

第6章 异步时序逻辑电路

时序逻辑电路按照工作方式的不同,可以分为同步时序逻辑电路和异步时序逻辑电路两种类型,异步时序逻辑电路又可以进一步分为脉冲异步时序逻辑电路和电平异步时序逻辑电路。前面已对同步时序逻辑电路进行了系统介绍,本章讨论异步时序逻辑电路,并在介绍异步时序逻辑电路特点的基础上,重点讨论两类异步时序逻辑电路的分析与设计方法。

6.1 异步时序逻辑电路的特点与分类

在同步时序逻辑电路中,存储元件采用时钟控制触发器,电路中各触发器的时钟控制端与统一的时钟脉冲(CP)相连接,仅当时钟脉冲作用时,电路状态才能发生变化,改变后的状态一直保持到下一个时钟脉冲到来之时(在此期间不受外部输入变化的影响)。换言之,由时钟脉冲信号决定电路状态转换时刻并实现“等状态时间”,整个电路在时钟脉冲作用下由一个稳定状态转移到另一个稳定状态。正因为时钟脉冲对电路的控制作用,所以不论输入信号是电平信号还是脉冲信号,对电路引起的状态响应都是相同的。因此,在研究同步时序逻辑电路时,没有对输入信号的形式加以区分。

此外,该类电路除了对时钟脉冲的宽度和周期有一定要求外,对输入信号的变化过程没有加任何约束。

异步时序逻辑电路的工作特点如下:

电路中没有统一的时钟脉冲信号同步,电路状态的改变是外部输入信号变化直接作用的结果;在状态转移过程中,各存储元件的状态变化不一定发生在同一时刻,不同状态的维持时间不一定相同,并且可能出现非稳定状态;在研究异步时序逻辑电路时,对输入信号的形式有所区分,无论输入信号是脉冲信号还是电平信号,对其变化过程均有一定约束。

根据电路结构模型和输入信号形式的不同,异步时序逻辑电路可分为脉冲异步时序逻辑电路和电平异步时序逻辑电路两种类型。

脉冲异步时序逻辑电路的存储电路由触发器组成(可以是时钟控制触发器或者非时钟控制触发器),电路输入信号为脉冲信号;电平异步时序逻辑电路的存储电路由延迟元件组成(可以是专用的延迟元件或者利用电路本身固有的延迟),通过延迟加反馈实现记忆功能,电路输入信号为电平信号。

根据电路输出是否与输入直接相关,两类异步时序逻辑电路均可分为 Mealy 型和 Moore 型两种不同的模型。

6.2 脉冲异步时序逻辑电路

6.2.1 脉冲异步时序逻辑电路的结构模型

脉冲异步时序逻辑电路的结构模型如图 6.1 所示。

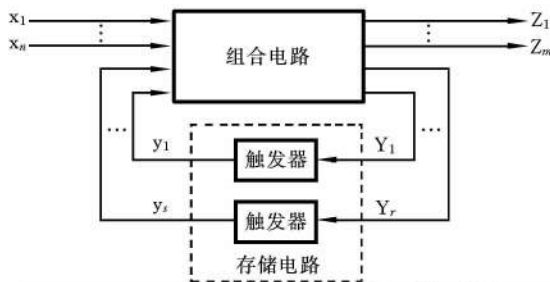


图 6.1 脉冲异步时序电路的结构模型

在脉冲异步时序逻辑电路中，引起触发器状态变化的脉冲信号是由输入端直接提供的。为了保证电路可靠地工作，输入脉冲信号必须满足如下约束条件：

- ① 输入脉冲的宽度，必须保证触发器可靠翻转；
- ② 输入脉冲的间隔，必须保证前一个脉冲引起的电路响应完全结束后，后一个脉冲才能到来；
- ③ 不允许在两个或两个以上输入端同时出现脉冲。因为客观上两个或两个以上脉冲是不可能准确地“同时”的，在没有时钟脉冲同步的情况下，由不可预知的时间延迟造成的微小时差，可能导致电路产生错误的状态转移。

此外，在脉冲异步时序逻辑电路中，Mealy 型和 Moore 型电路的输出信号会有所不同。对于 Mealy 型电路来说，由于输出不仅是状态变量的函数，而且是输入的函数，所以，输出通常是脉冲信号；而对于 Moore 型电路来说，由于输出仅仅是状态变量的函数，所以输出是电平信号，输出电平的值被定义在两个间隔不定的输入脉冲之间，即由两个输入脉冲之间的状态决定。

6.2.2 脉冲异步时序逻辑电路的分析

脉冲异步时序逻辑电路的分析方法与同步时序逻辑电路大致相同。分析过程中同样采用状态表、状态图、时间图等作为工具，分析步骤如下：

- ① 写出电路的输出函数和激励函数表达式；
- ② 列出电路次态真值表或次态方程组；
- ③ 作出状态表和状态图；
- ④ 画出时间图并用文字描述电路的逻辑功能。

显然，脉冲异步时序逻辑电路分析步骤与同步时序逻辑电路的完全相同。但是，由于脉冲异步时序逻辑电路没有统一的时钟脉冲以及对输入信号的约束，因此，在具体步骤的实施上是有区别的。其差别主要表现为两点。第一，当存储元件采用时钟控制触发器时，应将触发器的时钟控制端作为激励函数处理。分析时应特别注意触发器时钟端何时脉冲作用，仅当时钟端有脉冲

作用时,才根据触发器的输入确定状态转移方向,否则,触发器状态不变。若采用非时钟控制触发器,则应注意作用到触发器输入端的脉冲信号。第二,由于不允许两个或两个以上输入端同时出现脉冲,加之输入端无脉冲出现时,电路状态不会发生变化,因此,分析时可以排除这些情况,从而使分析过程中使用的图、表可以简化。具体地说,对 n 个输入端的一位输入,只需考虑各自单独出现脉冲的 n 种情况,而不像同步时序逻辑电路中那样需要考虑 2^n 种情况。例如,假定电路有 x_1 、 x_2 和 x_3 共 3 个输入,并用取值 1 表示有脉冲出现,则一位输入允许的取值只有 000、001、010、100 共 4 种,分析时需要讨论的只有后 3 种情况。下面举例说明脉冲异步时序逻辑电路的分析方法。

例 6.1 分析图 6.2 所示脉冲异步时序逻辑电路,指出该电路功能。

解 该电路由两个 J-K 触发器和一个与门组成,有一个输入端 x 和一个输出端 Z ,输出是输入和状态的函数,属于 Mealy 型脉冲异步时序逻辑电路。

① 写出输出函数和激励函数表达式。

$$Z = xy_2y_1$$

$$J_2 = K_2 = 1 \quad C_2 = y_1$$

$$J_1 = K_1 = 1 \quad C_1 = x$$

② 列出电路次态真值表。由于电路中的两个 J-K 触发器没有统一的时钟脉冲控制,所以,分析电路状态转移时,应特别注意各触发器时钟端何时由脉冲作用。J-K 触发器的状态转移发生在时钟端脉冲负跳变的瞬间,在次态真值表中用“ \downarrow ”表示。仅当时钟端有“ \downarrow ”出现时,相应触发器状态才能发生变化,否则状态不变。据此,可列出该电路的次态真值表,如表 6.1 所示。表中, x 为 1 表示输入端有脉冲出现,考虑到输入端无脉冲出现时电路状态不变,故省略了 x 为 0 的情况。

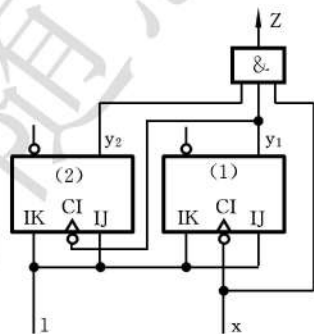


图 6.2 逻辑电路

表 6.1 次态真值表

输入	现态	激励函数							次态	
x	y_2	y_1	J_2	K_2	C_2	J_1	K_1	C_1	y_2^{n+1}	y_1^{n+1}
1	0	0	1	1		1	1	\downarrow	0	1
1	0	1	1	1	\downarrow	1	1	\downarrow	1	0
1	1	0	1	1		1	1	\downarrow	1	1
1	1	1	1	1	\downarrow	1	1	\downarrow	0	0

③ 作出状态表和状态图。根据表 6.1 所示次态真值表和输出函数表达式,可作出该电路的状态表如表 6.2 所示,状态图如图 6.3 所示。

表 6.2 状态表

现态	次态 $y_2^{n+1}y_1^{n+1}$ / 输出 Z		
$y_2 \quad y_1$	$x=1$		
0 0	0 1	/	0
0 1	1 0	/	0
1 0	1 1	/	0
1 1	0 0	/	1

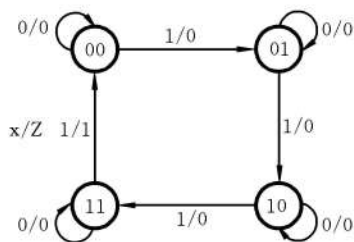


图 6.3 状态图

④ 画出时间图并说明电路逻辑功能。为了进一步描述该电路在输入脉冲作用下的状态和输出变化过程,可根据状态表或状态图画出该电路的时间图,如图 6.4 所示。

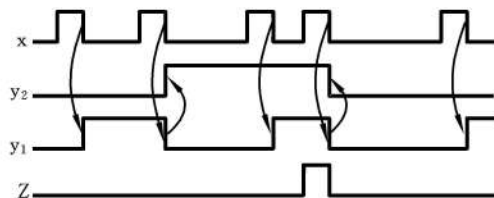


图 6.4 时间图

由状态图和时间图可知,该电路是一个模 4 加 1 计数器,当收到第四个输入脉冲时,电路产生一个进位输出脉冲。

例 6.2 分析图 6.5 所示脉冲异步时序逻辑电路。

解 该电路的存储电路部分由两个与非门构成的基本 R-S 触发器组成。电路有 3 个输入端 x_1 、 x_2 和 x_3 , 一个输出端 Z, 输出 Z 是状态变量的函数, 属于 Moore 型脉冲异步时序电路。

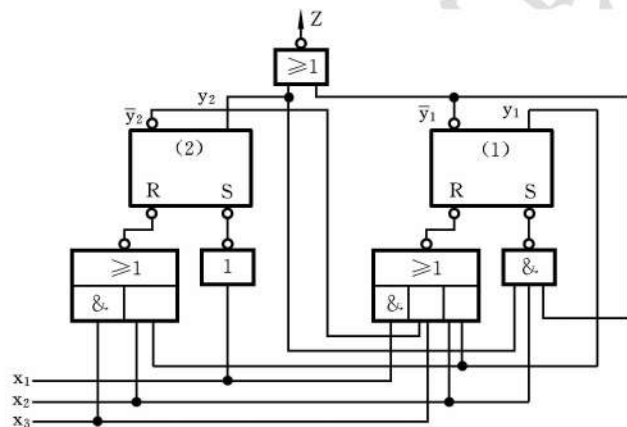


图 6.5 逻辑电路

① 写出输出函数和激励函数表达式。

$$\begin{aligned} Z &= y_2 + \bar{y}_1 = \bar{y}_2 y_1 & R_2 &= \bar{x}_3 + x_2 y_1 & S_2 &= \bar{x}_1 \\ R_1 &= x_1 + x_3 \bar{y}_2 + x_2 y_1 & S_1 &= x_2 y_2 \bar{y}_1 \end{aligned}$$

