

# 第5章 时序逻辑电路设计

## 【内容概要】

时序逻辑电路的设计,就是针对给定的时序逻辑命题,设计出能实现要求的电路。时序逻辑设计也称为时序逻辑综合,是时序逻辑分析的逆过程。

一般情况下,时序逻辑命题只给出要实现的功能及要达到的技术指标。设计者应根据实际条件,决定采用什么样的工作方式、电路结构及元器件。目标是在达到设计要求的前提下,确保稳定性和可靠性,尽可能使电路简化。

本章首先结合调宽码译码器设计的具体实例,介绍同步时序电路设计的基本方法和步骤;其次,结合典型实例对各关键步骤依次阐述其实现原则和基本原理;最后,以较复杂时序逻辑问题序列检测器设计和任意进制计数器设计为例,进一步讨论针对特定应用的时序逻辑电路设计的方法和技巧。

## 【学习目标】

通过学习本章内容,学生能利用同步时序电路设计的基本方法和步骤,针对给定的同步时序逻辑命题,设计出可实现要求的电路,从而提高解决复杂工程问题的能力,培养创新意识。

## 【思政元素】

时序逻辑电路的设计关键在于根据设计需求,确定电路的所有状态及状态之间的转换关系。通过学习时序逻辑电路设计中确定电路状态的方法和技巧,引导学生在面对纷繁复杂的事物或看似无从入手的问题时应沉着冷静、多观察、多思考,培养自身敏锐的洞察力和丰富的联想力,以发现事物的本质特征,提高获取有效信息、解决问题的能力。

## 5.1 同步时序逻辑电路设计的基本方法

在进行同步时序逻辑电路设计时,首先应根据文字描述的功能要求,建立时序逻辑电路的原始状态图和状态表,然后对原始的状态表加以化简,最后选择合适的集成电路器件或给定的集成电路器件来实现状态表,从而达到电路设计目的。需要强调指出的是,同步时序逻辑电路设计中,所有的触发器的时钟输入均由一个公共的时钟脉冲(即同步时序)所驱动。

时序电路的设计是一个比较复杂的问题。虽然同步时序电路的设计在许多方面已有较为完善的方法可以遵循,但在某些方面(如状态化简、状态分配等)还没有完全成熟的方法,需要靠设计者的经验,或从大量方法中进行比较选择。本章尽可能系统地介绍同步时序逻



视频讲解

辑电路设计的主要步骤和方法,并通过一些例子做进一步说明。

目前,中、大规模集成电路种类很多,通过对文字描述的逻辑功能要求做一定的分析后,不必按照本章介绍的步骤逐一套用,只要灵活地应用其中的一些思想及方法,就可以设计出简单、实用、可靠性高的电路。

从设计系统化的角度出发,同步时序电路的设计包括建立原始状态表、状态化简、状态分配(或称状态编码)、选择触发器类型、确定激励函数和输出函数、画出逻辑图、检查逻辑电路的功能等步骤。

本节以一个实例,引出同步时序逻辑设计的基本步骤及方法。

**【例 5-1】** 设计一个调宽码译码器。

**解:** 调宽码是一种串行码。因抗噪声干扰能力较强,常用于无线或红外数据通信中。例如,很多家用电器的红外遥控数据就是调宽码。图 5.1 所示为调宽码的编码格式,其中同步时钟的周期为  $T$ ,用于对调宽码进行定位。图 5.1 中假定待传送的原始数据为 10010100。由图 5.1 可看出,调宽码用不同的宽度和占空比代表原始数据的 1 和 0,调宽码的特征见表 5.1。

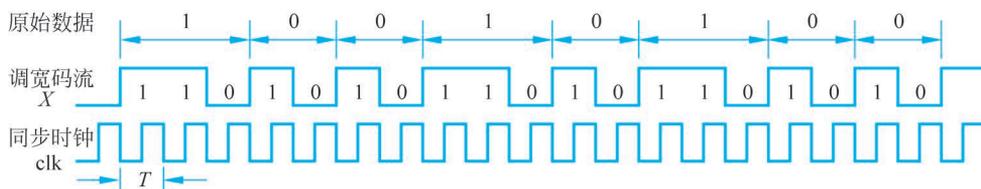


图 5.1 调宽码的编码格式

表 5.1 调宽码的特征

原始数据位	调宽码		
	宽度	占空比	说明
1	$3T$	$1/3$	前两个 $T$ 内为高电平,后一个 $T$ 内为低电平
0	$2T$	$1/2$	前一个 $T$ 内为高电平,后一个 $T$ 内为低电平

第 1 步,分析命题,规划电路框架。

记调宽码流为  $X$ ,译出的数据为  $Z$ ,同步时钟为  $clk$ 。要将  $X$  译为  $Z$ ,可在  $clk$  脉冲的上升沿对  $X$  取样。如果连续取样得到的序列为 110,则  $Z=1$ ; 如果为 10,则  $Z=0$ ; 否则就是误码。显然,待设计的逻辑电路应该记忆  $X$  中的 0 以前的取样,才能决定当前的  $Z$  是 1 还是 0。因此,待设计的电路是时序逻辑电路。

电路需要一个调宽码流输入端  $X$ ,一个公共时钟输入端  $clk$ ,电路中的所有触发器都要使用  $clk$ 。需要两个输出端  $Z$  及  $E$ , $Z$  用于输出译码值, $E$  用于指示当前的  $Z$  是否有效,约定  $E=1$  时表示  $Z$  值有效。其电路框架见图 5.2。

第 2 步,根据设计功能要求,建立原始状态图及状态表。

由图 5.1 看出,一个原始数据位对应的码流以  $X=0$  为结束标志,且由  $E$  在此时的值指明是否被译出。因此, $E$  的表达式中含有变量  $X$ ,即采用 Mealy 型电路有利于产生输出量  $E$ 。

下面逐步分析需要建立哪些状态,这些状态各代表什么含义。先给出原始状态图(见图 5.3),并说明各状态及相互关系。

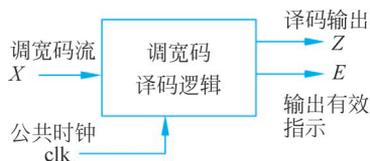


图 5.2 调宽码译码逻辑电路框架图

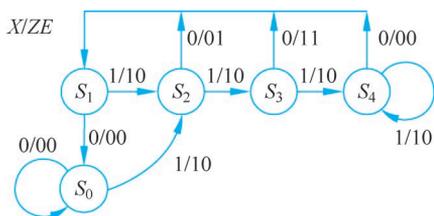


图 5.3 例 5-1 的原始状态图

状态  $S_0$ ：连续 0 误码状态。本状态表示已连续收到  $X=0$  的次数大于 1。在此状态下，若再收到  $X=0$ ，则应维持本状态；若再收到  $X=1$ ，则说明误码结束，可能下一个原始数据位正在到来，应转入状态  $S_2$  处理。

状态  $S_1$ ：等待状态。此时已译出一位原始数据，等待下一位原始数据的到来。在此状态下，若收到  $X=0$ ，则是误码，应转入  $S_0$  状态；若收到  $X=1$ ，则说明下一个原始数据位正在到来，应转入下个状态  $S_2$ ，进一步判断正在到来的原始数据位是 0 还是 1，或是误码。

状态  $S_2$ ：已收到待译出数据的第一次  $X=1$  的取样。在此状态下，若再收到  $X=0$ ，则译出 1 个为 0 的原始数据位，应转入状态  $S_1$ ，同时输出  $ZE=01$ ；若再收到  $X=1$ ，则说明正在到来的原始数据位可能为 1 或误码，需转入状态  $S_3$  进一步判断。

状态  $S_3$ ：已收到待译出数据到的第二次  $X=1$  的取样。在此状态下，若再收到  $X=0$ ，则译出一个为 1 的原始数据位，应转入状态  $S_1$ ，同时输出  $ZE=11$ ；若再收到  $X=1$ ，则说明是误码，需转入状态  $S_4$  处理。

状态  $S_4$ ：连续 1 误码状态。此时已连续收到  $X=1$  的次数大于 2。在此状态下，若再收到  $X=1$ ，则应维持本状态，等待误码结束；若再收到  $X=0$ ，则说明误码结束，转入  $S_1$ 。

除上面指定的  $ZE$  输出值外，其余情况下必须令  $E=0$ ，表示  $Z$  无效。现将  $Z$  无效时的值记为无关项  $\Phi$ ，以简化  $Z$  的生成逻辑。原始状态图如图 5.3 所示。由原始状态图可作出原始状态表，如表 5.2 所示。

表 5.2 例 5-1 的原始状态表

现态	次态/输出 ( $S_i/ZE$ )	
	$X=0$	$X=1$
$S_0$	$S_0/00$	$S_2/10$
$S_1$	$S_0/00$	$S_2/10$
$S_2$	$S_1/01$	$S_3/10$
$S_3$	$S_1/11$	$S_4/10$
$S_4$	$S_1/00$	$S_4/10$

第 3 步，原始状态化简。

在第 2 步中，按事件的自然发展规律指定了 5 个状态。状态越多，记忆状态所需的触发器也就越多。状态化简就是把某些多余的或重复的状态加以合并，使状态的数目减为最少。观察表 5.2 中的两个现态  $S_0$  及  $S_1$ ，不难发现：

- (1) 当输入为  $X=0$  时，它们都转到次态  $S_0$ ，且都输出  $ZE=00$ ；
- (2) 当输入为  $X=1$  时，它们都转到次态  $S_2$ ，且都输出  $ZE=10$ 。

