线性和非线性寄存器系统的并行化技术

秦晓懿,王瀚晟,曾烈光
(清华大学电子工程系微波与数字通信技术国家重点实验室,北京 100084)

摘 要: 并行化技术可降低电路工作速率、延时和功耗,广泛应用于通信处理中. 对线性寄存器系统,通过对系统状态方程和输出方程的讨论提出一般性的(1, N)并行化方法,其对任意并行路数 N 均有统一计算方法;并对某些情况下的(M, N)并行提出一种新实现方法. 对非线性寄存器系统,给出其定义,对其状态转移进行线性化,提出线性化矩阵法的并行方法;并对其特例——非线性移位寄存器的并行化提出推广延时因子法.

关键词: 线性寄存器系统; 非线性寄存器系统; 非线性移位寄存器; (M, N) 并行; 线性化矩阵法; 推广延时因子法

中图分类号: TP332.5; TP332.3 **文献标识码**: A **文章编号**: 0372-2112 (2003) 03-0406-05

Paralleling Techniques for Linear and Non-linear Register Systems

QIN Xiao-yi, WANG Han-sheng, ZENG Lie-guang
(State Key Laboratory on Microwave & Digital Communication, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

Abstract: Paralleling techniques play an important role to decrease circuits' frequency and to satisfy requirements of delay and power cost, and have wide applications in communication signal processing. For LRS, a universal (1, N) paralleling method is presented by discussions on the system state equation and the output equation. And the computing arithmetic is identical for arbitrary parallel number. And a novel method for (M, N) paralleling realization is also presented in some conditions. For NLRS, the definition is given at first, and the paralleling method of linearization matrix method is presented by linearizing states' transfer. For the non-linear shift register, a novel paralleling method called generalizing delay operator method is presented.

Key words: LRS/NLRS (linear/non-linear register systems); non-linear shift register; (M, N) paralleling method; linearization matrix method; generalizing delay operator method

1 引言

通信速率向更高速率飞速发展,集成电路工艺已无法跟上光通信带宽的发展,因此研究一些电路的并行实现,以降低电路的工作速率、延时和功耗也具有积极意义. 图 1 为反馈移位寄存器的框图,根据反馈函数是线性或非线性可以分为线性移位寄存器和非线性移位寄存器两大类[1]. 目前广泛应用于通信处理中的线性移位寄存器及其并行化技术的理论研究已非常成熟,针对不同应用的并行化方法也相继提出,如文[2]研究矩阵法对各种扰码的并行化;文[3]用矩阵法和延时因子法对帧同步扰码实现并行;文[4]用 Z 变换法实现并行循环编码;文[5,6]从序列空间和序列采样的角度对无输入的线性移位寄存器及其并行化作了详细研究. 但对更一般的线性寄存器系统的并行化问题则几乎没有讨论过. 而对非线性移位寄存器,虽然其理论还很不成熟,但由于非线性的反馈函数更多,可从中寻找更多更好的伪随机序列,因此也有很好的应用前景. 本文则对非线性移位寄存器及更一般的线性和非线性寄存器系统的并行化技术进行研究.

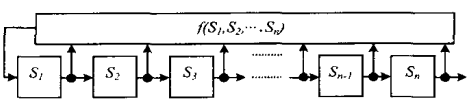

图 1 反馈移位寄存器

2 线性和非线性寄存器系统定义

线性寄存器系统 LRS(Linear Register System)是仅由寄存器和异或门构成的系统,其每个寄存器的下一状态和电路的输出均为输入和各寄存器当前状态的线性组合. 而非线性寄存器系统 NLRS(Non-Linear Register System)则指系统中寄存器的下一状态和当前输出为当前寄存器状态和当前输入的非线性组合. 因此,有 r 个输入端、m 个输出端和 n 个寄存器的(r, m, n)寄存器系统,每个时刻的输入、输出和寄存器状态分别用域 $GF(2)$ 上的向量 $x = [x_1, x_2, \cdots, x_r]^T$, $y = [y_1, y_2, \cdots, y_m]^T$ 和 $s = [s_1, s_2, \cdots, s_n]^T$ 表示,对 LRS 而言则有 $s_i(k+1) =$