② 列出电路次态真值表。根据激励函数表达式和 R-S 触发器的功能表,可列出电路的次态真值表,如表 6.3 所示。

表 6.3 次态真值表

输 入			现 态		激励函数				次 态	
x_1	x_2	x_3	y_2	y_1	R_2	S_2	R_1	S_1	y_2^{n+1}	y_1^{n+1}
1	0	0	0	0	1	0	0	1	1	0
1	0	0	0	1	1	0	0	1	1	0
1	0	0	1	0	1	0	0	1	1	0
1	0	0	1	1	1	0	0	1	1	0
0	1	0	0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	0	1	0	0
0	1	0	1	0	1	1	1	0	1	1
0	1	0	1	1	0	1	0	1	0	0
0	0	1	0	0	0	1	0	1	0	0
0	0	1	0	1	0	1	0	1	0	0
0	0	1	1	0	0	1	1	1	0	0
0	0	1	1	1	0	1	1	1	0	1

③ 作出状态表和状态图。根据表 6.3 和电路输出函数表达式,可作出该电路的状态表,如表 6.4 所示;状态图如图 6.6 所示。

表 6.4 状态表

现 态		次态 $y_2^{n+1} y_1^{n+1}$			输出
		x_1	x_2	x_3	
0	0	10	00	00	0
0	1	10	00	00	1
1	0	10	11	00	0
1	1	10	00	01	0

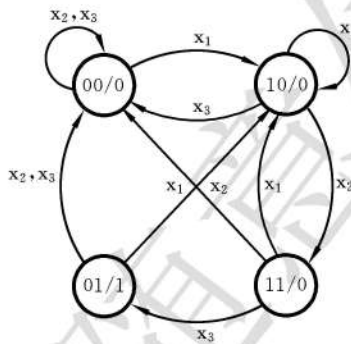


图 6.6 状态图

④ 画出时间图并说明电路功能。假定输入端 x_1 、 x_2 、 x_3 出现脉冲的顺序依次为 $x_1 - x_2 - x_1 - x_3 - x_1 - x_2 - x_3 - x_1 - x_3 - x_2$, 根据状态表或状态图可作出时间图,如图 6.7 所示。图中,假定电路状态转换发生在输入脉冲作用结束时,因此,转换时刻与脉冲后沿对齐。

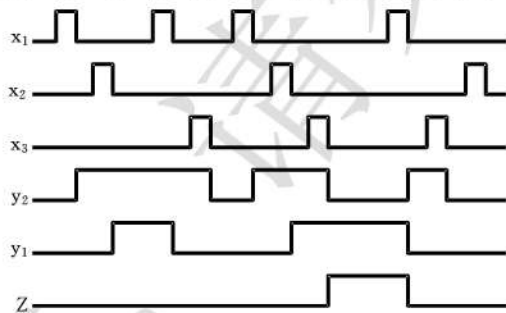


图 6.7 时间图

由状态图和时间图可知,在该电路中,当 3 个输入端按 x_1 、 x_2 、 x_3 的顺序依次出现脉冲时,产生一个“1”输出信号,其他情况下输出为“0”。因此,该电路是一个“ $x_1 - x_2 - x_3$ ”序列检测器。

6.2.3 脉冲异步时序逻辑电路的设计

脉冲异步时序逻辑电路设计的一般过程与同步时序逻辑电路设计大体相同。同样分为形成原始状态图和状态表、状态化简、状态编码、确定激励函数和输出函数、画逻辑电路图等步骤。但由于在脉冲异步时序逻辑电路中没有统一的时钟脉冲信号,以及对输入脉冲信号的约束,所以在某些步骤的处理细节上有所不同。

在脉冲异步时序逻辑电路设计时,主要应注意如下两点。

① 由于不允许两个或两个以上输入端同时为 1(用 1 表示有脉冲出现),所以在形成原始状态图和原始状态表时,若有多个输入信号,则只需考虑多个输入信号中仅一个为 1 的情况,从而使问题的描述得以简化。此外,在确定激励函数和输出函数时,可将两个或两个以上输入同时为 1 的情况,作为无关条件处理。无疑,这有利于函数的简化。

② 由于电路中没有统一的时钟脉冲,因此,当存储电路采用带时钟控制端的触发器时,触

发器的时钟端是作为激励函数处理的。这就意味着可以通过控制其时钟端输入脉冲的有、无来控制触发器的翻转或不翻转。基于这一思想,在设计脉冲异步时序逻辑电路时,可列出4种常用时钟控制触发器的激励表,如表6.5~表6.8所示。

表 6.5 D 触发器激励表

Q	Q^{n+1}	CP	D
0	0	d	0
		0	d
0	1	1	1
1	0	1	0
1	1	d	1
		0	d

表 6.6 J-K 触发器激励表

Q	Q^{n+1}	CP	J	K
0	0	d	0	d
		0	d	d
0	1	1	1	d
1	0	1	d	1
1	1	d	d	0
		0	d	d

表 6.7 T 触发器激励表

Q	Q^{n+1}	CP	T
0	0	d	0
		0	d
0	1	1	1
1	0	1	1
1	1	d	0
		0	d

表 6.8 R-S 触发器激励表

Q	Q^{n+1}	CP	R	S
0	0	d	d	0
		0	d	d
0	1	1	0	1
1	0	1	1	0
1	1	d	0	d
		0	d	d

从表6.5~表6.8可知,在要求触发器状态保持不变时,有两种不同的处理方法:一是令CP为d,输入端取相应值;二是令CP为0,输入端取任意值。例如,当要使D触发器维持0不变时,可令CP为d,D为0;也可令CP为0,D为d。显然,这将使激励函数的确定变得更加灵活,究竟选择哪种处理方法,应看怎样更有利于电路简化。一般选CP为0,输入任意,因为这样显得更清晰。

下面,举例说明异步时序逻辑电路设计的方法和步骤。

例 6.3 用D触发器作为存储元件,设计一个“ $x_1-x_2-x_2$ ”序列检测器。该电路有两个输入 x_1 和 x_2 ,一个输出Z。仅当 x_1 输入一个脉冲后, x_2 连续输入两个脉冲时,输出端Z由0变为1,该1信号将一直维持到输入端 x_1 或 x_2 再出现脉冲时才由1变为0。其输入/输出时间图如图6.8所示。

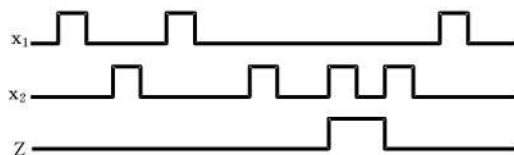


图 6.8 时间图

解 由题意可知,该序列检测器为 Moore 型脉冲异步时序逻辑电路。

① 作出原始状态图和原始状态表。设初始状态为 A,根据题意可作出原始状态图,如图6.9所示;原始状态表如表6.9所示。为了清晰起见,图、表中用 x_1 表示 x_1 端有脉冲输入, x_2 表示 x_2 端有脉冲输入。

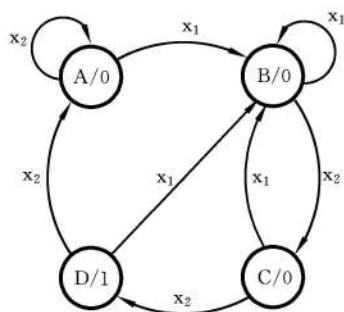


图 6.9 状态图

表 6.9 状态表

现 态	次 态		输 出 Z
	x_1	x_2	
A	B	A	0
B	B	C	0
C	B	D	0
D	B	A	1

② 状态化简。用隐含表法检查表 6.9 所示状态表,可知该状态表中的状态均不等效,即已为最简状态表。

③ 状态编码。由于最简状态表中有 4 个状态,故需用两位二进制代码表示。设状态变量用 y_2 、 y_1 表示,根据相邻编码法的原则,可采用表 6.10 所示编码方案。并由表 6.9、表 6.10 可得到二进制状态表,如表 6.11 所示。

表 6.10 编码方案

状 态	编 码	
	y_2	y_1
A	0	0
B	1	0
C	0	1
D	1	1

表 6.11 二进制状态表

现 态	次态 $y_2^{n+1} y_1^{n+1}$		输 出 Z
	y_2	y_1	
0 0	10	00	0
0 1	10	11	0
1 0	10	01	0
1 1	10	00	1

④ 确定输出函数和激励函数。假定次态与现态相同时,D 端取值随意,时钟端取值为 0;次态与现态不同时,D 端取值与次态相同,时钟端取值为 1(有脉冲出现)。根据表 6.11 所示状态表,可得到激励函数和输出函数真值表,如表 6.12 所示。

表 6.12 激励函数和输出函数真值表

输 入		现 态		激 励 函 数				输出函数
x_2	x_1	y_2	y_1	C_2	D_2	C_1	D_1	Z
0	1	0	0	1	1	0	d	0
		0	1	1	1	1	0	0
		1	0	0	d	0	d	0
		1	1	0	d	1	0	1
1	0	0	0	0	d	0	d	0
		0	1	1	1	0	d	0
		1	0	1	0	1	1	0
		1	1	1	0	1	0	1

令输入端无脉冲出现时,各触发器时钟端为 0,输入端取任意值“d”,并将两个输入端同时为 1(不允许)作为无关条件处理,可得到激励函数和输出函数卡诺图,如图 6.10 所示。

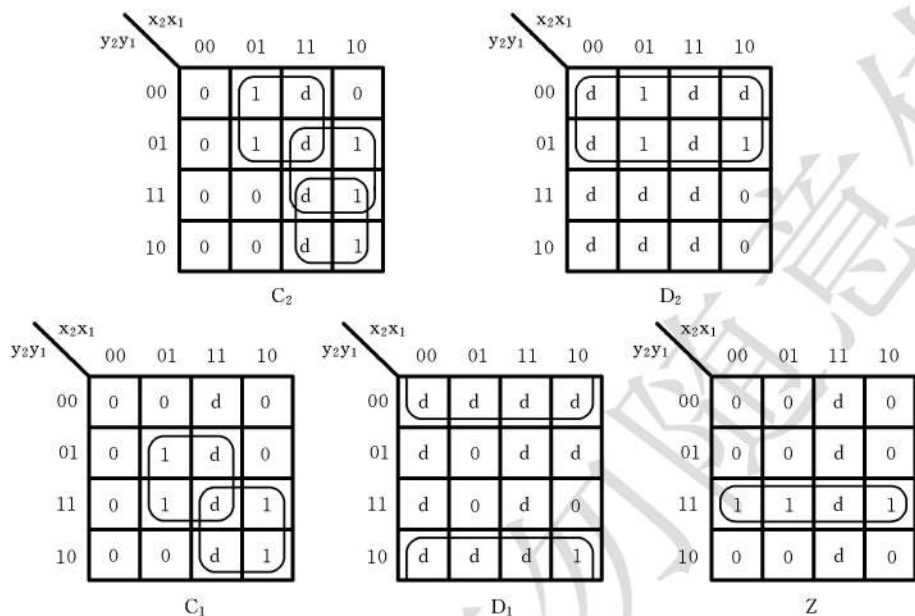


图 6.10 卡诺图

用卡诺图化简后的激励函数和输出函数如下：

$$C_2 = x_1 \bar{y}_2 + x_2 y_1 + x_2 y_2 \quad D_2 = \bar{y}_2$$

$$C_1 = x_1 y_1 + x_2 y_2 \quad D_1 = \bar{y}_1$$

$$Z = y_2 y_1$$

⑤ 画出逻辑电路图。根据激励函数和输出函数表达式，可画出该序列检测器的逻辑电路，如图 6.11 所示。

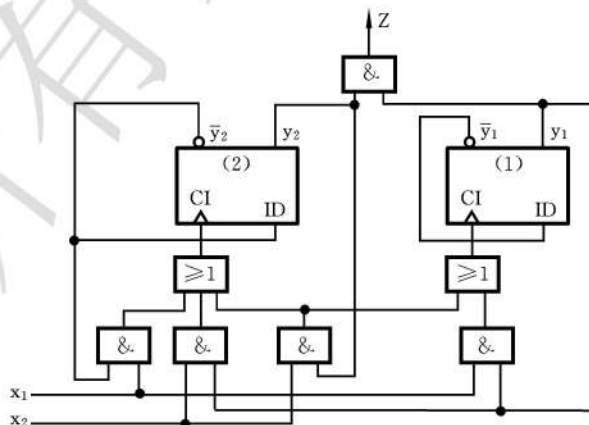


图 6.11 逻辑电路

例 6.4 用 T 触发器作为存储元件，设计一个异步模 8 加 1 计数器，该电路对输入端 x 出现的脉冲进行计数，当收到第八个脉冲时，输出端 Z 产生一个进位输出脉冲。