这说明  $S_0$  和  $S_1$  可以合并为一个状态，记为  $S_1$ ，于是得到化简后的状态图，如图 5.4 所示。对应的状态表如表 5.3 所示。通过化简使原来的 5 个状态减少了一个。

必须指出，仅凭一般观察很难全面、合理地完成状态化简，尤其是对于复杂的时序逻辑设计。这一问题将在 5.3 节中详细讨论。

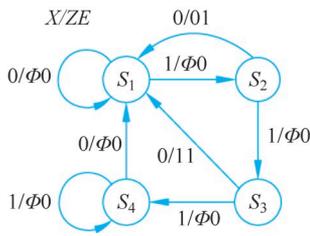


图 5.4 例 5-1 化简后的状态图

表 5.3 例 5-1 化简后的状态表

现态	次态/输出( $S_i/Z/E$ )	
	$X=0$	$X=1$
$S_1$	$S_1/\Phi 0$	$S_2/\Phi 0$
$S_2$	$S_1/01$	$S_3/\Phi 0$
$S_3$	$S_1/11$	$S_4/\Phi 0$
$S_4$	$S_1/\Phi 0$	$S_4/\Phi 0$

第 4 步,状态编码。

以上对各状态用符号进行了命名。状态是用触发器记忆的,因此可用二进制代码表示各状态,以便能在触发器中存储。已知  $n$  个触发器能表示的状态数为  $2^n$  个,用 2 个触发器恰好能表示 4 个状态。如果不化简,则存储原来的 5 个状态至少需要 3 个触发器。

2 个触发器能存储 4 种代码: 00,01,10,11。究竟哪个代码分配给哪个状态? 这就是编码。编码方案不同,所设计电路的复杂程度也不同。常见编码方案有自然二进制码、格雷码和独热码 3 种,本例采用如下自然二进制码方案:

$$S_1: 00, \quad S_2: 01, \quad S_3: 10, \quad S_4: 11$$

用编码代替状态名,得到编码后的状态表,如表 5.4 所示。

第 5 步,确定激励函数及输出方程。

要使状态按照既定的目标转移,必须为各触发器的激励端配置合适的激励逻辑电路。首先,要选择触发器的类型。原则上,选用任何类型的触发器都可达到设计目的,但触发器类型不同,激励函数的复杂程度也不同。究竟选用什么类型的触发器为好? 目前尚无行之有效的理论方法。实际操作时一般基于经验进行。这里选用 JK 触发器,时钟控制端为上升沿触发。

将状态代码记为  $y_2y_1$ ,两个触发器的激励记为  $J_2、K_2$  和  $J_1、K_1$ 。激励函数要根据当前的输入  $X$  及现态  $y_2y_1$ ,驱动触发器转移到指定的次态。所以, $J_2、K_2$  及  $J_1、K_1$  均为  $X、y_2、y_1$  的函数。当然, $X、Z$  也是  $X、y_2、y_1$  的函数。这是 1 个多输入、多输出组合逻辑电路的设计问题,其逻辑结构如图 5.5 所示。为了设计此组合逻辑,需列出其真值表,如表 5.5 所示。

表 5.4 编码后的状态表

现态 $y_2y_1$	次态/输出 $y_2^{n+1}y_1^{n+1}/ZE$	
	$X=0$	$X=1$
0 0	00/ $\Phi 0$	01/ $\Phi 0$
0 1	00/01	10/ $\Phi 0$
1 0	00/11	11/ $\Phi 0$
1 1	00/ $\Phi 0$	11/ $\Phi 0$

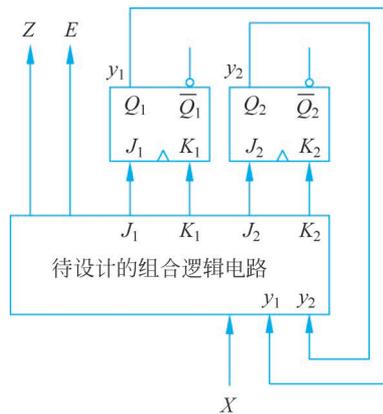


图 5.5 例 5-1 的逻辑结构

表 5.5 待设计的组合逻辑真值表

输入			$y_2^{n+1}$ $y_1^{n+1}$		输出					
$X$	$y_2$	$y_1$			$J_2$	$K_2$	$J_1$	$K_1$	$Z$	$E$
0	0	0	0	0	0	$\Phi$	0	$\Phi$	$\Phi$	0
0	0	1	0	0	0	$\Phi$	$\Phi$	1	0	1
0	1	0	0	0	$\Phi$	1	0	$\Phi$	1	1
0	1	1	0	0	$\Phi$	1	$\Phi$	1	$\Phi$	0
1	0	0	0	1	0	$\Phi$	1	$\Phi$	$\Phi$	0
1	0	1	1	0	1	$\Phi$	$\Phi$	1	$\Phi$	0
1	1	0	1	1	$\Phi$	0	1	$\Phi$	$\Phi$	0
1	1	1	1	1	$\Phi$	0	$\Phi$	0	$\Phi$	0

下面讨论表 5.5 中的数据是如何推导出来的。对于图 5.5 中的组合逻辑电路而言,  $X$ 、 $y_2$ 、 $y_1$  是输入量,  $J_2$ 、 $K_2$ 、 $J_1$ 、 $K_1$  及  $Z$ 、 $E$  是输出量。我们的目的是对  $X$ 、 $y_2$ 、 $y_1$  的各种组合值, 按状态图的要求确定  $J_2$ 、 $K_2$ 、 $J_1$ 、 $K_1$  的值。要想确定  $J_2$ 、 $K_2$ 、 $J_1$ 、 $K_1$  的值, 又须知道次态是什么, 故在表 5.5 中列出了次态  $y_2^{n+1}y_1^{n+1}$  栏。此栏完全是为推导方便而列出的, 在后面作卡诺图时不会涉及此栏的值。下面以第 1 行为例, 说明推导过程。为推导时方便起见, 将 JK 触发器的激励表列于表 5.6 中(表 5.6 中  $\Phi$  为任意项)。

表 5.6 JK 触发器激励表

$Q \rightarrow Q^{n+1}$		$J$	$K$
0	0	1	$\Phi$
0	1	1	$\Phi$
1	0	$\Phi$	1
1	1	$\Phi$	0

(1) 已知  $X=0, y_2 y_1=00$ , 查状态表 5.4 知:  $y_2^{n+1}y_1^{n+1}=00, ZE=\Phi 0$ 。将查表结果填入表 5.5 中(单波浪下画线部分)。

(2) 对于触发器 2, 已知现态  $y_2=0$ , 次态  $y_2^{n+1}=0$ , 查 JK 触发器的激励表知, 实现  $y_2 \rightarrow y_2^{n+1}$  转移所需的  $J_2 K_2=0\Phi$ , 将其填入表中(双波浪下画线部分)。

(3) 对于触发器 1, 已知现态  $y_1=0$ , 次态  $y_1^{n+1}=0$ , 查 JK 触发器的激励表可知, 实现  $y_1 \rightarrow y_1^{n+1}$  转移所需的  $J_1 K_1=0\Phi$ , 将其填入表 5.5 中(下画虚线)。

其余各行照此处理。由表 5.5, 分别作出各输出量的卡诺图, 如图 5.6 所示。

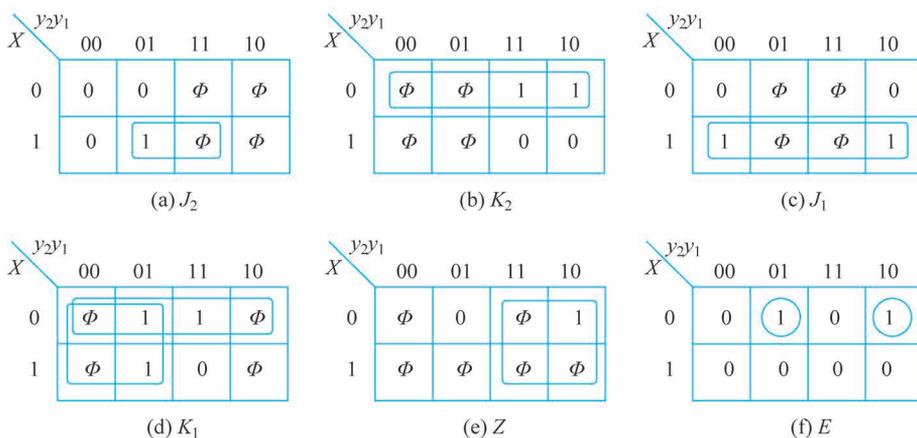


图 5.6 例 5-1 的卡诺图

由卡诺图化简,得到激励函数和输出函数如下:

$$J_2 = Xy_1 \quad K_2 = \bar{X}$$

$$J_1 = X \quad K_1 = \bar{y}_2 + \bar{X} = \overline{y_2 X}$$

$$Z = y_2 \quad E = \bar{X}\bar{y}_2y_1 + \bar{X}y_2\bar{y}_1 = \bar{X}(y_2 \oplus y_1)$$

由此可画出调宽码译码器的逻辑电路,如图 5.7 所示。

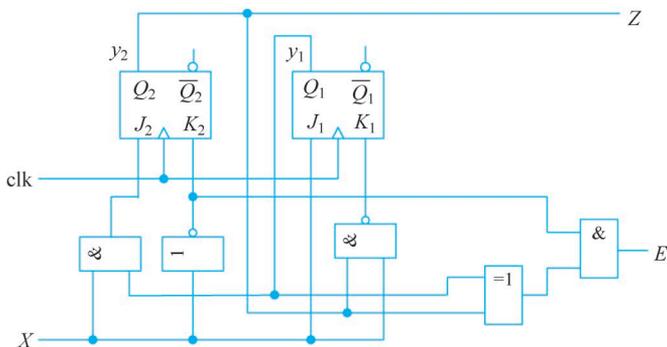


图 5.7 例 5-1 的逻辑电路图

本节用一个简单的例子介绍了时序逻辑电路设计的基本步骤和方法,同时也引出了许多需要进一步讨论的问题。如:采用 Moore 型还是 Mealy 型电路?如何化简状态?怎样合理设计分配状态编码?选择什么类型的触发器为好?由此看出,时序逻辑电路设计具有极大的灵活性。在 5.2 节~5.5 节中,将围绕实例,就这些问题展开专门讨论。



