(3) 具有与门输入的 8 位移位寄存器 74LS164

除了 4 位移位寄存器，现在也有许多 8 位移位寄存器芯片。74LS164 就是一种 8 位移位寄存器。

74LS164 的功能比较简单：只能单向移位。另外，也有异步清零的输入端。它也是由 D 触发器构成的移位寄存器。

但是 74LS164 的输入有些特点：它有两个移位输入 A 和 B，两个输入加到一个内部的与门，与门的输出才是移位寄存器的输入。

这样的结构对于构成反馈移位寄存器的应用也有用。

74LS164 的功能表如表 7－22 所示。

表 7－22　74LS164 功能表

$\overline{CLR}$	CLK	A	B	Q_A	Q_B	Q_C	Q_D	Q_E	Q_F	Q_G	Q_H
0	ϕ	ϕ	ϕ	0	0	0	0	0	0	0	0
1	↑	1	1	1	Q_A	Q_B	Q_C	Q_D	Q_E	Q_F	Q_G
1	↑	ϕ	0	0	Q_A	Q_B	Q_C	Q_D	Q_E	Q_F	Q_G
1	↑	0	ϕ	0	Q_A	Q_B	Q_C	Q_D	Q_E	Q_F	Q_G

(4) 具有预置功能的 8 位移位寄存器 74LS166

74LS166 是具有并行预置功能的移位寄存器。它的控制方式和 74LS195 很相似，也使用 SH/$\overline{LD}$端来控制移位寄存器是进行移位还是预置操作。它也有一个异步的清零端$\overline{CLR}$。这些特性都和 74LS195 相似。不同的地方有三：首先是位数的不同，一个是 4 位，一个是 8 位。第二是输入方式不同，74LS166 是由 D 触发器构成，串行输入只有一个 SER，而 74LS195 是 JK 输入；最主要的不同是 74LS166 有两个和时钟有关的输入端：CLK 和 CLKINH。其中的 CLK 就是时钟输入，CLKINH 则是“时钟禁止”输入，当 CLKINH＝0 时，允许时钟输入，而 CLKINH＝1 时，不允许时钟输入。

74LS166 的功能表和 74LS195 很相似，读者可以自己画出。

图 7－43 是移位寄存器 74LS194、74LS195、74LS164 和 74LS166 的逻辑符号。

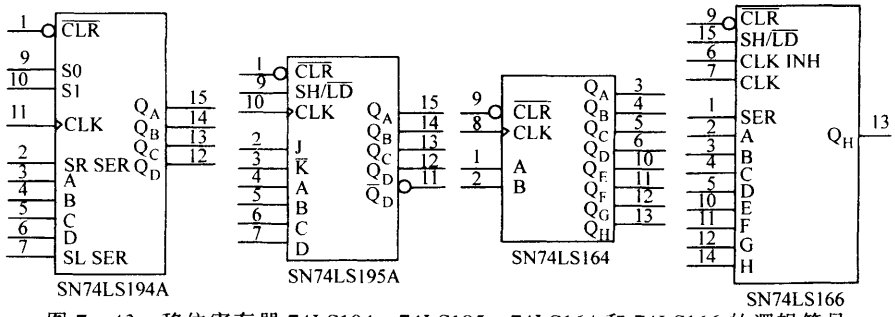

图 7－43　移位寄存器 74LS194、74LS195、74LS164 和 74LS166 的逻辑符号

7.5.4　中规模移位寄存器的应用

移位寄存器在通信系统和计算机系统中都有广泛的应用。其中最主要的应用就是串行数据和并行数据的互相转换。另外，移位寄存器也可以用作序列信号发生器和计数器。

1. 串－并、并－串变换

移位寄存器本身就有串入并出的功能。完成数据的串行－并行转换的关键在于控制信号的产生。例如，数据的长度是 8 位，控制信号应该保证 8 位数据串行移入移位寄存器后，产生一个并行输出控制信号，将已经移入移位寄存器的 8 位数据，并行输出到一个外部的锁存器。然后，控制信号再恢复为移位控制，继续进行串入－并出的变换。

图 7－44 是实现 8 位串行数据转换为并行数据的一种控制方式。

移位寄存器只需要有串入并出的功能，使用 74LS164 就可以满足需要。

用 8 位寄存器 74LS374 来存放并行的输出数据。74LS374 是上升沿写入的数据寄存器，并且必须在使能控制$\overline{OC}=0$时才能写入。在 8 位数据串行移入寄存器后，应该为 74LS374 提供一个写入数据的控制信号，连接到$\overline{OC}$。