收稿日期:2001-06-18;修回日期:2002-11-03
基金项目:国家"九五"科技攻关项目(No. 0239623)

$\sum_{j=1}^{n} a_{ij}s_j(k) + \sum_{j=1}^{r} b_{ij}x_j(k)$, $y_i(k) = \sum_{j=1}^{n} c_{ij}s_i(k) + \sum_{j=1}^{r} d_{ij}x_j(k)$, 写成向量矩阵方程的形式为:

$$s(k+1) = As(k) + Bx(k) \quad (1)$$

$$y(k) = Cs(k) + Dx(k) \quad (2)$$

其中 A 是 $n \times n$ 状态转移矩阵; B 是 $n \times r$ 矩阵; C 是 $m \times n$ 矩阵; D 是 $m \times r$ 矩阵; $k = 0,1,2,\cdots$. 式(1)和式(2)分别决定状态序列生成和输出序列生成. 对 NLRS 而言则有

$$s_i(k+1) = f_i(s_1(k), s_2(k), \cdots, s_n(k); x_1(k), x_2(k), \cdots, x_r(k)), i = 1, \cdots, n \quad (3)$$

$$y_j(k) = g_j(s_1(k), s_2(k), \cdots, s_n(k); x_1(k), x_2(k), \cdots, x_r(k)), j = 1, \cdots, m \quad (4)$$

其中 f_i、g_j ($i = 1, \cdots, n; j = 1, \cdots, m$) 中至少有一个为非线性.

3 线性寄存器系统的并行化

设线性寄存器系统 M_S 的第 i 个输出序列为 $\{y_{i,k}\}$, $i = 1, 2, \cdots m$, 若线性寄存器系统 M_P 同时产生 N 个并行序列 $Y_{i,j}$, $j = 1,2, \cdots, N$, 且满足

$$Y_{i,1} = \{y_{i,1}, y_{i,2}, \cdots, y_{i,M}; y_{i,MN+1}, y_{i,MN+2}, \cdots, y_{i,M(N+1)}; \cdots\cdots\}$$

$$Y_{i,2} = \{y_{i,M+1}, y_{i,M+2}, \cdots, y_{i,2M}; y_{M(N+1)+1}, y_{i,M(N+1)+2}, \cdots y_{i,M(N+2)}; \cdots\cdots\}$$

$$N$$

$$Y_{i,N} = \{y_{i,M(N-1)+1}, y_{i,M(N-1)+2}, \cdots, y_{i,MN}; y_{i,M(2N-1)+1}, y_{i,M(2N-1)+2}, \cdots, y_{i,2MN}; \cdots\cdots\}$$

则称线性寄存器系统 M_P 为序列 $\{y_{i,k}\}$ 的 (M, N) 并行线性寄存器系统 PLRS(Parallel Linear Register System). 对给定信息流, 经 (M, N) PLRS 后, 再并/串变换, 即将此 N 个并行序列 $Y_{i,j}$ ($j = 1,2, \cdots N$) 进行 M-bit 间插, 得到的串行码流与原串行码流 $\{y_{i,k}\}$ 完全一致. 同样, PLRS 也包含两部分: 并行状态序列生成和并行输出序列生成. 由式(2)知, 只要与输出有关的状态序列的生成并行化后, 输出序列的并行化则显而易见, 故 LRS 的并行化也就是状态序列生成的并行化问题.

首先讨论 $M = 1$ 的情况, 即并行序列通过比特间插即可得串行序列. 然后在此基础上研究 $M > 1$ 的情况.