视频讲解

## 5.2 建立原始状态

原始状态图是根据问题的文字描述作出的状态图,是对设计要求的第一次抽象,是后续设计的重要依据。因此,把解决问题的整体部署和具体细节,无遗漏地反映在原始状态图中,是成功地实现设计的关键一步。

在建立原始状态图时主要考虑以下 4 点。

(1) 确定采用 Moore 型还是 Mealy 型电路。Moore 型电路的输出由状态量决定,记忆历史输入需要状态量,产生输出也需要状态量参与,故一般情况下,Moore 型电路需要的状态数比 Mealy 型的多。但 Moore 型电路的设计较简单,如果输出可由状态编码完全确定中,或状态编码本身就是希望的输出(如计数器),则可采用 Moore 型电路;如果求输出量时,有输入量参与运算较简便,则可采用 Mealy 型电路。必须指出,很多时序逻辑设计既可采用 Moore 型又可采用 Mealy 型电路,但设计的复杂程度不同。

(2) 找准第一个状态(即初始状态)。时序逻辑电路在输入信号作用前的状态称为初始状态。描述某个电路的状态图或状态表中,用不同状态作为初始状态时,对相同输入序列所产生的状态响应序列和输出序列是不相同的。因此,在建立原始状态图时,应首先找到初始状态,然后从初始状态出发考虑在各种输入作用下的状态转移和输出响应。

(3) 根据需要记忆的信息,从初始状态开始,逐步扩充新状态。同步时序逻辑电路中状态数目的多少取决于需要记忆和区分的信息量。一般而言,当某个状态下输入信号作用的

结果能用已有的某个状态表示时,应转向相应的已有状态。仅当某个状态下输入信号作用的结果不能用已有的某个状态表示时,才令其转向新的状态。注意,所扩充的状态是现有状态不能表示的状态。如果一时不能确定,宁愿扩充也不要造成遗漏。如果该状态是重复的,在状态化简时会得以合并。

(4) 边扩充新状态边确定状态的转移及输入、输出变量。时序电路的功能是通过输出对输入的响应来体现的。在建立原始状态图时,必须确定各时刻的输出。在 Moore 型电路中,应指明每种状态下对应的输出;在 Mealy 型电路中则应指明从每个状态出发,在不同输入作用下对应的输出值。注意,如果输入有  $n$  个,则从任一状态出发、指向其他状态的转移线一般有  $2^n$  条。当然,可能存在几个不同的输入组合值共用一条转移线到达另一状态的情况,此时必须明确标明,否则原始状态表中将会缺少数据项。

**【例 5-2】** 设计一个 4 位串行二进制码奇偶检测器电路。

**解:** 数据的传输方式有并行传输与串行传输。并行传输即通过一组导线同时传输数据的各个二进制位,速度快,但线路成本高,适用于近距离传输。串行传输则是逐位传输数据,速度较慢,但因传输导线少而成本低,适用于远距离传输。

本例传输的数据为 4 位二进制码,即每 4 位二进制位为一组,以串行方式输入检测电路。图 5.8 所示为输入数据的格式,同步时钟  $\text{clk}$  的上升沿与每位数据  $x$  的中点对齐,检测电路在  $\text{clk}$  的上升沿采集  $x$ 。每当一组数据的最后一位到达后,即判断该组中含有多少个 1。含有偶数个 1 时电路的输出  $z=1$ ,否则  $z=0$ 。图 5.9 为奇偶检测电路的框架。

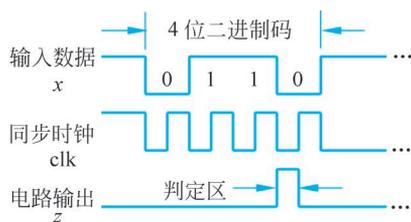


图 5.8 串行数据格式



图 5.9 奇偶检测电路的框架

依据上述关于电路的分析,本例采用 Mealy 型电路。

假设以等待一组数据的首位到来为第一个状态  $A$ 。当收到首位数据为 0 时,转到状态  $B$  并输出 0;若首位数据为 1,则转到状态  $C$  并输出 0,于是扩充了两个新状态  $B$  和  $C$ 。在  $B$  和  $C$  状态下,输入数据均可能是 0 或 1。故由  $B$  状态可扩充  $D$  和  $E$  状态,由  $C$  状态可扩充  $F$  和  $G$  状态。因此,共得到 15 个状态。其中  $H \sim O$  状态在收到第 4 位数据输入后,就能判断应输出  $z=0$  还是  $z=1$ ,并且都转到  $A$  状态,等待检测下一组数据。由此得到原始状态图如图 5.10 所示。

如果采用 Moore 型电路,则原始状态图将如图 5.11 所示。因为输出仅由状态量决定,图 5.11 中缺少输出  $z=1$  的状态,故需要增加一个状态  $P$ ,用于输出  $z=1$ 。由此可见,本例若采用 Moore 型电路,则需要的状态数比 Mealy 型的多。

由 Mealy 型原始状态图作出原始状态表,如表 5.7 所示。

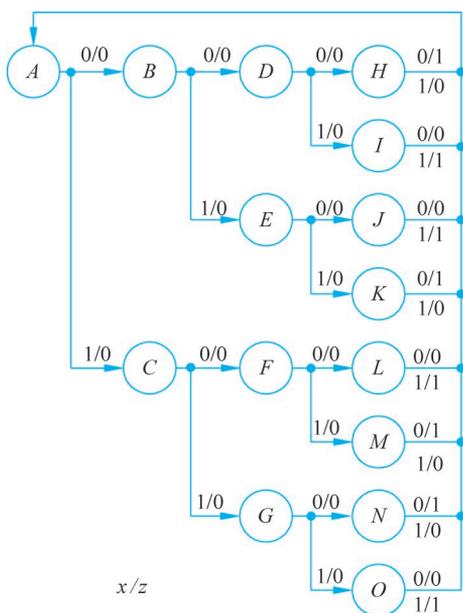


图 5.10 例 5.2 的 Mealy 型原始状态图

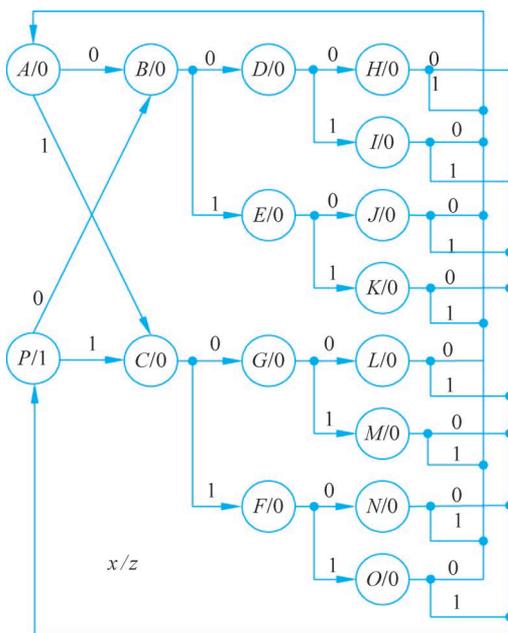


图 5.11 例 5.2 的 Moore 型原始状态图

表 5.7 例 5-2 的原始状态表

现 态	次态/输出		现 态	次态/输出	
	$x=0$	$x=1$		$x=0$	$x=1$
A	B/0	C/0	I	A/0	A/1
B	D/0	E/0	J	A/0	A/1
C	F/0	G/0	K	A/1	A/0
D	H/0	I/0	L	A/0	A/1
E	J/0	K/0	M	A/1	A/0
F	L/0	M/0	N	A/1	A/0
G	N/0	O/0	O	A/0	A/1
H	A/1	A/0			

## 5.3 状态化简

原始状态图往往带有主观性,与设计者的经验有很大的关系。状态化简是设计的重要步骤。现有多种化简方法,在介绍之前先讨论化简的基本原理。

### 5.3.1 状态化简的基本原理

#### 情形 1 次态相同

若有状态  $S_i$  和  $S_j$ , 在相同的输入下,都转到同一个次态,并且产生相同的输出,则  $S_i$  和  $S_j$  可以合并。称  $S_i$  和  $S_j$  为等效对,记为  $(S_i, S_j)$ 。这一情形如图 5.12 所示,将合并后的状态记为  $S_i$ 。图 5.12(a)中  $S_m, S_n$  为次态;图 5.12(b)中  $S_n$  为一个次态,另一个次态

即  $S_i$ 。表 5.7 中存在着大量的等效对,例如  $(H, K)$ 、 $(I, J)$ 、 $(J, L)$ 、 $(K, M)$  等,在 5.3.2 节中再详细讨论对它们的合并。

这里必须说明两点:次态可以是与现态相同的状态;欲考察两个状态是否为等效对,必须考察每个状态下的所有输入值,如果电路的输入量有  $n$  个,则输入值有  $2^n$  种,并且要考察各状态在相同的输入值下,是否都产生相同的输出并且都转到同一个次态。下面的讨论中,除特别说明外均默认如此。