解 该电路的状态数目和状态转移关系均非常清楚，故可直接作出二进制状态图和状态表。并由题意可知，电路模型为 Mealy 型。

① 作出状态图和状态表。

设电路初始状态为“000”，状态变量用 y_3 、 y_2 、 y_1 表示，根据题意可作出二进制状态图，如图 6.12 所示；二进制状态表如表 6.13 所示。

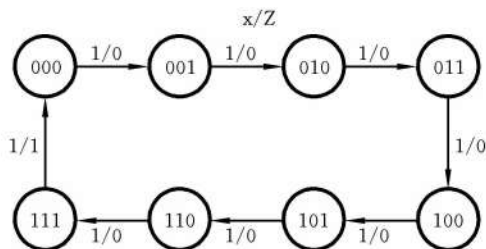


图 6.12 状态图

表 6.13 二进制状态表

现 态 $y_3 y_2 y_1$	次态 $y_3^{n+1} y_2^{n+1} y_1^{n+1}$ / 输出 Z			
	x=1			
0 0 0	0	0	1	0
0 0 1	0	1	0	0
0 1 0	0	1	1	0
0 1 1	1	0	0	0
1 0 0	1	0	1	0
1 0 1	1	1	0	0
1 1 0	1	1	1	0
1 1 1	0	0	0	1

② 确定激励函数和输出函数。

在例 6.3 所采用的设计方法中，规定 x 为 0 (无脉冲输入) 时，各触发器时钟端为 0 (无触发脉冲)，输入端任意； x 为 1 (有脉冲输入) 时，则根据触发器状态是否需要改变，有选择地确定其时钟端为 1 或者为 0。从而，在一定现态下某些触发器的时钟端会随着输入脉冲的出现而产生一个触发脉冲，使触发器的状态发生设计者规定的转移。显然，这样处理的结果必然使触发器的时钟端和输入脉冲信号相关。事实上，当电路中采用边沿触发器作为存储元件时，触发器时钟端所要求的跳变信号除了来自输入脉冲外，还可以由电路中触发器状态的改变产生，这样处理的结果通常可以使电路更简单。据此，可对模 8 加 1 计数器的激励函数按如下方法进行处理。

假定采用下降沿触发 (又称为负沿触发，用 \downarrow 表示) 的 T 触发器作为存储元件，根据表 6.13 所示状态表，可确定在输入脉冲作用下的状态转移关系，以及激励函数和输出函数真值表，如表 6.14 所示。

表 6.14 状态转移关系及激励函数、输出函数真值表

输入脉冲	现态	次态	状态跳变			激励函数						输出
x	$y_3 y_2 y_1$	$y_3^{n+1} y_2^{n+1} y_1^{n+1}$	y_3	y_2	y_1	C_3	T_3	C_2	T_2	C_1	T_1	Z
1(\downarrow)	0 0 0	0 0 1			\uparrow		d		d	\downarrow	1	0
1(\downarrow)	0 0 1	0 1 0		\uparrow	\downarrow		d	\downarrow	1	\downarrow	1	0
1(\downarrow)	0 1 0	0 1 1			\uparrow		d		d	\downarrow	1	0
1(\downarrow)	0 1 1	1 0 0	\uparrow	\downarrow	\downarrow	\downarrow	1	\downarrow	1	\downarrow	1	0
1(\downarrow)	1 0 0	1 0 1			\uparrow		d		d	\downarrow	1	0
1(\downarrow)	1 0 1	1 1 0		\uparrow	\downarrow		d	\downarrow	1	\downarrow	1	0
1(\downarrow)	1 1 0	1 1 1			\uparrow		d		d	\downarrow	1	0
1(\downarrow)	1 1 1	0 0 0	\downarrow	\downarrow	\downarrow	\downarrow	1	\downarrow	1	\downarrow	1	1

由表 6.14 可以看出，在输入脉冲作用下，状态转移过程中激励函数 C_3 所要求的触发信号“ \downarrow ”正好与 y_2 端产生的下跳变信号“ \downarrow ”一致，可由 y_2 端提供；激励函数 C_2 所要求的触发信号“ \downarrow ”正好与 y_1 端产生的下跳变信号“ \downarrow ”一致，可由 y_1 端提供；激励函数 C_1 所要求的触发信号“ \downarrow ”则可由 x 端的输入脉冲后沿提供；激励函数 T_3 、 T_2 、 T_1 的取值均为“1”或者任意值“d”。

而在输入端 x 为 0 (无脉冲输入) 时, 电路状态不变, 可令各触发器时钟端无下跳变信号“↓”出现, 输入端 T 为任意值“d”。据此, 可得到激励函数和输出函数表达式如下:

$$\begin{aligned} C_3 &= y_2 & C_2 &= y_1 & C_1 &= x \\ T_3 &= T_2 = T_1 & & & & Z = xy_3y_2y_1 \end{aligned}$$

③ 画出逻辑电路图。

根据激励函数和输出函数表达式, 可画出实现给定要求的逻辑电路, 如图 6.13 所示。

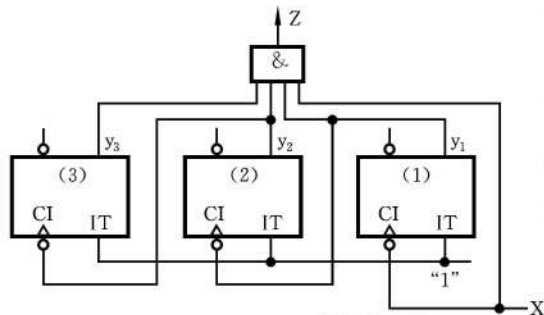


图 6.13 逻辑电路

思考: 假定采用上升沿触发 (又称为正沿触发, 用 ↑ 表示) 的 T 触发器作为存储元件, 则激励函数 C_3 、 C_2 、 C_1 应作何改变?

6.3 电平异步时序逻辑电路

6.3.1 电平异步时序逻辑电路的结构模型与描述方法

前面讨论的脉冲异步时序逻辑电路和同步时序逻辑电路有两个共同的特点。第一, 电路状态的转换是在脉冲作用下实现的。在同步时序逻辑电路中尽管输入信号可以是电平信号或者脉冲信号, 但电路的状态转换受统一的时钟脉冲控制; 脉冲异步时序逻辑电路没有统一的时钟脉冲, 因此, 规定输入信号为脉冲信号, 即控制电路状态转换的脉冲信号是由电路输入端提供的。第二, 电路对过去输入信号的记忆是由触发器实现的。在同步时序逻辑电路中采用带时钟控制端的触发器; 而在脉冲异步时序逻辑电路中既可用带时钟控制端的触发器, 也可用非时钟控制触发器。

事实上, 脉冲信号只不过是电平信号的一种特殊形式。所谓电平信号是指信号的“0”值和“1”值的持续时间是随意的, 它以电位的变化作为信号的变化。而脉冲信号的“1”值仅仅维持一个固定的短暂时刻, 它以脉冲的有、无标志信号的变化。显然, 电平信号在短时间内连续两次变化便形成了脉冲。至于电路中的触发器, 则不管是哪种类型, 都是由逻辑门加反馈回路构成的。将上述两个特点进一步推广到一般, 便可得到时序逻辑电路中更为本质的另一类电路——电平异步时序逻辑电路。

1. 电平异步时序逻辑电路的结构模型

电平异步时序逻辑电路同样由组合电路和存储电路两部分组成, 但存储电路是由反馈回路中的延迟元件构成的。延迟元件一般不用专门插入延迟线, 而是利用组合电路本身固有的

分布延迟在反馈回路中的“集总”。其一般结构模型如图 6.14 所示。

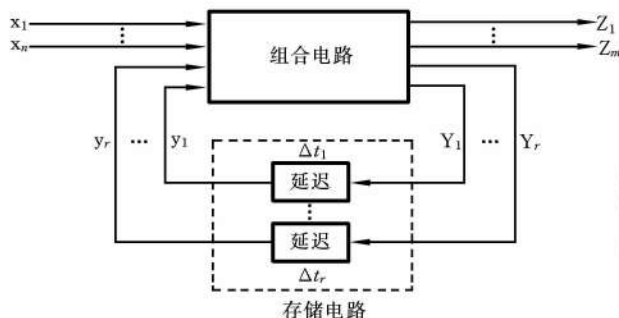


图 6.14 电平异步时序逻辑电路的结构模型

图中, x_1, x_2, \dots, x_n 为外部输入信号; Z_1, Z_2, \dots, Z_m 为外部输出信号; Y_1, Y_2, \dots, Y_r 为激励状态; y_1, y_2, \dots, y_r 为二次状态; $\Delta t_1, \Delta t_2, \dots, \Delta t_r$ 为反馈回路中的时间延迟。电路可用以下方程组描述:

$$\begin{aligned} Z_i &= f_i(x_1, \dots, x_n, y_1, \dots, y_r) & i &= 1, \dots, m \\ Y_j &= g_j(x_1, \dots, x_n, y_1, \dots, y_r) & j &= 1, \dots, r \\ y_j(t + \Delta t_j) &= Y_j(t) \end{aligned}$$

由图 6.14 所示的结构模型及相应方程组可知, 电路具有如下特点:

① 电路输出和状态的变化是由输入电平信号的变化直接引起的, 由于电平异步时序逻辑电路可以及时地对输入信号的变化作出响应, 所以工作速度较高。

② 电路的二次状态和激励状态仅仅相差一个时间延迟, 即二次状态 y 是激励状态 Y 经过 Δt 延迟后的“重现”, 因此, y 被命名为二次状态。当输入信号不变时, 激励状态与二次状态相同, 即 $y = Y$, 此时电路处于稳定状态。