3.1 状态序列的并行化

以 $s(k)$ 为初始状态, 由式(1)、(2)推出(定义 $A^0 = I$)

$$s(k+L) = A^L s(k) + \sum_{j=0}^{L-1} A^{L-j-1} Bx(k+j), \quad k = 0,1,2, \cdots; L = 1,2, \cdots \quad (5)$$

因此, 对 (r, m, n) LRS 的状态向量 $s = [s_1, s_2, \cdots, s_n]^T$ 的每个分量构成的状态序列的生成可由以下推导对其并行化. 由式(5)可知:

$$s_i(k+L) = (A^L)_i s(k) + \sum_{j=0}^{L-1} (A^{L-j-1}B)_i x(k+j), \quad k = 0,1,2, \cdots; L = 1,2, \cdots \quad (6)$$

其中符号 $(\cdot)_i$ 表示矩阵 $(\cdot)$ 的第 i 行, 即 $(A^L)_i$ 表示矩阵 A^L 的第 i 行, $(A^{L-j-1}B)_i$ 表示矩阵 $A^{L-j-1}B$ 的第 i 行. 可见, 通过 $s(k)$ 和并行输入 $x(k+j)$ ($j = 0,1, \cdots, N-1$) 即可同时产生 $s_i(k+j)$ ($j = 0,1, \cdots, N-1$). 因此按式(5)和式(6), LRS 的第 i 个状态序列 $\{s_{i,k}\}$ 的并行化生成可由如下实现:

(a) 由 $s(k)$ 和并行输入 $x(k+j)$ ($j = 0,1, \cdots, N-1$) 同时产生 $s(k+N)$ 和 $\{s_{i,k}\}$ 的 N 个并行序列值 $s_i(k+j)$ ($j = 0,1, \cdots, N-1$);

(b) 在下一并行时钟周期开始时将状态值 $s(k+N)$ 送入 $s(k)$, 即 $s(k) \leftarrow s(k+N)$, $k \leftarrow k+N$, 再跳转到(a);

以此循环类推, 即可得 $\{s_{i,k}\}$ 的 N 个并行序列. 步骤(a)可由异或门实现, 步骤(b)由寄存器实现. 对任意 N, 状态序列 $\{s_{i,k}\}$ 的 PLRS 所需寄存器个数均为 n, 与原 LRS 相同.

3.2 线性寄存器系统的并行化

用上述方法对 LRS 的 n 个状态分量 $s_1 \sim s_n$ 分别并行化, 由于 $s(k+N)$ 的产生对每个状态分量都相同, 故步骤(b)所需寄存器及产生 $s(k+N)$ 所需异或门对所有状态分量来说是公共的. 按照式(2)对并行化后的状态和输入向量进行线性组合(由矩阵 C, D 决定)即可得并行化后的输出向量. 故对应于 (r, m, n) LRS 的 $(1, N)$ PLRS 所需寄存器个数仍为 n; 只是异或门个数大大增加. 图 2 是 (r, m, n) LRS 并行化实现的电路结构, 表示电路实现的最大集, 实际应用中并不一定需要图中所有模块. 例如, 当并行输出序列的生成只与状态分量 s_i 和 s_j 有关时, 则 $L_1 \sim L_n$ 中只需 L_i 和 L_j 模块. 并且, L 和 $L_1 \sim L_n$ 模块间可能也有相同部分可以共用以节省电路.