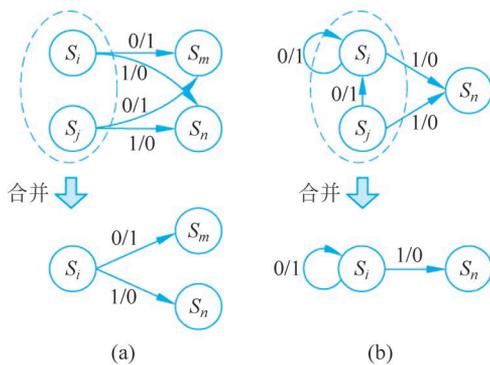


图 5.12 情形 1 次态相同

情形 2 次态交错

若  $S_i$  和  $S_j$  在某些输入值下互为次态且输出相同,但在其他输入值下满足情形 1,则  $S_i$  和  $S_j$  为等效对,可以合并。

如图 5.13 所示,当输入为 0 时,  $S_i$  和  $S_j$  互为次态且输出均为 1; 当输入为 1 时,  $S_i$  和  $S_j$  都转到  $S_m$  态且输出都为 0。因此,  $S_i$  和  $S_j$  为等效对,可将其合并为一个状态,记为  $S_i$ 。

情形 3 状态对封闭链

若有几对状态,对于每一对状态而言,在相同的输入下,能产生相同的输出但到达的次态不同。若它们构成所谓“状态对封闭链”,则这些状态对均为等效对。

如图 5.14 所示上部原始状态图,  $S_i$  和  $S_j$ 、 $S_q$  和  $S_p$ 、 $S_m$  和  $S_n$  均为等效对。

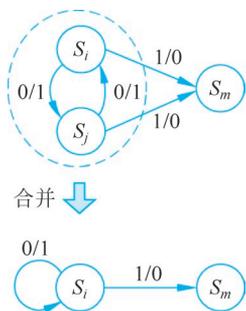


图 5.13 情形 2 次态交错

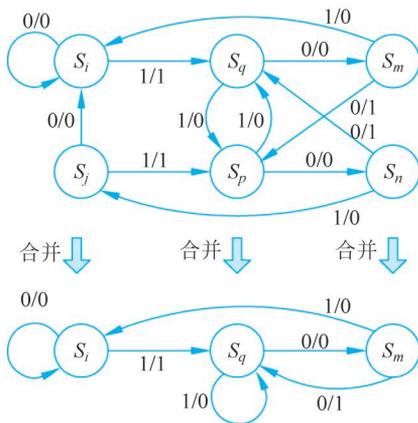


图 5.14 情形 3 状态对封闭链

对于  $S_i$  和  $S_j$ ,在输入为 0 时都转到  $S_i$  状态且都输出 0; 在输入为 1 时,尽管输出相同(都为 1),但分别转到不同的状态  $S_q$  和  $S_p$ 。如果  $S_q$  和  $S_p$  为可以合并,则  $S_i$  和  $S_j$  就为等效对。为此,需考察  $S_q$  和  $S_p$ 。

对于  $S_q$  和  $S_p$ ,在输入为 1 时次态交错且都输出 0; 在输入为 0 时,尽管输出相同(均为 0),但分别转到不同的状态  $S_m$  和  $S_n$ 。如果  $S_m$  和  $S_n$  可以合并,则  $S_q$  和  $S_p$  就为等效对。为此,又需考察  $S_m$  和  $S_n$ 。

对于  $S_m$  和  $S_n$ ,在输入为 0 时次态分别为上述需要考察的  $S_q$  和  $S_p$ ,且都输出 1; 在输入为 1 时,次态也分别为上述需要考察的  $S_i$  和  $S_j$ 。

综上所述可以看出,如果  $S_i$  和  $S_j$  为可以合并,则  $S_m$  和  $S_n$  就能合并;如果  $S_m$  和  $S_n$  可以合并,则  $S_q$  和  $S_p$  就能合并;如果  $S_q$  和  $S_p$  可以合并,则  $S_i$  和  $S_j$  就能合并。这种相互依从的关系称为状态对封闭链。

状态对封闭链中的每对状态都是可以合并的。这是因为:对于链中的任一对状态,例如  $S_q$  和  $S_p$ ,任给一个输入序列,分别从  $S_q$  和  $S_p$  出发,产生的输出序列必然相同。例如,给定输入序列 01101001,从  $S_q$  出发产生的输出序列为 00100001;从  $S_p$  出发产生的输出序列也为 00100001。图 5.14 所示下部的状态图即为化简结果。

### 5.3.2 完全定义状态化简方法

所谓完全定义状态图(或状态表),是指对于所涉及的任一状态,都明确定义了对全部输入值的具体次态响应。以上的讨论仅限于完全定义的状态图(或状态表)。

实际中,存在不完全定义状态图(或状态表)。所谓不完全定义,是指所涉及的状态中,有些状态对于某些输入值的次态响应不需要给出定义或为任意,即次态为任意态  $\Phi$ 。

完全定义和不完全定义状态图(或状态表)的化简方法不同。本节仅讨论完全定义的状态图(或状态表)的化简。

#### 1. 有关定义

首先,给出几个涉及化简的定义。

**等效对:**若两个状态  $S_i$  和  $S_j$ ,对于所有可能的输入序列,分别从  $S_i$  和  $S_j$  出发,产生的输出序列相同,则称  $S_i$  与  $S_j$  等效;称  $S_i$  和  $S_j$  为等效状态对,简称等效对,记为  $(S_i, S_j)$ 。所谓“所有可能的输入序列”是指序列的长度及值的组合任意。这样的序列有无穷多个,要想以此检测等效对是不现实的,而应采用 5.3.1 节所述的方法。

**等效对的传递性:**若状态  $S_i$  和  $S_j$  为等效对,状态  $S_j$  和  $S_m$  为等效对,则  $S_i$  和  $S_m$  也为等效对,称为等效对的传递性,记为  $(S_i, S_j), (S_j, S_m) \rightarrow (S_i, S_m)$ 。

**等效类:**若一个状态集合中的任何两个状态都互为等效,则称该状态集合为等效类。例如,若有  $(S_i, S_j), (S_j, S_m) \rightarrow (S_i, S_m)$ ,则  $\{S_i, S_j, S_m\}$  为等效类。

**注意:**只含有一个状态的集合也是等效类。

**最大等效类:**将状态图(或状态表)中的全部状态划分为若干等效类,若某个等效类中的状态不能与其他等效类中的状态构成等效对,则这个等效类称为最大等效类。最大等效类与其他等效类的交集为空集合。

由以上定义可知,最大等效类中的状态可以合并为 1 个状态。因此,状态化简的过程就是将全部状态划分为若干最大等效类的过程。

#### 2. 隐含表化简方法

从原始状态表或状态图上直接寻找状态对,对于 5.3.1 节中的情形 1 较容易,但对于情形 2 和情形 3 则较困难。用隐含表化简法则能全面找出状态对,进而确定全部最大等效类。现在继续求解例 5-2,完成状态化简。以表 5.7 为例,介绍隐含表化简方法。

隐含表即图 5.15 所示的三角形框架表格。在各行的左侧依从上到下次序标出状态名称,从第二个状态开始直到最后一个状态;各列的下边依从左到右次序标出状态名称,从第一个状态开始直到倒数第二个状态。每格代表其所在行、列的状态。这种表格能使所有状态彼此配对而又不重复,因而不会遗漏可能存在的等效对。

B	BD CE																
C	BF CG	DF EG															
D	BH CI	DH EI	FH GI														
E	BJ CK	DJ EK	FJ GK	HJ IK													
F	BL CM	DL EM	FL GM	HL IM	JL KM												
G	BN CO	DN EO	FN GO	HN IO	JN KO	LN MO											
H	×	×	×	×	×	×	×										
I	×	×	×	×	×	×	×	×									
J	×	×	×	×	×	×	×	×	√								
K	×	×	×	×	×	×	×	√	×	×							
L	×	×	×	×	×	×	×	×	√	√	×						
M	×	×	×	×	×	×	×	√	×	×	√	×					
N	×	×	×	×	×	×	×	√	×	×	√	×	√				
O	×	×	×	×	×	×	×	×	√	√	×	√	×	×	×		
	A	B	C	D	E	F	G	H	I	J	K	L	M	N			

图 5.15 例 5-2 的隐含表

化简分以下 3 步进行。

(1) 判断各方格代表的状态是否为等效对。例如：

① 状态 A 与 O, 当  $x=0$  时都输出 0, 但当  $x=1$  时输出不同, 因此断定 A 与 O 不是等效对, 在对应格中填入“×”。

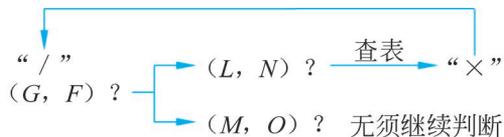
② 状态 I 与 O, 当  $x=0$  时都输出 0 且都转到 A 状态, 当  $x=1$  时都输出 1 且都转到 A 状态, 因此断定 I 与 O 是等效对, 在对应格中填入“√”。

③ 状态 A 与 B, 当  $x=0$  都输出 0, 但分别转到 B、D, 当  $x=1$  时都输出 0, 但分别转到 C、E, 由此可见, A 与 B 能否等效, 要看 B 与 D 能否等效且 C 与 E 能否等效, 因此在代表 BA 的格中同时填入“BD”和“CE”, 表示暂时未决。