计数器 74LS169 用来产生控制串行移位和并行输出的控制信号。计数器的低 3 位输出经过一个或门，产生寄存器所需要的写入数据控制信号。当计数器进入状态 1000（或者说，低 3 位是 000），或门输出 0，就可以连接到寄存器的$\overline{OC}$控制端。使得寄存器处于可以写入数据的状态，在时钟有效时，将并行数据写入寄存器。

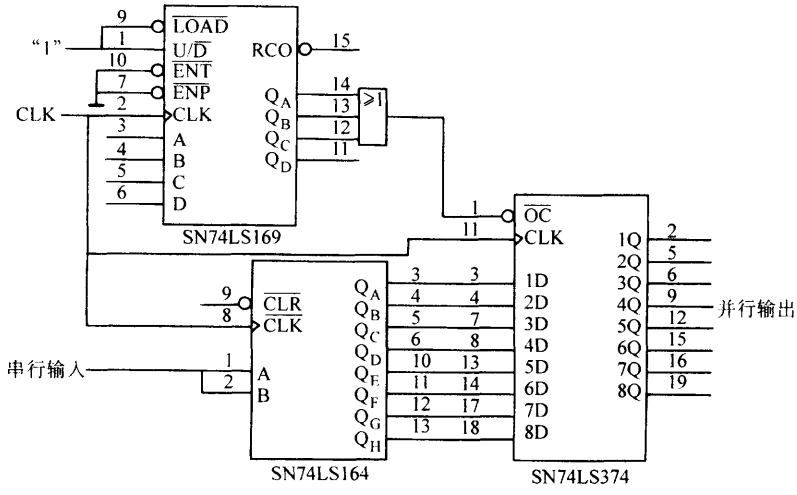

图 7－44　串行－并行数据转换

并行－串行数据变换的一种控制方式示于图 7－45。

这时使用的移位寄存器应该有并入串出的功能。选用具有预置功能的 8 位移位寄存器 74LS166。因为是用于并入串出，所以，串行输入 SER 和时钟禁止端 CLKINH 都接地电位。

另外，用计数器 74LS169 产生移位/预置的控制信号。计数器低 3 位的输出接到一个与非门，当计数器处于 0111 状态时，与非门的输出等于 0，就可以用来控制移位寄存器的 SH/$\overline{LD}$端，实现并行输入，当计数器不是处于 0111 状态时，与非门输出为 1，使得移位寄存器进行移位操作。所以，每当 8 位数据中的最后一位移位到最后一个触发器时，下一个时钟到来时，就实现并行输入，然后，再进行串行移位和串行输出。从而实现了将并行数据转换为串行数据。

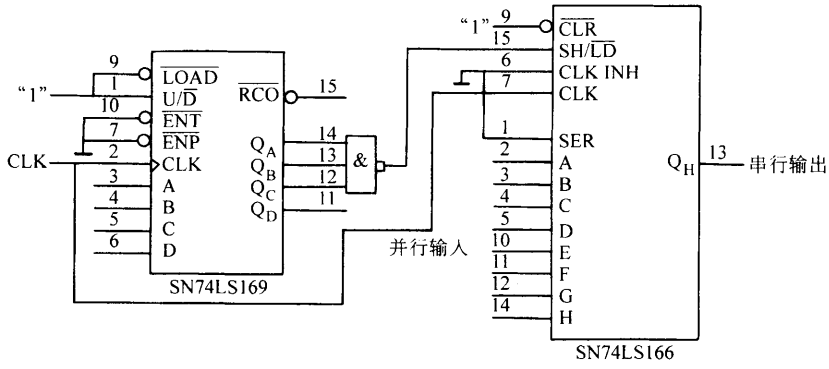

图 7－45　并行－串行数据转换

2. 序列信号发生器

中规模移位寄存器经常和数据选择器一起来构成序列信号发生器。因为构成序列信号发生器时，需要将各级触发器的输出连接到反馈电路，以便对第一级触发器产生反馈信号。有时会要求从触发器的$\overline{Q}$端取得信号。但是，中规模移位寄存器一般都不提供$\overline{Q}$端的输出信号。这样，就可能使用很多反相器，不是最好的方案。如果使用数据选择器作为反馈电路，这个问题就可以基本上解决，因为数据选择器的地址输入不需要反相信号。当然，数据选择器的数据输入部分，还可能需要一些反相器，但是，数量就会少很多。

其余的设计步骤和以前介绍过的相同。

【例 7－13】 用中规模移位寄存器和数据选择器设计一个序列信号发生器，输出序列：

0110011110001001…

解：1）确定移位寄存器的位数。序列的长度是 16，寄存器最小位数是 4。

检查位数 4 是否足够。将序列每 4 位作为一组，每组移一位，共写 16 组：

0110、1100、1001、0011、0111、1111、1110、1100

1000、0001、0010、0100、1001、0010、0101、1011

16 组代码中有好几种代码出现了两次，所以移位寄存器的位数不能是 4。再用位数等于 5 来试验，新写出的 16 组 5 位代码没有重复，所以 5 位已经足够。

选用 5 位移位寄存器 74LS96。它是可以并行预置的移位寄存器，通过并行输入允许信号 PE 来控制：PE＝0 时为串行移位，PE＝1 时为并行预值。在这个例题中，不需要并行预置，PE 端可以固定接地。