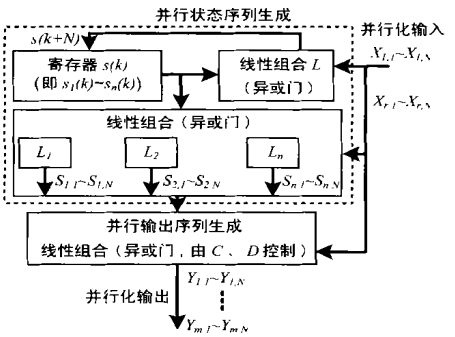

图 2　(r, m, n) LRS 的并行化实现

例: 对帧同步扰码, 对应于图 1 有 $f(s_1, s_2, \cdots, s_n) = \sum_{r=1}^{n} a_i s_i$, 即生成多项式为 $g_I(x) = \sum_{i=0}^{n} a_i x^i$, $a_0 = a_n = 1$; 且 $y(k) = s_n(k) + x(k)$, 按上述方法对其并行化. 由于 $B = 0_{n \times 1}$, 故 $s(k+N) = A^N s(k)$, 且有

$$\begin{bmatrix} S_1 \\ S_2 \\ M \\ S_N \end{bmatrix} = \begin{bmatrix} s_n(k) \\ s_n(k+1) \\ M \\ s_n(k+N-1) \end{bmatrix} = \begin{bmatrix} e_n \\ (A)_n \\ M \\ (A^N)_n \end{bmatrix} s(k) = \begin{cases} \begin{bmatrix} e_n \\ e_{n-1} \\ M \\ e_{n-N+1} \end{bmatrix} s(k), & N \le n \\ \begin{bmatrix} e_n \\ M \\ e_1 \\ \hline (A)_1 \\ M \\ (A^{N-n})_1 \end{bmatrix} s(k), & N > n \end{cases}$$

式(7)中 A 为

$$A_I=\begin{bmatrix} a_1 & a_2 & \cdots & a_{n-1} & a_n \\ 1 & 0 & \cdots & \cdots & 0 \\ 0 & O & O & & M \\ M & O & O & O & M \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \tag{7}$$

因此 $s_n(k)$ 的并行化过程为:在每个并行周期,对 $N \leq n$, $s(k)$ 的后 N 个分量 $s_n(k) \sim s_{n-N+1}(k)$ 直接作为 N 个并行序列 $S_1 \sim S_N$;对 $N \geq n$, $s(k)$ 的 n 个分量作为 $S_1 \sim S_n$, $S_{n+1} \sim S_N$ 则由 $s(k)$ 的分量的线性组合生成. 其中序列 S_1 对应于 MSB,即在串行序列中最先输出, S_N 对应于 LSB. 在下一并行周期的开始,用 A^N 和 $s(k)$ 生成的 $s(k+N)$ 更新寄存器状态 $s(k)$. 如此循环,即可得 $s_n(k)$ 的并行序列, $y(k)$ 的并行序列为 $s_n(k)$ 和 $x(k)$ 的并行序列相异或.

对比文献[2,3]易知:文献中并行扰码的矩阵法实际是上述方法的特例,正因为扰码序列的特殊性(即 A, B 特殊性)使得 $s(k+N)$ 和 $s_i(k+j)$ 的产生有大量可合并的部分,从而使并行扰码简化为矩阵法. 值得一提的是,文献中的方法在 $N > n$ 时均有对矩阵 A 扩展的问题,增加了问题复杂性;而本文提出的方法则不需扩展矩阵,对任意并行路数 N 均可完成并行化,计算公式均是统一形式.

3.3 (M, N)并行实现的讨论

$M > 1$ 时, (r, m, n) LRS 的 (M, N) PLRS 实现可在前述方法的基础上类似得到. 一种最简单直接的实现即:先将原 LRS 并行化为 $(1, MN)$ PLRS,设原 LRS 的一个输出序列对应的 MN 个并行序列为 $Z_1 \sim Z_{MN}$,然后分别对其中的相邻 M 个序列 $Z_{(i-1)M+1} \sim Z_{iM}$ $(i=1,2,\cdots,N)$ 进行并/串转换得到序列 Y_i $(i=1,2,\cdots,N)$,此 N 个并行序列 Y_i 即为 (M, N) 并行实现的输出. 所需寄存器个数为 $n+MN$,其中的 MN 个寄存器用于 N 个并串转换电路.