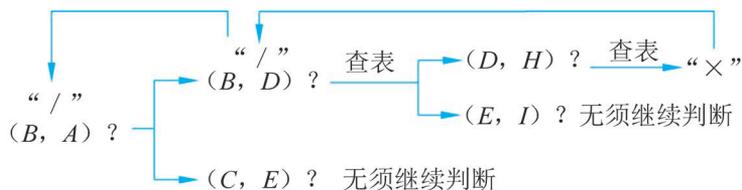
以此类推, 完成全部表格的填写。

(2) 审查未决格中记录的状态对, 只要有一对可断定为非等效对, 则该格代表的状态就不是等效对, 将其画上“/”线以示否定, 如：

① 代表 G 与 F 的格。该格中记录了 L 与 N、M 与 O 两个状态对。先看 L 与 N, 代表 L 与 N 的格中为“×”, 因此立即判定 G 与 F 不是等效对, 将对应格画上“/”线。判断流程示意如下：



② 代表 B 与 A 的格。该格中记录了 B 与 D、C 与 E 两个状态对。先看 B 与 D, 代表 B 与 D 的格中又记录了 D 与 H、E 与 I 两个状态对。再看 D 与 H, 代表 D 与 H 的格中为“×”。因此判定 B 与 D 不是等效对, 由此又判定 B 与 A 不是等效对。判断流程示意如下：



以此类推,完成全部未决格的判定。如果最后还有形成封闭链的未决格,则这些格代表的状态都是等效对,在这些格中标上“√”。

(3) 求最大等效类。用等效对的传递性容易证明:在同一列上的“√”格及这些“√”格所在的行上的“√”格涉及的状态构成一个最大等效类。观察图 5.15,可以看出:

- ①  $H$  列、 $M$  行、 $N$  行及  $K$  行上的所有“√”格涉及的状态构成最大等效类。

$$Q_1 = \{H, M, N, K\}$$

- ②  $I$  列、 $O$  行、 $L$  行及  $J$  行上的“√”格涉及的所有状态构成最大等效类。

$$Q_2 = \{I, O, L, J\}$$

- ③  $(D, G)$ 、 $(E, F)$  分别构成最大等效类。

$$Q_3 = \{D, G\}, \quad Q_4 = \{E, F\}$$

$Q_1$ 、 $Q_2$  中未涉及的状态与其他状态不能构成等效对,它们各自构成最大等效类。

将  $Q_1$  中的状态合并为 1 个状态,记为  $H$ ; 将  $Q_2$  中的状态合并为 1 个状态,记为  $I$ 。 $Q_3$  合并记为  $D$ ,  $Q_4$  合并记为  $E$ , 于是得到最小化状态表,如表 5.8 所示。所需状态数由原来的 15 个减少到 7 个。对应的最小化状态图如图 5.16 所示。

表 5.8 例 5-2 的最小化状态表

现态	次态/输出	
	$x=0$	$x=1$
A	B/0	C/0
B	D/0	E/0
C	E/0	D/0
D	H/0	I/0
E	I/0	H/0
H	A/1	A/0
I	A/0	A/1

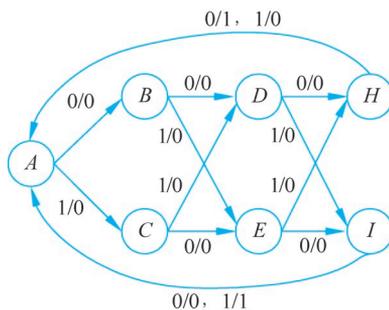


图 5.16 例 5-2 的最小化状态图

## 5.4 状态编码

状态编码就是用二进制代码表示各状态,从而实现用一组触发器来存储状态。为此需要研究:编码的长度(即二进制代码的位数)取多少合适,以确定要使用多少个触发器;把哪个代码指派给哪个状态,能使输出函数最简单。后者是个十分复杂的问题,尤其是当状态数目较多时,可用的分配方案数量极大。企图将每种方案一一实现,再从中选择最佳结果,是不现实的。而且,分配方案的好坏还与触发器类型的选择相关。在理论上,状态分配问题至今尚未很好解决,下面将介绍一种基于经验的状态分配方法——相邻编码法。

### 5.4.1 确定存储状态所需的触发器个数

设简化后的状态表共含有  $n$  个状态,希望能用尽可能短的编码长度代表这些状态,使所需触发器的个数  $m$  最少。则  $n$  与  $m$  应满足:

$$2^m \geq n \geq 2^{m-1}, \quad \text{或 } m = \lceil \log_2 n \rceil \quad (5-1)$$

式(5-1)中,  $\lceil \log_2 n \rceil$  表示求不小于  $\log_2 n$  的最小整数。

继续求解例 5-2。表 5.8 中共有 7 个状态,由式(5-1)求得所需触发器的个数  $m = 3$ 。故状态代码为 3 位,将其记为  $y_3 y_2 y_1$ 。

### 5.4.2 用相邻编码法实现状态编码

时序电路的输出是输入变量和状态变量的函数,对于例 5-2 有  $z = f(x, y_3, y_2, y_1)$ 。这说明,选择合适的状态分配方案是简化输出函数的一条途径。

对于一个状态,给予的代码不同,在卡诺图中对应的位置也不同。如图 5.17 所示,如果将代码 001 指派给状态 A,则在卡诺图中对应的位置为  $A'$ ; 如果将代码 101 指派给状态 A,则在卡诺图中对应的位置为  $A''$ 。这说明,通过指派代码使得各状态在卡诺图中形成合理的分布,获得尽可能大的卡诺圈,就能达到简化输出函数的目的。这就是相邻编码法的依据。

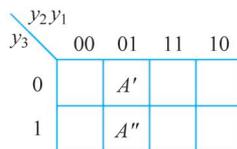


图 5.17 代码指派与位置示意

所谓相邻编码,就是为 2 个状态指派的二进制代码只有一位不同。在卡诺图上表现为 2 个状态左右或上下相邻。例如,上述代码 001 与 101 仅最高位不同,在图 5.17 中的位置  $A'$  与  $A''$  即为上下相邻。相邻编码法的规则如下。

**规则 1** 输入相同且次态相同的现态应为相邻编码。

观察表 5.8,  $H$  和  $I$  符合规则 1, 应取相邻编码。

**规则 2** 同一现态的次态应为相邻编码。

观察表 5.8, 状态  $B$  和  $C, D$  和  $E, H$  和  $I$ , 应分别取相邻编码。

**规则 3** 输入不同但输出相同的现态应为相邻编码。

观察表 5.8, 状态  $A \sim E$  应分别取相邻编码。

由此可以发现,有些状态同时满足几条规则。此时应按“规则 1→规则 2→规则 3”的优先顺序处理。对表 5.8 按上述规则进行操作的结果如图 5.18 所示。其中状态  $A$  作为初始态,配以代码 000。在系统上电时(即开始接通电源的时刻)一般都把所有的触发器清 0,故系统上电后电路即处于  $A$  态,等待输入。编码后的状态表如表 5.9 所示。

表 5.9 表 5.8 编码后的状态表

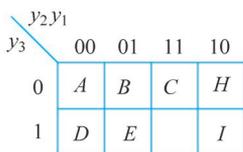


图 5.18 表 5.9 的状态分配

现 态 $y_3 y_2 y_1$	次态/输出	
	$x = 0$	$x = 1$
000	001/0	011/0
001	100/0	101/0
011	101/0	100/0
100	010/0	110/0
101	110/0	010/0
010	000/1	000/0
110	000/0	000/1

## 5.5 确定激励函数及输出方程

### 5.5.1 选定触发器类型

激励函数的任务是根据电路的现态和当前输入,驱动触发器转移到指定的次态。触发器类型不同,所需激励函数也不同。因此,在确定激励函数之前,应选定触发器的类型。

在触发器的选型上,常依赖经验判断。对于数据锁存、移位类时序逻辑电路,宜选用 D 触发器;对于计数类时序逻辑电路,宜选用 T 触发器;当无明显规律时,则可选用 JK 触发器,也可以同时选用几种类型的触发器。但情况并非总是如此,设计者应以激励函数最简单为目标。D 触发器只有一个激励端,这意味着只需要一个激励函数便可驱动翻转,可作为优选对象。JK 触发器虽有两个激励端,但其功能丰富,不失为一种可取的选择。

现在为例 5-2 选择触发器。电路需要存储 4 位串行数据,最终作出判断,在性质上属数据锁存、移位类逻辑,不妨选用 D 触发器。对应于  $y_3y_2y_1$ ,各触发器的激励端记为  $D_3D_2D_1$ 。

### 5.5.2 求激励函数及输出函数

各驱动方程激励函数和输出函数都是组合逻辑问题,其输入为串行数据  $x$  和电路的现态  $y_3y_2y_1$ ,输出为各激励量  $D_3D_2D_1$  和判断结果  $z$ 。因需要根据现态和次态来确定激励量,故在列真值表时将次态作为索引的中间量列入表中,如表 5.10 所示。由 D 触发器的激励方程  $D=Q^{n+1}$  可知, $D_3D_2D_1=y_3^{n+1}y_2^{n+1}y_1^{n+1}$ ,因此很容易列出各输出量。

表 5.10 例 5-2 的真值表

输 入				中 间 量			输 出			
$x$	$y_3$	$y_2$	$y_1$	$y_3^{n+1}$	$y_2^{n+1}$	$y_1^{n+1}$	$z$	$D_3$	$D_2$	$D_1$
0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0	0
0	0	1	1	1	0	1	0	1	0	1
0	1	0	0	0	1	0	0	0	1	0
0	1	0	1	1	1	0	0	1	1	0
0	0	1	0	0	0	0	1	0	0	0
0	1	1	0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	0	0	1	1
1	0	0	1	1	0	1	0	1	0	1
1	0	1	1	1	0	0	0	1	0	0
1	1	0	0	1	1	0	0	1	1	0
1	1	0	1	0	1	0	0	0	1	0
1	0	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	1	0	0	0
0	1	1	1	—	—	—	$\Phi$	$\Phi$	$\Phi$	$\Phi$
1	1	1	1	—	—	—	$\Phi$	$\Phi$	$\Phi$	$\Phi$