③ 输入信号的一次变化可能会引起二次状态的多次变化。当电路处在稳定状态下输入信号发生变化时, 若激励状态 Y 的值与二次状态 y 的值是相同的, 则电路处于稳定状态; 若激励状态 Y 的值与二次状态 y 的值不同, 则变化后的 Y 经过 Δt 延迟后形成新的二次状态 y 反馈到组合电路输入端, 这个新的二次状态 y 又会引起输出 Z 和激励状态 Y 的变化, 这是一个循环过程, 该过程将一直进行到激励状态 Y 等于二次状态 y 为止。在变化过程终止前, 电路处于不稳定状态; 变化过程结束后, 电路进入一个新的稳定状态。这一现象, 是电平异步时序逻辑电路的一个重要特征。

2. 输入信号的约束

考虑到电平异步时序逻辑电路输入信号的变化将直接引起输出和状态的变化, 为了保证电路可靠地工作, 对输入信号有如下两条约束。

① 不允许两个或两个以上输入信号同时发生变化。因为客观上不可能有准确的“同时”, 而微小的时差都可能使最终到达的状态不确定。

② 输入信号变化引起的电路响应必须完全结束后, 才允许输入信号再次变化。换句话说, 必须使电路进入稳定状态后, 才允许输入信号发生变化。

以上两条是使电平异步时序逻辑电路能可靠工作的基本条件, 通常将满足上述条件的工作方式称为基本工作方式, 将按基本工作方式工作的电平异步时序逻辑电路称为基本型电路。

3. 描述方法

由于电平异步时序逻辑电路的组成与同步时序逻辑电路和脉冲异步时序逻辑电路不同,因此,电路的分析和设计方法以及分析和设计中使用的描述工具也不相同。在电平异步时序逻辑电路中,除了逻辑方程外,一般使用流程表和总态图描述一个电路的工作过程和逻辑功能。

流程表是用来反映电路输出信号、激励状态与电路输入信号、二次状态之间关系的一种表格形式。其一般格式如表 6.15 和表 6.16 所示。

表 6.15 Mealy 型流程表格式

二次 状态	激励状态/输出		
		输入 x	
y		Y/Z	

表 6.16 Moore 型流程表格式

二次 状态	激励状态		输出
		输入 x	
y		Y	Z

在构造流程表时,为了能够明显地区分电路的稳态和非稳态,当表中的激励状态与其对应的二次状态相同时,将激励状态加上圆圈,以表示电路处于稳态,否则,表示电路处于非稳态。其次,为了更好地体现不允许两个或两个以上输入信号同时变化的约束,将一位输入的各种取值按代码相邻的关系排列(类似卡诺图),以表示输入信号只能在相邻位置上发生变化。

例如,图 6.15(a)所示的是一个用与非门构成的基本 R-S 触发器,假定在逻辑关系不变的前提下对该电路的器件和连线位置稍作变动,将从电路输入到输出的延迟时间集总成反馈回路中的延迟元件,并将延迟前的状态用激励状态 Y 表示,而将经过延迟后的状态用二次状态 y 表示,即可将该触发器变成图 6.15(b)所示电平异步时序逻辑电路的结构模型。

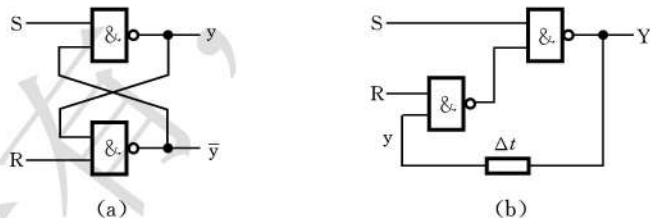


图 6.15 一个简单的电平异步时序逻辑电路

显然,基本 R-S 触发器是一个简单的电平异步时序逻辑电路,该电路的输出即状态,属于 Moore 型电平异步时序逻辑电路的特例。该电平异步时序逻辑电路的激励方程为

$$Y = \bar{S} + Ry$$

根据激励方程和与非门构成的 R-S 触发器不允许两个输入信号同时为 0 的约束,可作出相应流程表,如表 6.17 所示。

表 6.17 R-S 触发器流程表

二次状态 y	激励状态 Y				输出
	RS=00	RS=01	RS=11	RS=10	
0	d	①	①	1	0
1	d	0	①	①	1

由于电平异步时序逻辑电路在输入信号作用下存在稳态和非稳态,而且在同一种输入信号作用下,可能有一个稳态也可能有多个稳态,因此,为了对电路的工作状态和逻辑功能作出确切的说明,除了流程表和常用的时间图之外,引入了总态和总态图的概念。

总态是指电路输入和二次状态的组合,记作 (x,y) 。在流程表中,代表某个二次状态的一行和代表某种输入取值的一列的交叉点对应一个总态。当输入信号作相邻变化不引起电路状态变化时,在表内总态只作水平方向的移动。例如,表6.17中,当处在稳定总态 $(01,0)$,输入RS由01变为11时,总态沿水平方向移动,到达稳定总态 $(11,0)$,等待输入信号作新的变化。当输入信号作相邻变化引起状态改变时,总态先作水平移动,进入非稳定总态,然后再作垂直方向的移动,直至进入稳定总态为止。例如,表6.17中,当处在稳定总态 $(11,0)$,输入RS由11变为10时,总态先作水平移动,进入非稳定总态 $(10,0)$,由于此时激励状态由0变成了1,紧跟着二次状态将由0变为1,所以总态接着作垂直移动,进入稳定总态 $(10,1)$ 。

总态图是反映稳定总态之间转移关系及相应输出的一种有向图。一个电平异步时序逻辑电路的逻辑功能,是由该电路在输入作用下稳定状态之间的转移关系以及各时刻的输出来体现的。总态图能够清晰地描述一个电路的逻辑功能。表6.17所示流程表对应的总态图如图6.16所示。

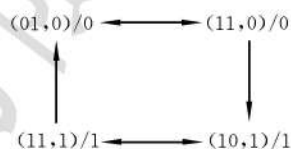


图 6.16 与表 6.17 对应的总态图

6.3.2 电平异步时序逻辑电路的分析

电平异步时序逻辑电路的分析过程比较简单,其一般步骤如下:

- ① 根据逻辑电路图写出输出函数和激励函数表达式;
- ② 作出流程表;
- ③ 作出总态图或时间图;
- ④ 说明电路逻辑功能。

下面举例说明电平异步时序逻辑电路的分析过程。

例 6.5 分析图 6.17 所示电平异步时序逻辑电路。

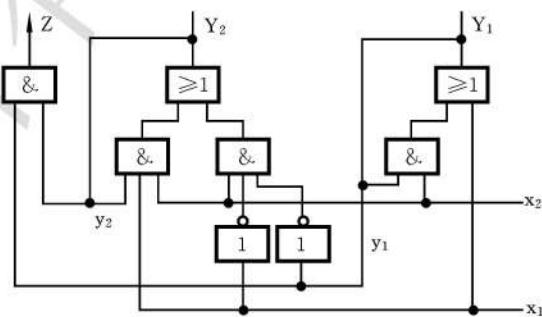


图 6.17 逻辑电路

解 该电路有两个外部输入 x_1 、 x_2 ; 两条反馈回路,对应的激励状态为 Y_1 、 Y_2 ,二次状态为 y_1 、 y_2 ; 一个外部输出 Z ,输出与输入没有直接关系,仅仅是二次状态的函数,所以,该电路为 Moore 模型。

- ① 写出输出函数和激励函数表达式

$$Z = y_2 y_1$$

$$Y_2 = x_2 x_1 y_2 + x_2 \bar{x}_1 \bar{y}_1$$

$$Y_1 = x_2 y_1 + x_1$$

② 作出流程表。根据激励函数和输出函数表达式可作出流程表,如表6.18所示。

表 6.18 流程表

二次状态 $y_2 \ y_1$	激励状态 $Y_2 Y_1$				输 出 Z
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$	
0 0	①①	01	01	10	0
0 1	00	①①	①①	①①	0
1 1	00	01	①①	01	1
1 0	00	01	11	①①	0

③ 作出总态图。根据流程表上稳定总态之间的关系,可作出图 6.18 所示的总态图。

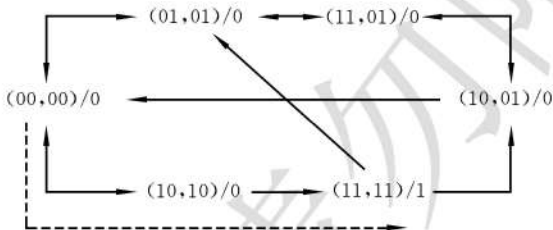


图 6.18 总态图

为了更直观地描述电路功能,还可以作出时间图。假定电路初始总态为 $(x_2 x_1, y_2 y_1) = (00, 00)$, 输入 $x_2 x_1$ 的变化序列为 $00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00 \rightarrow 01 \rightarrow 11 \rightarrow 10$, 根据流程表可作出总态和输出响应序列如下:

时刻 t_i	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
输入 $x_2 x_1$	00	10	11	01	00	01	11	10
总态 $(x_2 x_1, y_2 y_1)$	(00,00)	(10,00) *	(11,10) *	(01,11) *	(00,01) *	(01,00) *	(11,01)	(10,01)
		(10,10)	(11,11)	(01,01)	(00,00)	(01,01)		
输出 Z	0	0	1	0	0	0	0	0

在总态响应序列中加“*”的表示是非稳定总态。根据以上总态和输出响应序列可作出时间图,如图 6.19 所示。

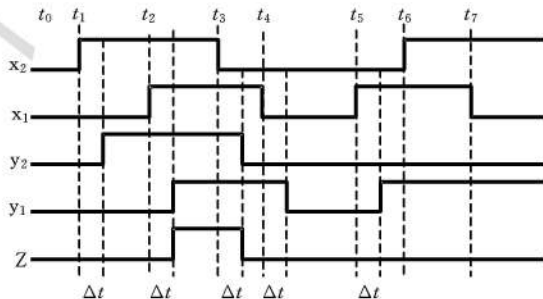


图 6.19 时间图

④ 说明电路功能。从总态图和时间图可以看出,仅当电路收到输入序列“ $00 \rightarrow 10 \rightarrow 11$ ”时,才产生一个高电平输出信号,其他情况下均输出低电平。因此,该电路是一个“ $00 \rightarrow 10 \rightarrow 11$ ”序列检测器。

6.3.3 电平异步时序逻辑电路的竞争

电平异步时序逻辑电路是利用各反馈回路的时间延迟实现记忆功能的。前面对电路进行分析时,没有对各反馈回路之间时间延迟的长短进行讨论,也就是说,是在假定各回路之间延迟时间相同的情况下对电路的工作过程进行分析的。事实上,各反馈回路的延迟时间往往各不相同。当电路中存在多条反馈回路,而各回路之间的延时又互不相同,则可能由于输入信号的变化在反馈回路之间引起竞争。这里的所谓竞争,是指当输入信号变化引起电路中两个或两个以上状态变量发生变化时,由于各反馈回路延迟时间的不同,使状态变量的变化有先有后而导致不同状态响应过程的现象。

根据竞争对电路状态转移产生的影响,可将竞争分为非临界竞争和临界竞争两种类型。若竞争的结果最终能到达预定的稳态,则称为非临界竞争;若竞争的结果可能使电路到达不同的稳态,即状态转移不可预测,则称为临界竞争。