另外,对于某些情况,即对状态序列进行 M 采样得到的 M 个并行序列可由相同的生成多项式生成的此类特殊的 LRS 而言(如设某最长线性移位寄存器序列对应的生成多项式为 $G(x)$,则该序列的 2^i 采样得到的 2^i 个并行序列均可由相同的生成多项式 $G(x)$ 产生[1]),由于 M 个并行序列之间的相似性,对应的 (M, N) 并行实现也可有特殊方法. 下面仅以一个状态序列的 (M, N) 并行实现为例给出一种新的实现方法. 设该状态序列为 $\{s_i\}$, $i=1,2,\cdots$,对应的 (M, N) 并行序列为 $Y_1 \sim Y_N$: $Y_1=\{s_1, s_2, \cdots, s_M; s_{MN+1}, s_{MN+2}, \cdots, s_{M(N+1)}; \cdots\cdots\}$, $\cdots$, $Y_j=\{s_{M(j-1)+1}, s_{M(j-1)+2}, \cdots, s_{Mj}; s_{M(N+j-1)+1}, s_{M(N+j-1)+2}, \cdots, s_{M(N+j)}; \cdots\cdots\}$, $\cdots$, $Y_N=\{s_{M(N-1)+1}, s_{M(N-1)+2}, \cdots, s_{MN}; s_{M(2N-1)+1}, s_{M(2N-1)+2}, \cdots, s_{2MN}; \cdots\cdots\}$. 由于序列 $\{s_i\}$ 的 M 个 M 采样序列 $H_i=\{s_i, s_{M+i}, s_{2M+i}, \cdots\}$, $i=1,2,\cdots,M$,可由相同生成多项式产生,设此生成多项式为 $g(x)$,其最高次数为 n,由此可写出对应的具有例 1 中 A_I 类型的状态转移矩阵 $A_{g(x)}$. 将采样序列 H_i 分别进行 N 路并行化(即 $(1, N)$ 并行实现),由前述方法或文献[2,3]中的并行方法可知(主要通过计算 $A^N_{g(x)}$ 等矩阵来实现),由于 $g(x)$ 和 $A_{g(x)}$ 对所有的 H_i 均相同,因此其分别的 N 路并行实现对应的电路对所有的 H_i 也完全相同,差别仅在于不同的 H_i 对应的初值不同. 对于 A_I 类型的形式,每个 H_i 的初值即为该序列的前 n 个元素. 图 3 为 (M, N) 并行实现框图,图示为 $N > n$ 的情况;当 $N < n$ 时,虚线框内的部分消失,其它部分仍类似. 以下也仅以 $N > n$ 的情况对实现原理进行说明, $N < n$ 时类似. 图中的每组 M 个移位寄存器向右进行移位,即 $r_{i+1} \rightarrow r_i$, $i=1,2,\cdots,M-1$,第 j 组移位寄存器的初值为 $r_M=s_{Mj}, \cdots, r_1=s_{M(j-1)+1}$, $j=1,2,\cdots,n$. 考虑序列 H_1 有: $Y_1(1)=s_1$, $Y_2(1)=s_{M+1}, \cdots, Y_n(1)=s_{M(n-1)+1}$ 构成序列 H_1 的前 n 个元素; $Y_{n+1}(1) \sim Y_N(1)$ (即 $H_1(n+1) \sim H_1(N)$) 可由 $Y_1(1) \sim Y_n(1)$ 的线性组合生成. 且 $[H_1((k+1)N+n), \cdots, H_1((k+1)N+2), H_1((k+1)N+1)]^T=A^N_{g(x)}[H_1(kN+n), \cdots, H_1(kN+2), H_1(kN+1)]^T$, $k=0,1,\cdots$ 故 $H_1(N+1)=s_{MN+1}=Y_1(M+1)$, $H_1(N+2)=s_{M(N+1)+1}=Y_2(M+1), \cdots, H_1(N+n)=s_{M(N+n-1)+1}=Y_n(M+1)$,送入对应寄存器组的最左端元素(即 r_M),在移位 M 次后正好送出到 $Y_1 \sim Y_n$,并依此类推. 同样,对于相应于 $H_i(i=2, \cdots, M)$ 的 $Y_1(i) \sim Y_n(i)$ 及 $Y_{n+1}(i) \sim Y_N(i)$,也有同样结果. 从而得到图 3 的结构,图中 $Z_1 \sim Z_n$ 在不同时刻分别对应不同的 $H_i(kN+1) \sim H_i(kN+n)$. 该方法实现时所需寄存器个数为 Mn 个,当 $N \geq n$ 时,该方法所需寄存器个数更少一些. 由于实际中最常用的 M、N 值均为 2 的方幂大多属于该情况,因此该方法也有较大应用价值.