在编码时,多余的代码“111”未指派具体状态。如果电路能按照状态转换图工作,就不会进入这种状态。即“111”代表的状态为无关状态(任意项),故表 5.10 中将其对应的输出以任意项  $\Phi$  列出,以期达到简化激励函数的目的。

由表 5.10 可以看出,输出量  $z$  的最小项很少,故直接求出其函数较简单

$$z = \bar{x}\bar{y}_3y_2\bar{y}_1 + xy_3y_2\bar{y}_1 = \overline{\bar{x}\bar{y}_3y_2\bar{y}_1xy_3y_2\bar{y}_1} \quad (5-2)$$

画出各激励量的卡诺图并化简,如图 5.19 所示。显然,图 5.19 中不存在相切的卡诺圈,即无产生险象的因素。

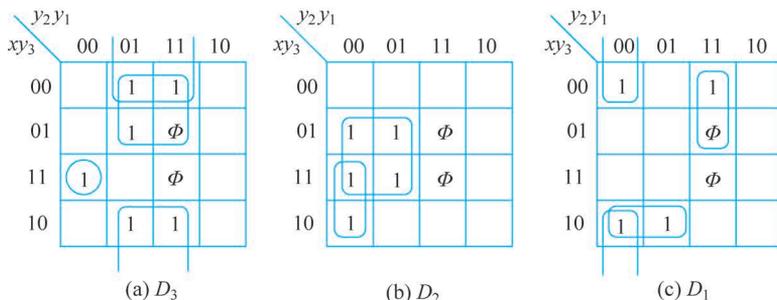


图 5.19 例 5-2 的卡诺图

由图 5.19 可得

$$\left. \begin{aligned} D_3 &= xy_3\bar{y}_2\bar{y}_1 + \bar{x}y_1 + \bar{y}_3y_1 = \overline{\bar{x}y_3\bar{y}_2\bar{y}_1 \bar{x}y_1 \bar{y}_3y_1} \\ D_2 &= x\bar{y}_2\bar{y}_1 + y_3\bar{y}_2 = \overline{x\bar{y}_2\bar{y}_1 y_3\bar{y}_2} \\ D_1 &= \bar{y}_3\bar{y}_2\bar{y}_1 + x\bar{y}_3\bar{y}_2 + \bar{x}y_2y_1 = \overline{y_3\bar{y}_2\bar{y}_1 x\bar{y}_3\bar{y}_2 \bar{x}y_2y_1} \end{aligned} \right\} \quad (5-3)$$

由式(5-2)、式(5-3)画出电路图,如图 5.20 所示。

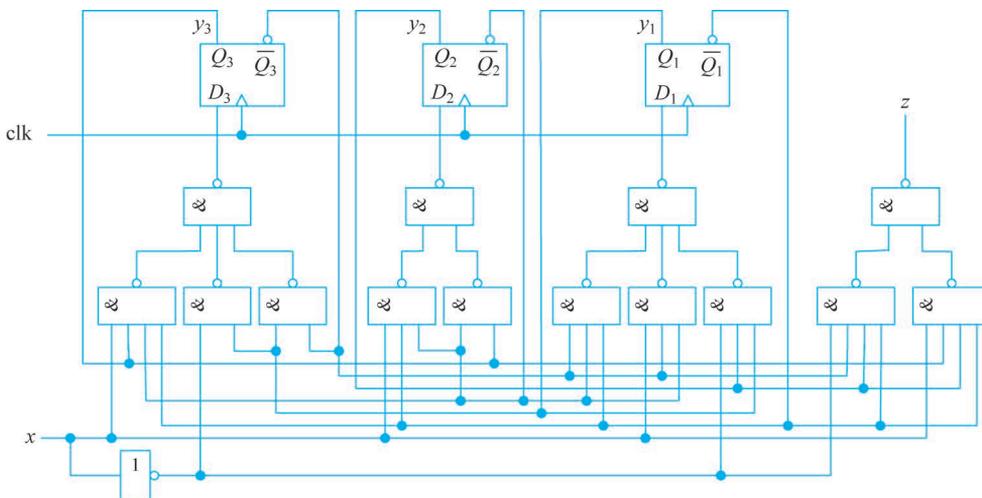


图 5.20 例 5-2 的电路图

### 5.5.3 电路的“挂起”及恢复问题

例 5-2 假设电路能按照状态转换图工作,不会进入无关状态“111”,这只是一种理想的

情况。实际运行中,电路可能受到外界的强电磁干扰,使触发器产生误翻转而进入这种所谓的“无关状态”。未分配的编码越多,进入无关状态的可能性越大。由此产生的后果有以下3种情形。

(1) 进入无关状态后,无论什么输入都不能使电路回到正常状态,即在无效状态间形成无限循环。称这种情况为“挂起”。显然,电路挂起后将丧失全部功能。

(2) 进入无关状态后,再经过若干时钟周期能自动恢复到正常状态,但中途产生错误的输出,进而使后续电路执行错误动作。

(3) 进入无关状态后,再经过若干时钟周期能自动恢复到正常状态,且不产生错误的输出。

上述3种情形中,前两种是不允许发生的。后一种情形虽危害性较小,但希望恢复速度越快越好。

要解决这一问题,首先要判断电路是否存在这一情形。由式(5-2)及式(5-3)可计算出表5.10中各输出量 $\Phi$ 的具体值如下:

(1) 当 $xy_3y_2y_1=0111$ 时,计算得 $zD_3D_2D_1=0101$ 。即一旦进入“无关状态”后,若输入为0,则输出 $z=0$ ,并在下个时钟脉冲的下降沿转到101状态(即正常状态E)。

(2) 当 $xy_3y_2y_1=1111$ 时,计算得 $zD_3D_2D_1=0000$ 。即一旦进入“无关状态”后,若输入为1,则输出 $z=0$ ,并在下个时钟脉冲的下降沿转到000状态(即正常状态A)。

由此,可作出如图5.21所示的完整状态图,图中虚线表示无关状态及其转移线。由图5.21可见,当电路进入无关状态后,无论输入如何,只要下个时钟脉冲到达,就立即转移到正常状态。故本例的电路不会挂起,但正常的工作秩序已被短暂扰乱。

如果电路挂起或产生错误输出,则应在保证设计功能的前提下,修改激励函数或输出函数,强制其进入正常状态或消除错误输出。

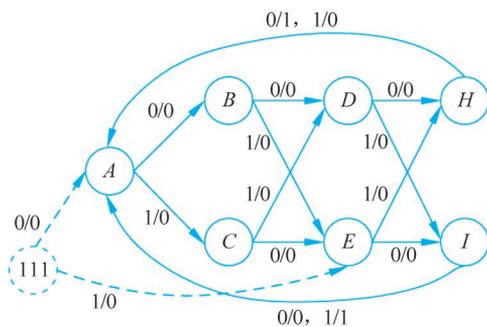


图 5.21 例 5-2 的完整状态图

## 5.6 时序逻辑设计举例

本节以几个常用的逻辑问题为例,进一步讨论时序逻辑电路的设计,并引出设计与实现中的若干具体问题及其处理方法与技巧。

### 5.6.1 序列检测器设计

序列检测器的功能是,在串行传输的数据序列中找到特定的子序列。这一功能通常用于串行数据的定位。例如,在如下数据序列中寻找子序列“110”,如找到,则输出高电平脉冲。

数据序列:  $\cdots 0\underline{11}00010111\underline{110}10010 \cdots$

输出序列:  $\cdots 000100000000100000 \cdots$

数据序列中带下画线的数为“110”序列。



视频讲解

**【例 5-3】** 设计 1 个“111”序列检测器,当连续收到 3 个(或 3 个以上)“1”后,电路输出  $Z=1$ ; 否则,输出  $Z=0$ 。

**解:** 第 1 步,分析题意,构建电路框架。

依题意,序列检测器框图如图 5.22 所示,输入、输出时序图如图 5.23 所示。



图 5.22 序列检测器框图

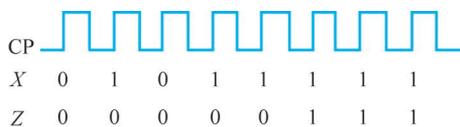


图 5.23 输入、输出时序图

第 2 步,建立原始状态图和状态表。

设初态  $S_0$  收到 1 个“0”,并且用  $S_i (i=1,2,3)$  表示收到第  $i$  个“1”,由此可得到如图 5.24(a)所示 Mealy 型原始状态图,以及原始状态表(见表 5.11),表 5.11 中设电路开始处于初始状态为  $S_0$ 。第一次输入“1”时,由状态  $S_0$  转入状态  $S_1$ ,并输出“0”;若继续输入“1”,由状态  $S_1$  转入状态  $S_2$ ,并输出“0”;如果仍接着输入“1”,由状态  $S_2$  转入状态  $S_3$ ,并输出“1”;此后若继续输入“1”,电路仍停留在状态  $S_3$ ,并输出“1”。电路无论处在什么状态,只要输入“0”,都应回到初始状态  $S_0$ ,并输出“0”,以便重新计数“1”。