例如,图 6.20 所示为某电平异步时序逻辑电路的结构框图,假定描述该电路的流程表如表 6.19 所示。

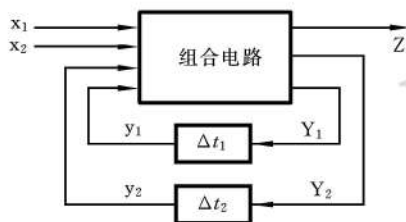


图 6.20 某电平异步时序电路框图

表 6.19 流程表

二次状态 $y_2 \quad y_1$	激励状态 $Y_2 Y_1$ / 输出 Z			
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$
0 0	00/0	00/0	01/0	11/0
0 1	00/0	01/0	01/0	01/0
1 1	00/0	00/0	10/0	11/0
1 0	00/0	00/0	10/1	10/0

从表 6.19 可以看出,当电路处于稳定总态(00,00)、输入 $x_2 x_1$ 由 00→10 时,电路应经过非稳定总态(10,00)到达稳定总态(10,11),由于此次输入变化引起激励状态 $Y_2 Y_1$ 从 00→11,即两个状态变量均发生变化,所以,当电路中两条反馈回路的延迟时间 Δt_1 和 Δt_2 不相等时,电路中 will 产生竞争。

此外,当电路处于稳定总态(10,11)、输入 $x_2 x_1$ 由 10→00 时,由于激励状态 $Y_2 Y_1$ 从 11→00,所以,电路同样可能发生竞争。

下面,按照两条反馈回路延迟时间 Δt_1 和 Δt_2 的大小关系,对上述两处输入信号变化引起的状态响应过程进行分析,讨论所存在的竞争各属于何种类型。

当电路处于稳定总态(00,00)、输入 $x_2 x_1$ 由 00→10 时,其状态响应过程如下。

① $\Delta t_2 = \Delta t_1$: 二次状态 y_2, y_1 将同时响应激励状态 Y_2, Y_1 的变化,即 $y_2 y_1$ 由 00→11,总态变化过程为(00,00)→(10,00)→(10,11),即到达预定的稳定总态(10,11)。

② $\Delta t_2 < \Delta t_1$: 二次状态 y_2 对激励状态 Y_2 的响应先于 y_1 对 Y_1 的响应,即 $y_2 y_1$ 将由 00→10,总态变化过程为(00,00)→(10,00)→(10,10),由于(10,10)是稳定总态,故电路停留在该稳定总态,即电路到达了一个非期望的稳定总态(10,10)。

③ $\Delta t_2 > \Delta t_1$: 二次状态 y_2 对激励状态 Y_2 的响应落后于 y_1 对 Y_1 的响应,即 $y_2 y_1$ 将由 00→01,总态变化过程为(00,00)→(10,00)→(10,01),由于(10,01)是稳定总态,故电路停留在该稳态,即电路到达了一个非期望的稳定总态(10,01)。

由此可见,此次输入信号变化,使电路最终到达的稳定状态随电路反馈回路中延迟时间的不同而不同,即状态转移不可预测,所以本次竞争为临界竞争。

当电路处于稳定总态(10,11)、输入 x_2x_1 由 10→00 时,其状态响应过程如下。

① $\Delta t_2 = \Delta t_1$: 二次状态 y_2, y_1 将同时响应激励状态 Y_2, Y_1 的变化,即 y_2y_1 由 11→00,总态变化过程为(10,11)→(00,11)→(00,00),到达预定的稳定总态(00,00)。

② $\Delta t_2 < \Delta t_1$: 二次状态 y_2 对激励状态 Y_2 的响应先于 y_1 对 Y_1 的响应,总态变化过程为(10,11)→(00,11)→(00,01)→(00,00),到达预定的稳定总态(00,00)。

③ $\Delta t_2 > \Delta t_1$: 二次状态 y_2 对激励状态 Y_2 的响应落后于 y_1 对 Y_1 的响应,总态变化过程为(10,11)→(00,11)→(00,10)→(00,00),到达预定的稳定总态(00,00)。

由此可见,无论反馈回路中延迟时间的大小如何,此次输入信号变化引起的竞争最终都能到达预定稳态,所以,本次竞争属于非临界竞争。

从上述分析不难得出用流程表检查电路竞争的一般法则:当从某一稳态出发,输入信号发生所允许的变化、引起两个或两个以上激励状态发生变化时,由于反馈回路之间延迟时间的不同会使电路产生竞争。若输入信号变化所到达的列只有一个稳态,则该竞争属非临界竞争;若输入信号变化所到达的列有两个或两个以上稳态,则该竞争属临界竞争。

显然,非临界竞争的存在不会影响电路的正常工作,但临界竞争的存在却将导致电路状态转换的不可预测。为了确保电平异步时序逻辑电路能可靠地实现预定功能,电路设计时必须避免发生临界竞争。该问题一般可在状态编码时解决,具体方法在 6.3.4 节中介绍。

* 6.3.4 电平异步时序逻辑电路的设计

电平异步时序逻辑电路设计的一般步骤如下:

- ① 根据设计要求,建立原始流程表;
- ② 化简原始流程表,得到最简流程表;
- ③ 状态编码,得到二进制流程表;
- ④ 确定激励状态和输出函数表达式;
- ⑤ 画出逻辑电路图。

下面对完成上述步骤的具体方法分别进行介绍。

1. 建立原始流程表

原始流程表是按照电平异步时序逻辑电路的描述方法对设计要求的一种最原始的抽象。为了实现从一个逻辑问题的文字描述到流程表的过渡,在建立原始流程表时通常借助时间图或原始总态图,即首先根据题意画出典型输入、输出时间图或作出原始总态图,然后再逐步形成原始流程表。

根据时间图建立原始流程表的过程如下。

(1) 画出典型输入/输出时间图并设立相应状态

画典型输入/输出时间图应注意以下 3 点:

- ① 符合题意,即正确体现设计要求;
- ② 满足电平异步时序逻辑电路不允许两个或两个以上输入信号同时改变的约束条件;
- ③ 尽可能反映输入信号在各种取值下允许发生的变化。

作出输入/输出时间图后,按输入信号的变化进行时间划分,将每次变化作为一个新的输入,用不同时刻进行区分。由于电平异步时序逻辑电路约定,每次输入信号变化,必须保证电路进入稳定状态后才允许输入信号再次变化,所以,应根据题意设立与各时刻输入/输出对应的稳定状态。

(2) 建立原始流程表

根据时间图和所设立的状态建立原始流程表,一般分为3步进行。

① 画出原始流程表,并填入稳定状态和相应输出。由于根据时间图设立状态时,开始并不知道哪些输入取值可用同一状态表示,因此,对不同的输入取值总是设立不同的状态进行区分,这就使得原始流程表中每一行只有一个稳定状态。显然,每设立一个状态,在原始流程表上便占一行,每行对应一个二次状态。根据每个状态设立时的输入值,在表中找到相应列,并在该总态中填入与二次状态相同的激励状态和相应输出,即可得到原始流程表的稳定状态部分。

② 填入非稳定状态并指定非稳定状态下的输出,完善流程表。由于表中每行只有一个稳定状态,所以,在稳态下输入信号发生允许变化时,电路不可能直接进入另一个稳态。假定每次输入信号发生变化时,电路总是经过一个非稳定状态后进入另一个稳定状态,根据时间图中的状态转移关系,可在原始流程表中填入相应的非稳定状态。

在填写非稳定状态时应注意,由于时间图中不一定反映了所有输入信号变化的情况,所以往往要根据题意作适当的补充和完善。当从某一稳态出发,输入信号发生允许变化所引起的状态转移,不能用时间图中所设立的状态来表示时,则应根据题意补充新的状态,以便无遗漏地反映设计要求。

为了使电路经过非稳定状态时,其输出不产生尖脉冲信号,规定非稳定状态下输出指定的法则为:若转换前后两个稳定状态的输出相同,则指定非稳定状态下的输出与稳态下的输出相同;若转换前后两个稳定状态的输出不同,则可指定非稳定状态下的输出为任意值“d”。

③ 填入无关状态和无关输出。因为不允许两个或两个以上输入信号同时改变,所以对稳态下输入不允许到达的列,在相应处填入任意状态和任意输出,用“d”表示,即作为无关处理。

至此,可得到一个完整的原始流程表。

例 6.6 某电平异步时序逻辑电路有两个输入端 x_1 和 x_2 , 一个输出端 Z 。输出与输入之间的关系为:只要 $x_1 x_2 = 00$, 则 $Z = 0$, 在此之后当 $x_1 x_2 = 01$ 或 10 时, $Z = 1$; 只要 $x_1 x_2 = 1$, 则 $Z = 1$, 在此之后当 $x_1 x_2 = 01$ 或 10 时, $Z = 0$ 。作出该电路的原始流程表。

解 根据借助时间图建立原始流程表的方法,形成该电路原始流程表的过程如下。

(1) 画出典型输入/输出时间图并设立相应状态

根据题意,可画出该电路典型输入/输出时间图,如图 6.21 所示。

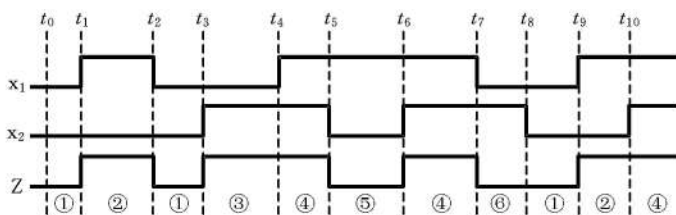


图 6.21 典型输入/输出时间图

图中,假定 t_0 为起始时刻,在该时刻输入 $x_1x_2=00$,输出 Z 为 0,用状态①表示;在 t_1 时刻,输入 x_1x_2 由 $00 \rightarrow 10$,输出 Z 为 1,用状态②表示;在 t_2 时刻,输入 x_1x_2 由 $10 \rightarrow 00$,输出 Z 为 0,因为任何时刻只要 $x_1x_2=00$,则输出 Z 为 0,故与 t_0 时刻相同,仍用状态①表示; t_3 时刻,输入 x_1x_2 由 $00 \rightarrow 01$,输出 Z 为 1,用状态③表示; t_4 时刻,输入 x_1x_2 由 $01 \rightarrow 11$,输出 Z 为 1,用状态④表示; t_5 时刻,输入 x_1x_2 由 $11 \rightarrow 10$,输出 Z 为 0,用状态⑤表示; t_6 时刻,输入 x_1x_2 由 $10 \rightarrow 11$,输出 Z 为 1,因为任何时刻只要 $x_1x_2=11$,则输出 Z 为 1,故与 t_4 时刻相同,仍用状态④表示; t_7 时刻,输入 x_1x_2 由 $11 \rightarrow 01$,输出 Z 为 0,用状态⑥表示; t_8 时刻,输入 x_1x_2 由 $01 \rightarrow 00$,输出 Z 为 0,与 t_0 时刻相同,用状态①表示; t_9 时刻,输入 x_1x_2 由 $00 \rightarrow 10$,输出 Z 为 1,与 t_1 时刻相同,用状态②表示; t_{10} 时刻,输入 x_1x_2 由 $10 \rightarrow 11$,输出 Z 为 1,与 t_4 时刻相同,用状态④表示。

(2) 建立原始流程图

① 画出原始流程图并填入稳定状态和相应输出。图 6.20 中共设立了 6 个稳定状态,按照每行一个稳定状态,将时间图中设立各状态和相应的输出填入流程表中与各二次状态、输入取值对应的行、列位置,即可得到表 6.20 所示的部分流程图(I)。

表 6.20 部分流程图(I)