2）写出状态转移表，如表 7－23 所示。DATA 是 74LS96 的串行数据输入端，也就是接受反馈输出的端子。

表 7－23　例 7－13 的状态转移表

Q_4	Q_3	Q_2	Q_1	Q_0	DATA
0	1	1	0	0	1
1	1	0	0	1	1
1	0	0	1	1	1
0	0	1	1	1	1

（续）

Q_4	Q_3	Q_2	Q_1	Q_0	DATA
0	1	1	1	1	0
1	1	1	1	0	0
1	1	1	0	0	0
1	1	0	0	0	1
1	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	1	0	1
0	0	1	0	1	1
0	1	0	1	1	0
1	0	1	1	0	0

3）画出 DATA 的卡诺图，如图 7－46 所示。由于是用数据选择器来实现组合逻辑电路，首先要选择数据选择器的地址输入。现在共有 5 个触发器的输出可供选择。可以选择 $Q_4Q_3Q_2$ 作为地址输入。

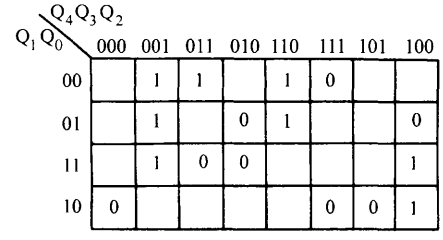

图 7－46　例 7－13 的卡诺图

在卡诺图上，按照 $Q_4Q_3Q_2$ 的 8 种组合，分为 8 个子图，然后可以得到数据选择器的 8 个数据输入为：

$$I_0=0 \quad I_1=1 \quad I_2=0 \quad I_3=\overline{Q_1}$$

$$I_4=Q_1 \quad I_5=0 \quad I_6=1 \quad I_7=0$$

但是，如果这样选择数据，当进入状态 00000 时，下一状态将仍然是 00000，不能自启动。所以，应该将卡诺图中 00000 格的任意项的值选为 1，I_0 的数据输入修改为：

$$I_0=\overline{Q_1}$$

4）画逻辑图，如图 7－47 所示。

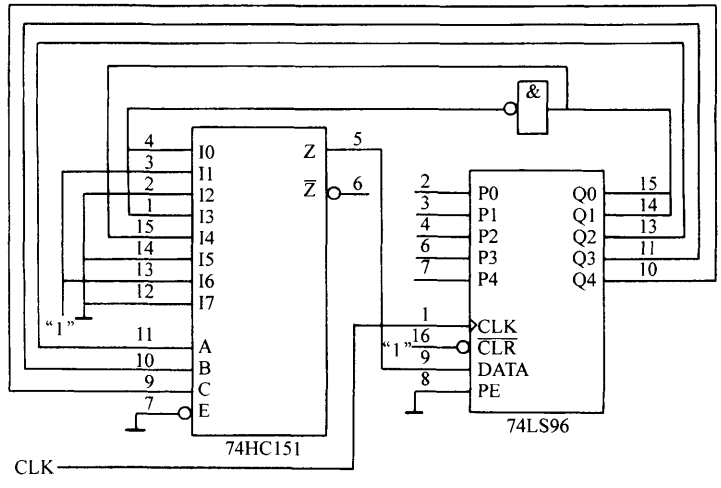

图 7－47　例 7－13 的逻辑图

3. 移存型计数器

用中规模移位寄存器也可以构成计数器。特别是可以很方便地构成环形计数器和扭环计数器。有些移位寄存器的输出端还包括最高一级触发器$\overline{Q}$的输出，这样的移位寄存器就可以直接用来连接成扭环计数器。

为了构成任意进制的计数器，可以采用以下的方法：

（1）先将移位寄存器连接为 M 序列发生器。因为 k 位移位寄存器构成的 M 序列的长度是 2^k-1，接近 k 位移位寄存器可以产生的最大状态数 2^k。

（2）作出 M 序列发生器的状态转移图。用移位寄存器构成的计数器的状态转移关系是不能任意选择的，必须满足移位寄存器的状态转移关系。也就是只能在这个 M 序列状态转移图中选择。