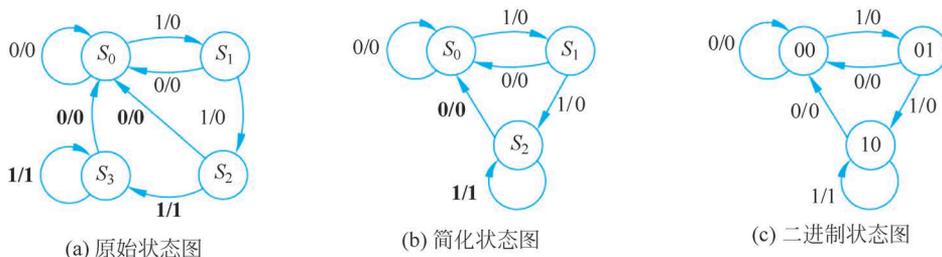


图 5.24 例 5-3 状态图及其化简

第 3 步,状态化简。

原始状态图中,凡是在输入相同时,输出相同、要转换到的次态也相同的状态,称为等价状态。状态化简就是将多个等价状态合并成 1 个状态,把多余状态都去掉,得到最简状态图。图 5.24(a)中,状态  $S_2$  和  $S_3$  等价,因为它们在输入为“1”时输出都为“1”,且都转换到次态  $S_3$ ;在输入为“0”时输出都为“0”,且都转换到次态  $S_0$ 。所以它们可以合并为一个状态,合并后的状态用  $S_2$  表示,得到如图 5.24(b)所示的简化状态图。

第 4 步,状态分配。

根据简化状态图,可得状态编码:  $S_0=00$ ;  $S_1=01$ ;  $S_2=10$ 。最后画出其简化后的二进制状态图如图 5.24(c)所示。由原始状态表 5.11 及简化后的二进制状态图 5.24(c),可得其简化状态转移表(见表 5.12)。表 5.12 中,由于  $S_3$  状态为不确定项,用任意项 d 表示。

表 5.11 例 5-3 的原始状态表

现 态	次态/输出 Z	
	x=0	x=1
$S_0$	$S_0/0$	$S_1/0$
$S_1$	$S_0/0$	$S_2/0$
$S_2$	$S_0/0$	$S_3/0$
$S_3$	$S_0/0$	$S_3/1$

表 5.12 例 5-3 的简化状态转移表

输入 $X$	现 态		次 态		输出 $Z$
	$Q_1$	$Q_0$	$Q_1^{n+1}$	$Q_0^{n+1}$	
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	d	d	d
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	d	d	d

第 5 步,选触发器,求时钟、输出、状态、驱动方程。

采用同步设计方案,选用 2 个 CP 下降沿触发的 JK 触发器,分别用  $JK_0$ 、 $JK_1$  表示。根据表 5.12 所示的简化状态转移表及 JK 触发器激励方程,可得到序列检测器状态方程和输出方程表达式(5-4)和式(5-5)。

次态方程:

$$Q_0^{n+1} = X\bar{Q}_1^n\bar{Q}_0^n; \quad Q_1^{n+1} = XQ_0^n\bar{Q}_1^n + XQ_1^n \quad (5-4)$$

输出方程:

$$Z = XQ_1^n \quad (5-5)$$

用 JK 触发器的特性方程  $Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n$  和式(5-4)、式(5-5)比较,得  $J$ 、 $K$  的激励方程式(5-6):

$$\begin{cases} J_0 = X\bar{Q}_1^n & K_0 = 1 \\ J_1 = XQ_0^n & K_1 = \bar{X} \end{cases} \quad (5-6)$$

第 6 步,检查电路能否自启动。

将无效状态  $Q_1Q_0 = 11$  代入式(5-4)和式(5-5)计算得:  $X = 0$  时,  $Q_1Q_0 = 00$ ,  $Z = 0$ ;  $X = 1$  时,  $Q_1Q_0 = 10$ ,  $Z = 1$ 。电路能够自启动,因此设计符合要求。

第 7 步,画逻辑电路图。

根据式(5-5)和式(5-6)得图 5.25 所示例 5-3 逻辑电路图。

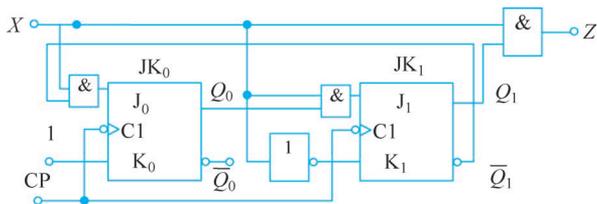


图 5.25 例 5-3 逻辑电路图

## 5.6.2 计数器设计

如 4.5.2 节所述,计数器是一种最常用的时序逻辑部件。例如,计算机中的定时器、地址发生器、节拍发生器等,都要用到计数器。电子钟实际上就是一个六十进制、十二/二十四进制的计数器。计数器的基本功能是记录某事件发生的次数。



计数器的种类很多,通常有不同的分类方法。按其工作方式可分为同步计数器和异步计数器;按其进位制可分为二进制计数器、十进制计数器和任意进制计数器;按其功能又可分为加法计数器、减法计数器和加/减可逆计数器等。

下面以一个模 7 加法计数器设计为例,讨论时序逻辑电路计数器的设计方法。

**【例 5-4】** 用 JK 触发器设计一个按自然态序变化的七进制同步加法计数器,计数规则为逢七进一,产生一个进位输出。

**解:** 第 1 步,分析题意。

根据题目所给的条件,待设计的计数器默认为模 7 计数,且不要求加载初值。故电路只需时钟输入端 clk,clk 作为电路的同步时钟,不必当作输入变量对待;输出一个七进制数要 3 个输出端,记为  $Q_2Q_1Q_0$ 。要有输出进位信号 Y,故共需要 4 个输出端。因输出量  $Q_2Q_1Q_0$  就是计数值,故采用 Moore 型电路较合适。

第 2 步,建立原始状态图。

模 7 计数器要求有 7 个记忆状态,且逢七进一。由此可以作出如图 5.26 所示的原始状态转移图。由于模 7 计数器必须要有 7 个记忆状态,所以不需要再简化。

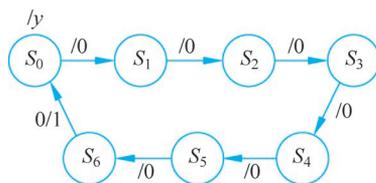


图 5.26 例 5-4 原始状态转移图

第 3 步,状态分配。

由于最大模值为 7,因此必须取编码位数  $n=3$ 。假设令  $S_0=000, S_1=001, S_2=010, S_3=011, S_4=100, S_5=101, S_6=110$ ,则可以作出状态转移表,如表 5.13 所示。由于在状态转移表中 111 状态未出现(偏离状态),此处作任意项  $x$  处理。

表 5.13 例 5-4 状态转移表

输入 CP	现 态			次 态			输出 $Y(t)$
	$Q_2$	$Q_1$	$Q_0$	$Q_2^{n+1}$	$Q_1^{n+1}$	$Q_0^{n+1}$	
0	0	0	0	0	0	1	0
1	0	0	1	0	1	0	0
2	0	1	0	0	1	1	0
3	0	1	1	1	0	0	0
4	1	0	0	1	0	1	0
5	1	0	1	1	1	0	0
6	1	1	0	0	0	0	1
7	1	1	1	$x$	$x$	$x$	$x$

第 4 步,选触发器,求时钟、输出、状态、驱动方程。

因需用 3 位二进制代码,选用 3 个 CP 下降沿触发的 JK 触发器,分别用 FF0、FF1、FF2 表示。由于要求采用同步方案,故时钟方程为

$$CP_0 = CP_1 = CP_2 = CP$$

由表 5.13 可以作出次态卡诺图及输出函数的卡诺图,如图 5.27 所示。

根据卡诺图求出状态方程式(5-7),输出方程式(5-8)。不必化简,以便使之与 JK 触发器的特性方程的形式一致。

$$\begin{aligned}
 Q_0^{n+1} &= \overline{Q_2^n} \overline{Q_0^n} + \overline{Q_1^n} \overline{Q_0^n} \\
 &= \overline{Q_2^n Q_1^n} \overline{Q_0^n} + \overline{1} Q_0^n
 \end{aligned}
 \tag{5-7}$$

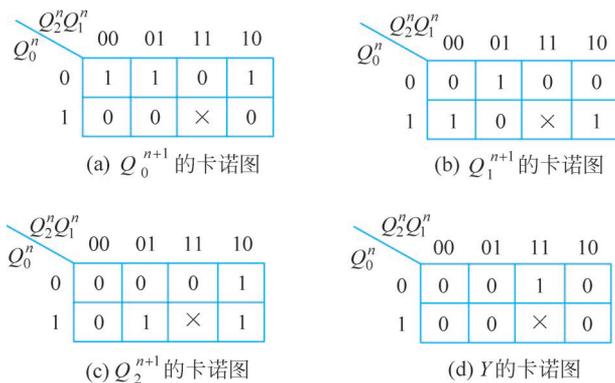


图 5.27 例 5-4 的次态及输出函数卡诺图

$$\begin{aligned}
 Q_1^{n+1} &= Q_0^n \bar{Q}_1^n + \bar{Q}_2^n \bar{Q}_0^n Q_1^n \\
 Q_2^{n+1} &= Q_1^n Q_0^n \bar{Q}_2^n + \bar{Q}_1^n Q_2^n \\
 Y &= Q_1^n Q_2^n
 \end{aligned} \tag{5-8}$$

与 JK 触发器的特性方程  $Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n$  比较,得驱动方程式(5-9)。

$$\begin{aligned}
 J_0 &= \bar{Q}_2^n \bar{Q}_1^n, K_0 = 1 \\
 J_1 &= Q_0^n, K_1 = \bar{Q}_2^n \bar{Q}_0^n \\
 J_2 &= Q_1^n Q_0^n, K_2 = Q_1^n
 \end{aligned} \tag{5-9}$$