二次状态 y	激励状态 Y /输出 Z			
	$x_1x_2=00$	$x_1x_2=01$	$x_1x_2=11$	$x_1x_2=10$
1	①/0			
2				②/1
3		③/1		
4			④/1	
5				⑤/0
6		⑥/0		

表 6.21 部分流程图(II)

二次状态 y	激励状态 Y /输出 Z			
	$x_1x_2=00$	$x_1x_2=01$	$x_1x_2=11$	$x_1x_2=10$
1	①/0	3/d		2/d
2	1/d		4/1	②/1
3	1/d	③/1	4/1	
4		6/d	④/1	5/d
5	1/0		4/d	⑤/0
6	1/0	⑥/0	4/d	

② 填入非稳定状态并指定非稳定状态下的输出,完善流程图。根据时间图中的状态转移关系和非稳定状态下输出指定的法则,在流程表中填入非稳定状态并指定其输出,即可得到表 6.21 所示的部分流程图(II)。在图 6.21 所示时间图中,未体现出电路处于稳定状态③、输入 x_1x_2 由 $01 \rightarrow 00$ 的情况,此时,可根据题意令其转向状态①;同样,当处于稳态⑤,输入 x_1x_2 由 $10 \rightarrow 00$ 时,可令其转向状态①;当处于稳态⑥,输入 x_1x_2 由 $01 \rightarrow 11$ 时,可令其转向状态④。由于时间图中设立的 6 个状态已能反映电路在各种输入取值作用下的状态响应,所以无需补充新的状态。

③ 填入无关状态和无关输出。对表 6.21 中各稳定状态下输入变化不允许到达的列,在相应位置填入无关状态和无关输出“d”,即可得到表 6.22 所示的完整流程图。

表 6.22 完整流程图

二次状态 y	激励状态 Y /输出 Z			
	$x_1x_2=00$	$x_1x_2=01$	$x_1x_2=11$	$x_1x_2=10$
1	①/0	3/d	d/d	2/d
2	1/d	d/d	4/1	②/1
3	1/d	③/1	4/1	d/d
4	d/d	6/d	④/1	5/d
5	1/0	d/d	4/d	⑤/0
6	1/0	⑥/0	4/d	d/d

2. 化简原始流程表

在建立原始流程表时,设计者一般将注意力集中在如何正确、清晰地描述给定的设计要求上,并没有刻意追求如何使用最少的状态,因而所得到的流程表往往不是最简的。在进行电平异步时序逻辑电路设计时,流程表中的状态数目决定了电路中反馈回路的数目。显然,状态数目的多少与电路的复杂程度直接相关。为了获得一种经济、合理的设计方案,必须对原始流程表进行化简,求出最简流程表。

原始流程表的化简是建立在状态相容这一概念基础之上的。由于原始流程表中的每一行代表一个稳定状态,因而相容状态的概念被引申为相容行的概念。

对于原始流程表中的某两行,如果每一列确定的输出相同,且确定的激励状态相同、交错、循环、相容或为各自本身,则称这两行为相容行。在检查输出时,对于一个确定而另一个任意,或者两个均任意的情况,都作为相同情况处理。在检查激励状态时,按以下原则确定稳定状态、非稳定状态和任意状态的相容性:

- ① 稳定状态①和非稳定状态 i 是相容的;
- ② 若稳定状态①和①是相容的,则稳定状态①和非稳定状态 j 是相容的;
- ③ 若稳定状态①和①是相容的,则非稳定状态 i 和 j 是相容的;
- ④ 稳定状态①和非稳定状态 i 均与任意状态“ d ”相容;任意状态“ d ”与任意状态“ d ”相容。

引入相容行的概念后,原始流程表的化简过程与不完全给定状态表的化简过程类似,同样可用隐含表、合并图和覆盖闭合表作为化简工具,其一般步骤如下:

- ① 作隐含表,找出相容行;
- ② 作合并图,求出最大相容行类;
- ③ 从相容行类中选择一个最小闭覆盖;
- ④ 作出最简流程表。

对最小闭覆盖中各相容行类中的相容行进行合并时注意,当输出存在确定值和任意值“ d ”时,合并后取确定值;当激励状态存在稳定状态和非稳定状态时,合并时取稳定状态;当激励状态存在确定状态和任意状态“ d ”时,合并时取确定状态。

下面举例说明原始流程表的化简过程。

例 6.7 化简表 6.22 所示原始流程表。

解 根据化简原始流程表的方法和步骤,化简过程如下。

① 作隐含表,找相容行。图 6.22 给出了与表 6.22 所示原始流程表对应的隐含表。根据相容行的判断规则,可找出相容行对:(1,2),(1,3),(2,3),(2,6),(3,5),(4,5),(4,6),(5,6)。

② 作合并图,求最大相容行类。根据所得出的相容行对,可作出合并图如图 6.23 所示。可见,最大相容行类为(1,2,3),(4,5,6),(3,5),(2,6)。

③ 选择一个最小闭覆盖。显然,选择由两个最大相容行类构成的集合{(1,2,3),(4,5,6)},便可满足覆盖、闭合和最小 3 个条件。因此,该集合即为表 6.22 所示原始流程表的最小闭覆盖。

④ 作出最简流程表。将最小闭覆盖中的最大相容类(1,2,3),(4,5,6)分别用 A、B 代替,即可得到最简流程表,如表 6.23 所示。

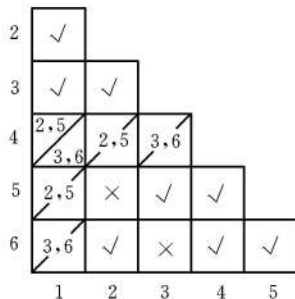


图 6.22 隐含表

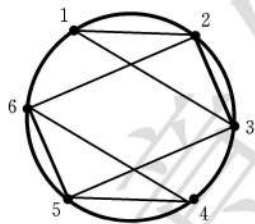


图 6.23 合并图

表 6.23 最简流程表

二次状态 y	激励状态 Y/输出 Z			
	$x_1 x_2 = 00$	$x_1 x_2 = 01$	$x_1 x_2 = 11$	$x_1 x_2 = 10$
A	Ⓐ/0	Ⓐ/1	B/1	Ⓐ/1
B	A/0	Ⓑ/0	Ⓑ/1	Ⓑ/0

3. 状态编码

状态编码的任务是根据化简后的状态数目确定二进制代码的位数，并选择一种合适的状态分配方案，将每个状态用一个二进制代码表示。在同步时序逻辑电路设计中，选择分配方案时需考虑的主要是如何使电路结构最简单。而在电平异步时序逻辑电路设计中，确定分配方案时应考虑的主要是如何避免反馈回路之间的临界竞争，保证电路可靠地实现预定功能。

为了消除临界竞争，在确定状态分配方案时常用以下几种方法。

(1) 相邻状态，相邻分配

由 6.3.3 节对电平异步时序逻辑电路中竞争现象的分析可知，仅当输入变化引起两个或两个以上状态变量发生变化时，电路中才会产生竞争。换言之，如果能保证每次状态转移时，仅有一个状态变量变化，则不会产生竞争。据此，可通过“相邻状态，相邻分配”的方法消除竞争。所谓相邻状态，是指稳态下输入取值作相邻变化时，需要直接发生转换的状态。而所谓相邻分配是指分配给相邻状态的代码为相邻代码（仅一位不同）。

为了找出流程表中各状态的相邻关系，通常借助状态相邻图。画状态相邻图的方法是：先将流程表中的每一个状态用一个圆圈表示（在圆圈内标出状态名），然后从流程表中每一个稳态出发，找出输入取值作相邻变化时的下一个稳态，并用有向线段将其连接起来，表示这两个状态为相邻状态。

例 6.8 对表 6.24 所示流程表进行状态编码，作出二进制流程表。

表 6.24 流程表

二次状态 y	激励状态 Y			
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$
A	Ⓐ	Ⓐ	B	C
B	A	Ⓑ	Ⓑ	Ⓑ
C	Ⓒ	A	D	Ⓒ
D	C	Ⓓ	Ⓓ	Ⓓ

解 根据“相邻状态,相邻分配”的法则,首先作出表 6.24 所示流程表的状态相邻图,如图 6.24 所示。由相邻图可知,A 和 B、A 和 C、C 和 D 为相邻状态,状态分配时应令其代码相邻。流程表中共有 4 个状态,需两位代码,设二次状态用 y_2 、 y_1 表示,可选择状态分配方案,如图 6.25 所示。即:用 00 表示 A,01 表示 B,10 表示 C,11 表示 D。

将表 6.24 中的状态用相应二进制编码表示,即可得到表 6.25 所示二进制流程表。由该流程表可知,在任一稳态下输入信号发生允许变化时,均不会引起两个状态变量发生变化,因而从根本上消除了竞争现象。

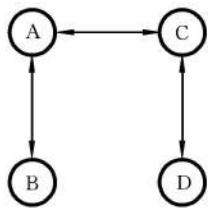


图 6.24 状态相邻图

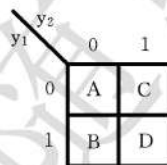


图 6.25 状态分配方案

表 6.25 二进制流程表

二次状态 $y_2 \ y_1$	激励状态 $Y_2 Y_1$			
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$
0 0	00	00	01	10
0 1	00	01	01	01
1 1	10	11	11	11
1 0	10	00	11	10

值得指出的是,并不是所有流程表都能直接用最少的二进制代码位数实现“相邻状态,相邻分配”。设状态数为 n ,二进制代码位数为 m ,则 n 和 m 的关系为 $2^m \geq n > 2^{m-1}$ 。由于一个 m 位代码最多只有 m 个相邻代码,因此,当相邻图上状态的最大相邻状态数 L 大于 m 时,则不可能用 m 位代码实现相邻分配。通常解决的办法是增加二进制代码位数,实现相邻分配。由于代码位数对应着电路中的反馈回路数,因此,这将增加电路的复杂性。

(2) 增加过渡状态,实现相邻分配

对于某些流程表,尽管相邻图上状态的最大相邻状态数 L 不大于状态分配的最小代码位数 m ,但状态之间的相邻关系出现由奇数个状态构成的闭环,因而无法直接实现状态的相邻分配。一种常用的方法是通过增加过渡状态,实现相邻分配,得到一个无竞争的二进制流程表。

例 6.9 对表 6.26 所示流程表进行状态编码,得到二进制流程表。

表 6.26 流程表

二次状态 y	激励状态 Y			
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$
A	Ⓐ	B	C	Ⓐ
B	A	Ⓑ	Ⓑ	C
C	A	B	Ⓒ	Ⓒ

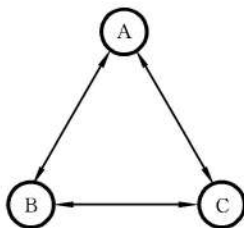


图 6.26 状态相邻图

解 根据表 6.26 所示流程表,可作出状态相邻图,如图 6.26 所示。尽管相邻图上每个状态只有两个相邻状态,但由于 3 个状态之间的相邻关系构成一个闭环,所以用两位代码无论怎样分配均无法满足其相邻关系。如果增加一个过渡状态,例如,在状态 A 和 C 之间增加过渡状态 D,将 $A \rightarrow C$ 改为 $A \rightarrow D \rightarrow C$, $C \rightarrow A$ 改为 $C \rightarrow D \rightarrow A$,那么,表 6.26 所示流程表可被修改成如表 6.27 所示。修改后的流程表中增加了新的一行,但该行没有稳定状态,因为状态 D 仅在稳态 A 和 C 发生转换时完成过渡作用。作出表 6.26 和表 6.27 的总态图(略)可以表明,增加过渡状态后的流程表与原流程表描述的逻辑功能相同。表 6.27 所示流程表的状态相邻图如图 6.27 所示。显然,用两位代码可以方便地满足图 6.27 所示的相邻关系。

