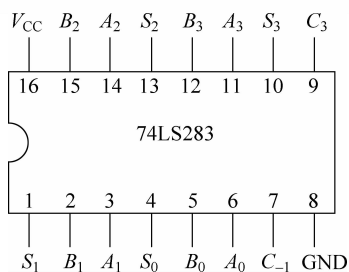


图 3-20 74LS283 芯片引脚图

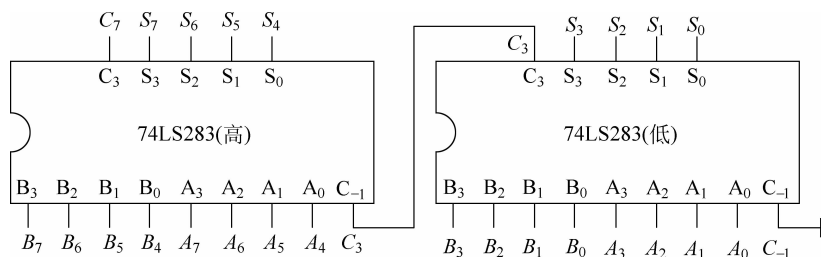


图 3-21 例 3-6 的电路图

3.4.2 数值比较器

所谓数值比较器是指用来比较两个二进制数大小的逻辑电路,其输入变量是两个 n 位二进制数 $A_{n-1}A_{n-2} \cdots A_1A_0$ 和 $B_{n-1}B_{n-2} \cdots B_1B_0$, 输出变量有 $Y_{(A>B)}$ 、 $Y_{(A=B)}$ 、 $Y_{(A<B)}$ 三个,任何时刻有且仅有一个输出变量有效。

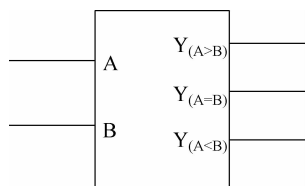


图 3-22 1 位数值比较器逻辑符号

1. 1 位数值比较器

1 位数值比较器的逻辑符号如图 3-22 所示。设 $A > B$ 时, 有 $Y_{(A>B)} = 1$; $A = B$ 时, 有 $Y_{(A=B)} = 1$; $A < B$ 时, 有 $Y_{(A<B)} = 1$, 列出真值表见表 3-8。

表 3-8 1 位数值比较器真值表

输 入		输 出			输 入		输 出		
A	B	$Y_{(A<B)}$	$Y_{(A=B)}$	$Y_{(A>B)}$	A	B	$Y_{(A<B)}$	$Y_{(A=B)}$	$Y_{(A>B)}$
0	0	0	1	0	1	0	0	1	
0	1	1	0	0	1	1	0	1	0

由真值表可写出逻辑函数表达式:

$$\begin{cases} Y_{(A<B)} = \bar{A}B \\ Y_{(A=B)} = \bar{A}\bar{B} + AB = \overline{\bar{A}B + A\bar{B}} \\ Y_{(A>B)} = A\bar{B} \end{cases}$$