第 5 步,检查电路能否自启动。

将无效状态 111 代入状态方程式(5-7),计算得

$$\begin{cases}
 Q_0^{n+1} = \bar{Q}_2^n \bar{Q}_1^n \bar{Q}_0^n + 1Q_0^n = 0 \\
 Q_1^{n+1} = Q_0^n \bar{Q}_1^n + \bar{Q}_2^n \bar{Q}_0^n Q_1^n = 0 \\
 Q_2^{n+1} = Q_1^n Q_0^n \bar{Q}_2^n + \bar{Q}_1^n Q_2^n = 0
 \end{cases} \tag{5-10}$$

分析式(5-10)可见,无效状态 111 的次态可转移到有效状态 000,电路能够自启动。

第 6 步,画逻辑电路图。

根据式(5-8)和式(5-9)得图 5.28 所示例 5-4 逻辑电路图。

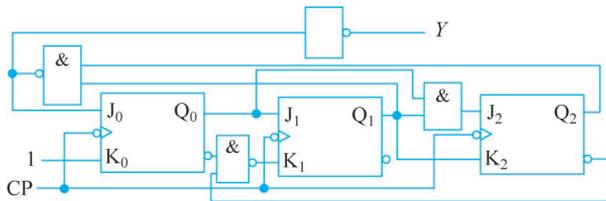


图 5.28 例 5-4 逻辑电路图

### 5.6.3 基于 MSI 器件实现任意模值计数器

中规模集成(medium scale integration, MSI)计数器应用范围很广,从简单的二进制计数器到十进制可逆计数器等,种类较多。表 5.14 给出了常见 MSI 计数器芯片。不过需提

醒读者注意的是：集成十进制同步加法计数器 74160、74162 的引脚排列图、逻辑功能示意图与 74161、74163 相同；74190 引脚排列图和逻辑功能示意图与 74191 相同；74192 引脚排列图和逻辑功能示意图与 74193 相同。

表 5.14 常见 MSI 计数器芯片

CP 脉冲引入方式	型号	计数模式	清零方式	预置数方式
同步	74160	十进制加法计数器	异步(低电平)	同步
同步	74162	十进制加法计数器	同步	同步
同步(单时钟)	74190	十进制可逆计数器	无	异步
同步	74161	4 位二进制加法计数器	异步(低电平)	同步
同步	74163	4 位二进制加法计数器	同步(低电平)	同步
同步(双时钟)	74193	4 位二进制可逆计数器	异步(高电平)	异步
同步(单时钟)	74191	4 位二进制可逆计数器	无	异步
异步(双时钟)	74293	4 位二进制加法计数器	异步	无
异步	74290	二-五-十进制加法	异步	异步

应用  $N$  进制中规模集成计数器实现任意模值  $M(M < N)$  计数器时,主要是从  $N$  进制计数器的状态转移表中跳越  $(N - M)$  个状态,从而得到  $M$  个状态转移,即  $M$  进制计数器。通常通过控制中规模集成计数器的清 0 端(复位法)和置数端(置数法)来实现。

### 1. 复位法

当中规模  $N$  进制计数器从  $S_0$  状态开始计数时,计数脉冲输入  $M$  个脉冲后, $N$  进制计数器处于  $S_M$  状态。如果利用  $S_M$  状态产生一个清零信号,加到计数器的清零端,使计数器返回到  $S_0$  状态,这样就跳越了  $(N - M)$  个状态,从而实现模值为  $M$  的计数器。

**【例 5-5】** 用 MSI 器件 74161 来构成一个十二进制计数器。

**解:** 74161 是 4 位二进制加法计数器,其功能真值表如表 5.15 所示。

表 5.15 74161/74160 功能真值表

输 入									输 出				
CLR <sub>N</sub>	LD <sub>N</sub>	ENT	ENP	CP	D	C	B	A	Q <sub>D</sub>	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>	RCO
0	X	X	X	X	X	X	X	X	0	0	0	0	0
1	0	X	X	↑	D	C	B	A	D	C	B	A	
1	1	1	1	↑	X	X	X	X	步进计数				
1	1	0	X	X	X	X	X	X	保持				
1	1	X	0	X	X	X	X	X	保持				0

十二进制计数器即为模 12 计数器。模 12 计数要求在输入 12 个脉冲后电路返回到 0000,且产生一个输出脉冲。74161 共有 16 个状态。模 12 计数器只需 12 个状态,因此在 74161 基础上,外加判别和清 0 信号产生电路。图 5.29 所示为应用 74161 构成的模 12 计数器电路。

图 5.29 中  $G_1$  门为判别门,当第 12 个计数脉冲上升沿输入后,74161 的状态进入 1100,则门  $G_1$  输出  $X=0$ ,作用于门  $G_2$  和  $G_3$  组成的基本触发器,使  $Q$  端为 0,作用 74161 的 CLR<sub>N</sub> 端,则使 74161 清 0。在计数脉冲 CP 下降沿到达后,又使门  $G_3$  输出  $Q=1, NQ=0$ 。



## 2. 置位法

置位法适用于具有置数控制端的中规模集成器件,通过置入某一固定二进制数值的方法,使  $N$  进制计数跳越  $(N-M)$  个状态,实现规模值为  $M$  的计数分频。在其计数过程中,可将它输出的任何 1 个状态通过译码,产生 1 个预置数控制信号,并反馈至预置数控制端,在下一个 CP 脉冲作用下会把预置数输入端  $DCBA$  的状态置入输出端,预置数控制信号消失后,计数器从被置入的状态开始重新计数。

**【例 5-6】** 用 4 位二进制同步计数器 74163,实现模 12 计数分频器。

**解:** 由 74163 功能表(表 5.15),当  $LDN=0$  时,执行同步置数功能。用 74163 构成十二进制分频器,可把输出  $Q_D Q_C Q_B Q_A = 1011$  状态译码产生的预置数控制信号  $P$ (此时为 0),反馈至  $LDN$  端,在下一个 CP 的上升沿到达时置入 0000 状态,如图 5.31 所示。

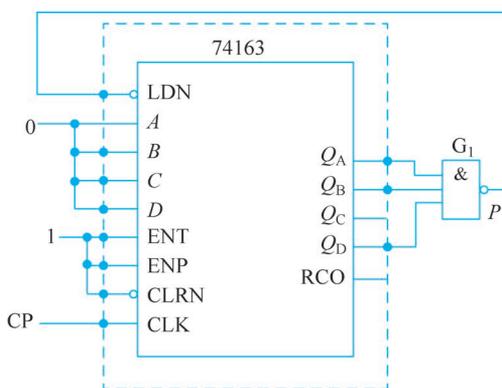


图 5.31 基于置位法的模 12 计数分频器原理图

反馈置数操作可在 74163 计数循环状态(0000~1111)中的任何一个状态下进行,如将  $Q_D Q_C Q_B Q_A = 1111$  状态译码产生的预置数控制信号  $P$  加至  $LDN$  端,这时预置数输入端应为  $1111-1100+1=0100$ 。

## 3. MSI 计数器的级联应用

若一片计数器的计数容量不够用时,可取若干片扩展。同步式连接:以低位片的进位输出作为高位片的工作状态控制信号,各片共用同一时钟。

异步计数器一般没有专门的进位信号输出端,通常可以用本级的高位输出信号驱动下一级计数器计数,即采用串行进位方式来扩展容量。

同步计数器有进位或借位输出端,可以选择合适的进位或借位输出信号来驱动下一级计数器计数。同步计数器级联的方式有两种:一种级间采用串行进位方式,即异步方式,这种方式是将低位计数器的进位输出直接作为高位计数器的时钟脉冲,异步方式的速度较慢;另一种级间采用并行进位方式,即同步方式,这种方式一般是把各计数器的 CP 端连在一起接统一的时钟脉冲,而低位计数器的进位输出送高位计数器的计数控制端。

### 【延伸思考】

根据需求设计相应的时序逻辑电路,是学习本课程所必须掌握的一项基本技能,其关键是确定待设计的时序电路的所有状态以及状态之间的转换关系。在当今信息化的时代,世事纷繁复杂,实际生活中的设计需求也往往较为复杂,因此需要我们具备透过现象看本质的能力,从看似复杂的问题需求中抽丝剥茧,明确问题的根本和关键,从而高效地解决问题、推

进工作。

回望历史,透过现象看本质是中国共产党一贯倡导的科学认识方法,是中国共产党领导革命、建设和改革事业取得伟大成就的重要经验总结。毛主席曾说过:“我们看事情必须要看它的实质,而把它的现象只看作入门的向导,一进了门就要抓住它的实质,这才是可靠的科学的分析方法。”土地革命时期,面对黑云压城、白色恐怖、革命低潮,中国共产党透过极为险恶和残酷的环境,看到了“星星之火,可以燎原”的革命必胜前景,为中国革命指明了正确方向;党的十一届三中全会后,中国共产党总结正反两方面经验教训,抓住了社会主义的本质,为经济发展和国家富强开辟了道路。当前,面对世界百年未有之大变局,以习近平同志为核心的党中央,洞察我国发展环境面临的深刻复杂变化,指出我国发展仍然处于重要战略机遇期,我们正处于大有可为的新时代的重要论断,为广大党员矢志奋斗鼓足了劲、加满了油。实践表明,只有善于透过现象看本质,才能科学认识事物的客观规律,准确把握时代的发展大势,从而正确地指导实践。

作为新时代的爱国青年,我们只有不断提升运用马克思主义辩证法和方法论的能力,练就透过现象看本质的“火眼金睛”,做到眼睛亮、见事早、行动快,才能在学习、工作中把握关键、剑指要害,更好地解决实际问题,推动社会主义建设的持续高速发展。