表 6.27 增加过渡状态后的流程表

二次状态 y	激励状态 Y			
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$
A	Ⓐ	B	D	Ⓐ
B	A	Ⓑ	Ⓑ	C
C	D	B	Ⓒ	Ⓒ
D	A	d	C	d

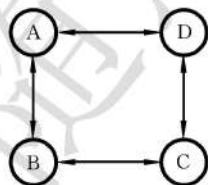


图 6.27 增加过渡状态后的状态相邻图

设二次状态用 y_2, y_1 表示,令 $y_2 y_1$ 取值 00 表示 A,01 表示 B,10 表示 D,11 表示 C,即可得到与表 6.27 对应的二进制流程表,如表 6.28 所示,该流程表描述的电路中不存在竞争。

(3) 允许非临界竞争,避免临界竞争

由于非临界竞争并不影响电路正常工作,所以,在进行状态分配时,只需避免临界竞争。对于有的流程表,虽然无法用最少位数的代码实现无竞争的状态分配,但可以通过将竞争限制在只有一个稳态的列,即允许非临界竞争,从而可以实现无临界竞争的状态分配。

表 6.28 二进制流程表

二次状态 $y_2 \quad y_1$		激励状态 $Y_2 Y_1$			
		$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$
0	0	Ⓐ	01	10	Ⓐ
0	1	00	Ⓑ	Ⓑ	11
1	1	10	01	Ⓒ	Ⓒ
1	0	00	dd	11	dd

例 6.10 对表 6.29 所示流程表进行状态编码,得到二进制流程表。

解 根据表 6.29 所示流程表可作出状态相邻图,如图 6.28 所示。显然,用两位二进制代码无法实现相邻状态相邻分配。解决的方法之一是通过增加过渡状态和增加代码位数实现相邻分配,但这样处理的结果必然增加电路的复杂性。解决该问题的另一种方法是允许非临界竞争,避免临界竞争。观察表 6.29 不难发现,状态 A 和 C 之间的转换只发生在 $x_2 x_1 = 00$ 和

表 6.29 流程表

二次状态 y	激励状态 Y				输出 Z
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$	
A	Ⓐ	C	D	Ⓐ	0
B	A	C	Ⓑ	Ⓑ	0
C	A	Ⓒ	Ⓒ	B	0
D	A	C	Ⓓ	A	1

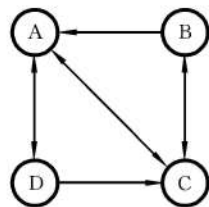


图 6.28 状态相邻图

$x_2x_1=01$ 这两列,而这两列各只有一个稳定状态,这就意味着 A 和 C 发生转换时,即使产生竞争也属于非临界竞争。即在状态分配时 A 和 C 可以不相邻。排除 A 和 C 的相邻关系后,状态编码只需满足 A 和 B、A 和 D、D 和 C、C 和 B 相邻即可。

设二次状态用 y_2, y_1 表示,令 y_2y_1 取值 00 表示 A, 01 表示 B, 10 表示 D, 11 表示 C, 将其代入表 6.29, 即可得到表 6.30 所示二进制流程表。该流程表描述的电路不会产生临界竞争。

表 6.30 二进制流程表

二次状态 $y_2 \quad y_1$	激励状态 Y_2Y_1				输 出 Z
	$x_2x_1=00$	$x_2x_1=01$	$x_2x_1=11$	$x_2x_1=10$	
0 0	00	11	10	00	0
0 1	00	11	01	01	0
1 1	00	11	11	01	0
1 0	00	11	10	00	1

4. 确定激励状态和输出函数表达式

二进制流程表给出了激励状态、输出函数与电路输入和二次状态之间的取值关系。根据流程表可作出激励状态、输出函数的卡诺图,化简后即可得到激励状态和输出函数的最简表达式。例如,根据表 6.30 可作出 Y_2 、 Y_1 和输出 Z 的卡诺图,如图 6.29 所示。

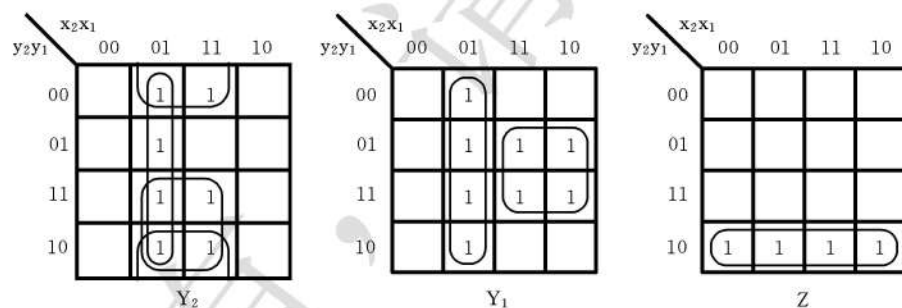


图 6.29 激励状态和输出函数卡诺图

化简后可得到激励状态和输出函数表达式为

$$Y_2 = \bar{x}_2x_1 + x_1y_2 + x_1\bar{y}_1$$

$$Y_1 = \bar{x}_2x_1 + x_2y_1 \quad Z = y_2\bar{y}_1$$

根据激励状态和输出函数表达式,即可画出相应逻辑电路图(略)。

前面对电平异步时序逻辑电路设计的各主要步骤和方法进行了讨论。为了使读者系统地掌握设计的全过程,下面给出一个设计实例。

例 6.11 用与非门设计一个单脉冲发生器,电路结构框图如图 6.30 所示。

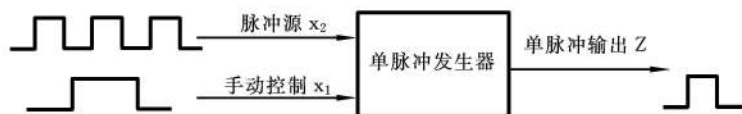


图 6.30 单脉冲发生器的结构框图

该电路有两个输入端 x_1 、 x_2 和一个输出端 Z。 x_2 接时钟脉冲源, x_1 接手动控制按钮。当不按按钮($x_1=0$)时, x_2 端的脉冲被封锁, 输出 Z 为 0, 无脉冲输出; 当按下按钮并释放(x_1 由

0→1再由1→0)之后,输入端 x_2 出现的第一个完整脉冲被送至输出端Z,即用手启动一次,输出一个完整脉冲。电路规定每启动一次,必须在输出一个完整脉冲后才可再次启动。

解 单脉冲发生器是一种在系统调试、维修、测试中常用的逻辑电路,主要用来控制系统运行于单步工作状态。根据给定要求,设计过程如下。

① 建立原始流程表。根据题意可作出典型输入/输出时间图,如图6.31所示。

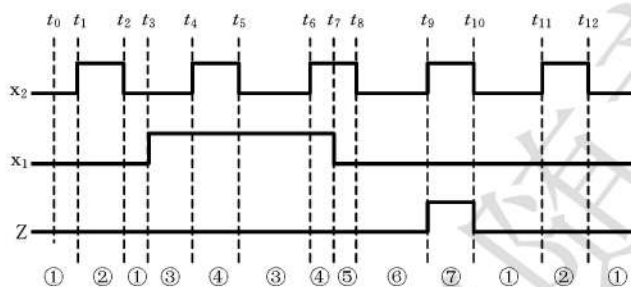


图 6.31 典型时间图

图中,按照输入信号的变化进行时间划分后,根据题意共设立了7个不同状态。其中, t_0 时刻 $x_2x_1=00$,启动信号和脉冲信号均未出现,输出Z为0,设用状态①表示; t_1 时刻 $x_2x_1=10$,有脉冲出现但没有启动信号,输出Z为0,设用状态②表示; t_2 时刻与 t_0 时刻相同; t_3 时刻 $x_2x_1=01$,有启动信号但无脉冲信号,输出Z为0,设用状态③表示; t_4 时刻 $x_2x_1=11$,启动信号和脉冲信号同时出现,输出Z为0,设用状态④表示; t_5 时刻与 t_3 时刻相同; t_6 时刻与 t_4 时刻相同; t_7 时刻 $x_2x_1=10$,此时 x_2 端有启动信号结束后的不完整脉冲,输出Z为0,设用状态⑤表示; t_8 时刻 $x_2x_1=00$,此时启动信号已结束,但第一个完整脉冲尚未出现,输出Z为0,设用状态⑥表示; t_9 时刻,出现了启动信号结束后的第一个完整脉冲,输出Z为1,产生一个完整输出脉冲,设用状态⑦表示。此后,电路将重复此过程,实现每启动一次,输出一个完整脉冲的功能。

根据时间图中设立的状态可建立原始流程表,如表6.31所示。

表 6.31 原始流程表

二次状态 y	激励状态 Y/输出 Z			
	$x_2x_1=00$	$x_2x_1=01$	$x_2x_1=11$	$x_2x_1=10$
1	①/0	3/0	d/d	2/0
2	1/0	d/d	4/0	②/0
3	6/0	③/0	4/0	d/d
4	d/d	3/0	④/0	5/0
5	6/0	d/d	d/d	⑤/0
6	⑥/0	d/d	d/d	7/d
7	1/d	d/d	d/d	⑦/1

由于问题中规定每启动一次,必须输出一个完整脉冲后才能再次启动,所以,处在稳态⑤时输入取值不允许从10→11,处在稳态⑥时输入取值不允许从00→01,处在稳态⑦时输入取值不允许从10→11。因此,在流程表上的相应位置填入任意状态和任意输出“d”。

② 化简流程表。根据相容行的判断法则,可作出与表6.31对应的隐含表,如图6.32(a)所示。由隐含表可得到相容行对(1,2)、(3,4)、(3,5)、(3,6)、(4,5)。据此,可作出状态合并图,如图6.32(b)所示,其最大相容行类为(1,2)、(3,4,5)、(3,6)、(7)。

根据选择最小闭覆盖的条件,可选择相容行类集合{(1,2),(3,4,5),(6),(7)}。令(1,2)

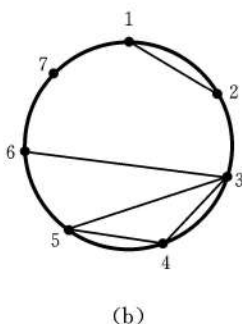
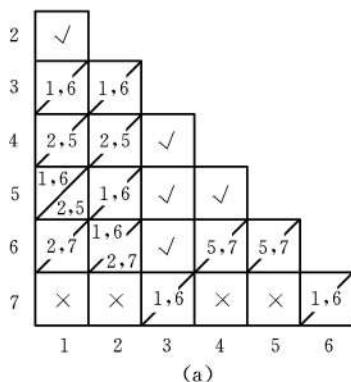