（3）按照需要的模值决定计数器计数的初始状态和终止状态。因为移存型计数器没有进位，终止状态要自己选择。选定了终止状态，就可以确定初始状态，进行预置值的确定。

【例 7-14】 用移位寄存器设计一个模值等于 10 的计数器。

解： 1）模值等于 10，选择 4 位移位寄存器就可以。为了便于构成 M 序列发生器，可以选择 JK 输入的 74LS195。

4 位移位寄存器构成的 M 序列的反馈函数是：

$$Q_A^n \odot Q_D^n = \overline{Q_D^n}\ \overline{Q_A^n} + Q_A^n Q_D^n$$

所以，只要将 Q_D 连接到 J，$\overline{Q_D}$连接到$\overline{K}$，就实现了 M 序列发生器。

不难写出 M 序列为 000010100110111，长度等于 15。也就是，构成计数器的最大模值是 15。

2）作状态转移图，如图 7-48 所示。

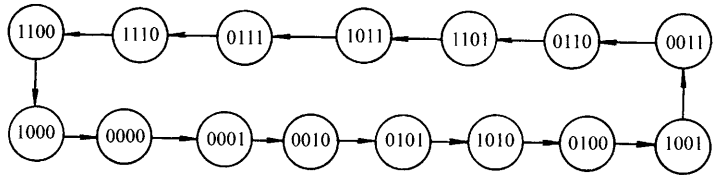

图 7-48　例 7-14 的状态转移图

3）确定终止状态和初始状态。原则上说，对于移存型计数器来说，终止状态可以任意选择。但是要考虑是否便于检测，检测时尽量少用逻辑门，并且不容易和其他状态相混。例如，可以选择 0111、1110、1101 或 1011 状态中的一个作为终止状态。只要外加一个与非门就可以产生预置控制信号。

现在选用 0111 状态为终止状态。用一个与非门就可以检测这个状态。

终止状态确定后，在状态转移图上，从终止状态开始，倒数 M 个状态就是初始状态。现在要求模值是 10，从 0111 状态倒数 10 个状态就是 0010，0010 就是计数器的初始状态。计数循环是：

$$0010\rightarrow 0101\rightarrow 1010\rightarrow 0100\rightarrow 1001\rightarrow 0011\rightarrow 0110\rightarrow 1101\rightarrow 1011\rightarrow 0111$$

4）作逻辑图，如图 7－49 所示。

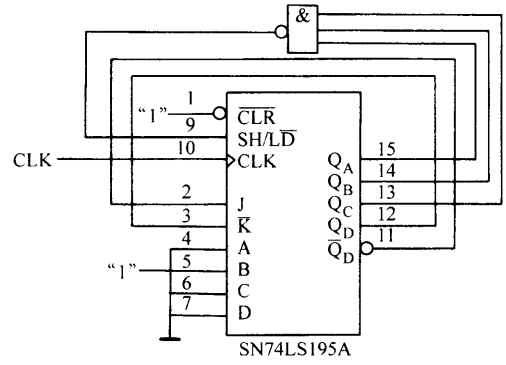

图 7－49　例 7－14 的逻辑图

7.5.5　时序部件的 VHDL 描述

用 VHDL 描述时序部件，特别是描述计数器时，还要用到两个标准的程序包：

IEEE.STD_LOGIC_ARITH

IEEE.STD_LOGIC_UNSIGNED

主要使用的是前一个程序包。在 IEEE.STD_LOGIC_ARITH 程序包中定义了两种新的数据类型：

TYPE SIGENED IS ARRAY (NATURAL RANGE< >) OF STD_LOGIC;

TYPE UNSIGENED IS ARRAY (NATURAL RANGE< >) OF STD_LOGIC;

从数据类型的定义来看，和 STD_LOGIC 类型没有什么差别。但是在这个程序包中，定义了许多新的函数，可以允许 SIGENED 或者 UNSIGENED 类型的数据进行算术运算，例如加法运算，减法运算等。还允许这些类型的数据和整型数据一起进行运算。或者在整型数据和逻辑类型数据之间进行转换。

IEEE.STD_LOGIC_UNSIGNED 程序包进一步补充了一些函数，这些函数主要是 STD_LOGIC 类型之间的算术运算，以及 STD_LOGIC 类型和整数之间的算术运算。

VHDL 环境可以根据函数参数的类型，选择相应的函数进行运算。

在使用这些程序包和数据类型时，要特别注意表达式中数据类型的一致。

1. 计数器的 VHDL 描述

【例 7－15】 分析一个简单计数器的描述，说明这个计数器的功能。

解：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY counter IS
PORT( clock: IN STD_LOGIC;
            clear: IN STD_LOGIC;
            count: IN STD_LOGIC;
            q: OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END counter;
ARCHITECTURE behav OF counter IS
   SIGNAL pre_q: STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN
    PROCESS(clock, count, clear)
    BEGIN
    IF clear = '1' THEN
```