图 6.32 隐含表和状态合并图

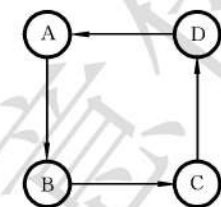


图 6.33 状态相邻图

用 A 表示, (3,4,5) 用 B 表示, (6) 用 C 表示, (7) 用 D 表示, 合并后的最简流程表如表 6.32 所示。

③ 状态编码。根据表 6.32 可作出状态相邻图, 如图 6.33 所示。设二次状态用 y_2 、 y_1 表示, 为了满足图 6.33 所示的相邻关系, 可令 y_2y_1 取值 00 表示 A, 01 表示 B, 11 表示 C, 10 表示 D。将各状态的编码代入表 6.32, 即可得到表 6.33 所示二进制流程表。

表 6.32 最简流程表

二次状态 y	激励状态 Y/输出 Z			
	$x_2x_1=00$	$x_2x_1=01$	$x_2x_1=11$	$x_2x_1=10$
A	Ⓐ/0	B/0	B/0	Ⓐ/0
B	C/0	Ⓑ/0	Ⓑ/0	Ⓑ/0
C	Ⓒ/0	d/d	d/d	D/d
D	A/d	d/d	d/d	Ⓓ/1

表 6.33 二进制流程表

二次状态 $y_2 y_1$		激励状态 $Y_2 Y_1$ /输出 Z			
y_2	y_1	$x_2x_1=00$	$x_2x_1=01$	$x_2x_1=11$	$x_2x_1=10$
0	0	Ⓐ/0	01/0	01/0	Ⓐ/0
0	1	11/0	Ⓑ/0	Ⓑ/0	Ⓑ/0
1	1	Ⓒ/0	d/d	d/d	10/d
1	0	00/d	d/d	d/d	Ⓓ/1

④ 确定激励状态和输出函数表达式。根据二进制流程表可作出激励状态和输出函数卡诺图, 如图 6.34 所示。

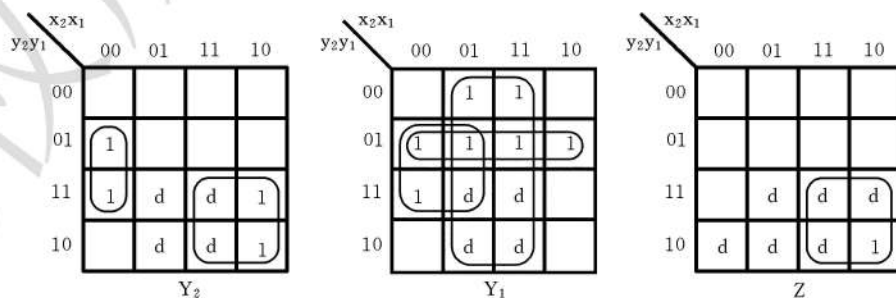


图 6.34 激励函数和输出函数卡诺图

用卡诺图化简后,可得到激励状态和输出函数的最简表达式:

$$Y_2 = x_2 y_2 + \bar{x}_2 \bar{x}_1 y_1 = \overline{x_2 y_2 \cdot x_2 x_1 y_1}$$

$$Y_1 = x_1 + \bar{x}_2 y_1 + \bar{y}_2 y_1 = \overline{x_1 \cdot x_2 y_2 \cdot y_1}$$

$$Z = x_2 y_2 = \overline{x_2 y_2}$$

⑤ 画出逻辑电路图。根据激励状态和输出函数的与-非-与非表达式,可画出用与非门实现给定功能的逻辑电路,如图 6.35 所示。

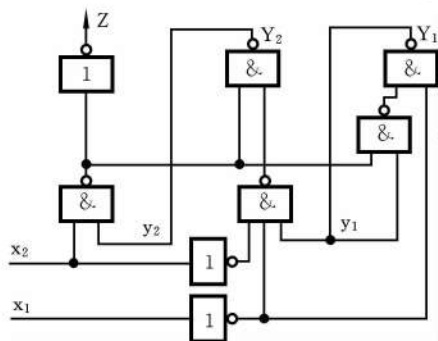


图 6.35 逻辑电路

习 题 六

6.1 分析图 6.36 所示脉冲异步时序逻辑电路。

- (1) 作出状态表和状态图;
- (2) 说明电路逻辑功能。

6.2 分析图 6.37 所示脉冲异步时序逻辑电路。

- (1) 作出状态表和时间图;
- (2) 说明电路逻辑功能。

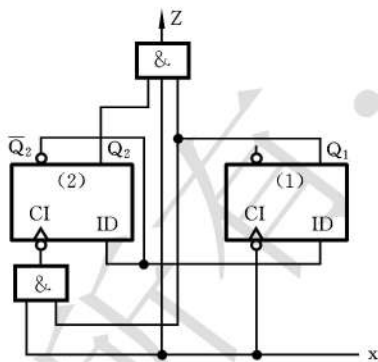


图 6.36 逻辑电路

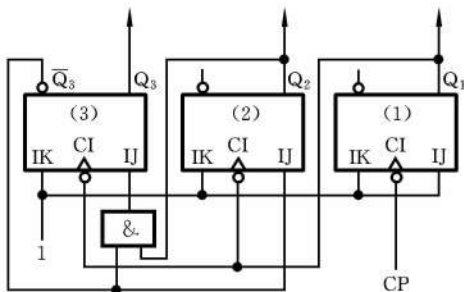


图 6.37 逻辑电路

6.3 分析图 6.38 所示脉冲异步时序逻辑电路。

- (1) 作出状态表和状态图;
- (2) 说明电路逻辑功能。

6.4 分析图 6.39 所示脉冲异步时序逻辑电路,作出时间图并说明该电路逻辑功能。

6.5 用 D 触发器作为存储元件,设计一个脉冲异步时序逻辑电路。该电路在输入端 x 的脉冲作用下,实现 3 位二进制减 1 计数的功能,当电路状态为“000”时,在输入脉冲作用下输出端 Z 产生一个借位脉冲,平时 Z 输出 0。

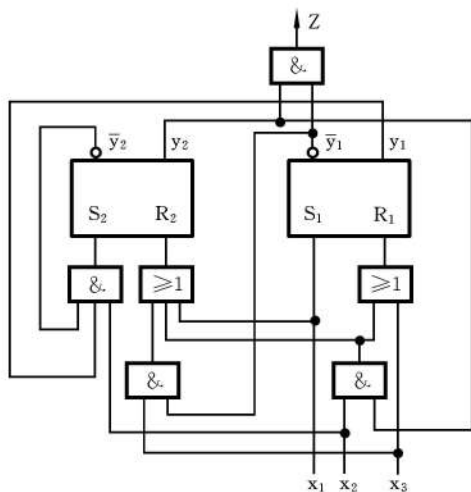


图 6.38 逻辑电路

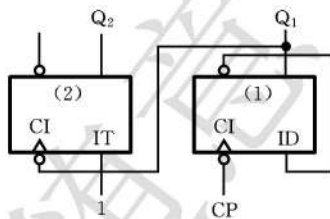


图 6.39 逻辑电路

6.6 用 T 触发器作为存储元件,设计一个脉冲异步时序逻辑电路,该电路有两个输入 x_1 和 x_2 , 一个输出 Z , 当输入序列为“ $x_1-x_1-x_2$ ”时,在输出端 Z 产生一个脉冲,平时 Z 输出为 0。

6.7 试用与非门构成的基本 R-S 触发器设计一个模 4 加 1 计数器。

6.8 分析图 6.40 所示电平异步时序逻辑电路,作出流程表。

6.9 分析图 6.41 所示电平异步时序逻辑电路,作出流程表和总态图,说明该电路的逻辑功能。

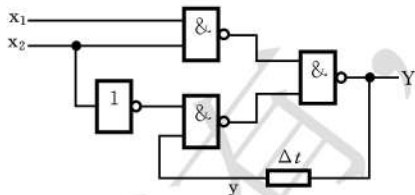


图 6.40 逻辑电路

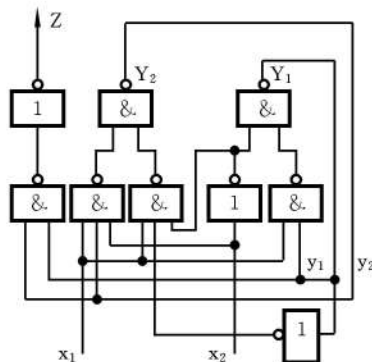


图 6.41 逻辑电路

6.10 某电平异步时序逻辑电路的流程表如表 6.34 所示。作出输入 x_2x_1 变化序列为 $00 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00$ 时的总态(x_2x_1, y_2y_1)响应序列。

表 6.34 流程表

二次状态 $y_2 \ y_1$	激励状态 Y_2Y_1 /输出 Z			
	$x_2x_1=00$	$x_2x_1=01$	$x_2x_1=11$	$x_2x_1=10$
0 0	00/0	01/0	01/0	10/0
0 1	00/0	01/0	01/0	11/0
1 1	00/0	01/0	10/0	11/0
1 0	00/d	00/1	10/1	10/1

6.11 某电平异步时序逻辑电路有一个输入 x 和一个输出 Z , 每当输入 x 出现一次 $0 \rightarrow 1$ $\rightarrow 0$ 的跳变后, 当 x 为 1 时输出 Z 为 1, 典型输入/输出时间图如图 6.42 所示。试建立该电路的原始流程表。

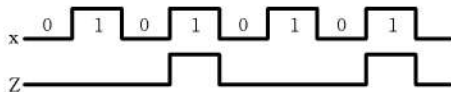


图 6.42 时间图

6.12 简化表 6.35 所示的原始流程表。

表 6.35 原始流程表

二次状态 y	激励状态/输出 (Y/Z)			
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$
1	①/0	5/d	d/d	2/d
2	1/d	d/d	3/d	②/0
3	d/d	5/d	③/1	4/d
4	1/d	d/d	3/d	④/1
5	1/d	⑤/0	6/d	d/d
6	d/d	5/d	⑥/0	4/d

6.13 图 6.43 为某电平异步时序逻辑电路的结构框图。图中,

$$Y_2 = x_2 y_2 + \bar{x}_1 y_2 + x_2 \bar{x}_1 y_1$$

$$Y_1 = x_2 x_1 + \bar{x}_2 \bar{x}_1 y_2 + x_1 y_2 \bar{y}_1$$

$$Z = y_2 y_1$$

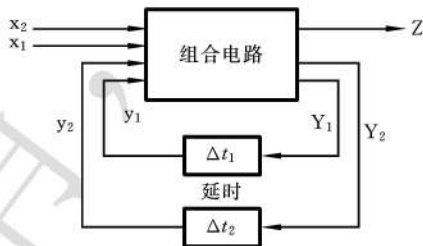


图 6.43 结构框图

试问该电路中是否存在竞争? 若存在, 请说明竞争类型?

6.14 对表 6.36 所示的最简流程表进行无临界竞争的状态编码, 并确定激励状态和输出函数表达式。

表 6.36 最简流程表

二次状态 y	激励状态 Y /输出 Z			
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 11$	$x_2 x_1 = 10$
A	①/0	①/0	③/0	C/d
B	②/0	A/0	C/d	④/0
C	B/d	A/d	③/1	③/1

6.15 某电平异步时序逻辑电路有两个输入 x_1 、 x_2 和一个输出 Z 。当 $x_2 = 1$ 时, Z 总为 0; 当 $x_2 = 0$ 时, x_1 第一次从 $0 \rightarrow 1$ 的跳变使 Z 变为 1, 该 1 输出信号一直保持到 x_2 由 $0 \rightarrow 1$, 才使 Z 为 0。试用与非门实现该电路功能。