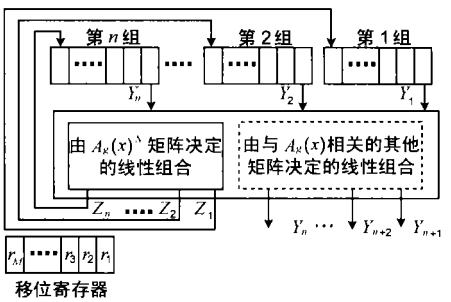

图 3 一种 (M, N) 并行方法的结构框图

4 非线性寄存器系统的并行化

由于输出的并行化只需将状态和输入的并行序列按照函数 $g_j(j=1, \cdots, m)$ 相应组合即可实现,NLRS 的并行化实际也是状态序列生成的并行化问题.

4.1 线性化矩阵法

文献[7]中给出一种非线性移位寄存器的线性化方法,但不适用于并行实现. 下面则对 NLRS 给出一种新的状态转移线性化方法,从而得到新的并行方法——线性化矩阵法.

由于组合电路均可用"非"(记为 $\bar{x}$)、"与"(记为 x_1x_2)和"或"(记为 $x_1 \vee x_2$)电路实现,且有 $\bar{\bar{x}}=x+1$, $x_1 \vee x_2=x_1x_2+x_1+x_2$(注:"+"均指模 2 和,即异或),故所有组合电路均可用"与"和"异或"电路实现. 对有 n 个寄存器的 NLRS,设向量 $S=(1, s_1, s_2, \cdots, s_n, s_1s_2, s_1s_3, \cdots, s_{n-1}s_n, \cdots\cdots, s_1s_2\cdots s_n)^T_{2^n \times 1}$,即状态及其各元素间的"与"项构成的向量(当状态转

移与输入有关时,还需将输入也作为状态同样把其相关项加入 S 中),则可有 $S(k+1)=AS(k)$,A 为对向量 S 的转移矩阵,即向量 S 的下一时刻值可由 S 的当前时刻值的线性组合得到. 另外,当 $s_1 \sim s_n$ 下一时刻值与 S 中的某些"与"项无关时,可省略这些"与"项以减小 A 的维数. 这样,就将 NLRS 的状态变换转化为对向量 S 的转移矩阵 A 的操作之上,从而可采用 LRS(包括线性移位寄存器)已有的矩阵法实现并行化. 然而,由于转移矩阵 A 的最大维数为 2^n,该方法的最大计算复杂度随 n 的增加增长很快,故仅适于 n 较小或 A 的维数可减到足够小的情况.

4.2 推广延时因子法

对于特殊的 NLRS ——非线性移位寄存器而言,由于各个移位寄存器发出的序列是平移等价的,其并行化还可采用一些特殊方法. 下面先定义非线性移位寄存器中的生成多项式,然后将文献[3]中适于线性移位寄存器的延时因子法推广到非线性移位寄存器中.

对非线性移位寄存器,图 1 中所示的反馈函数 $f(s_1, s_2, \cdots, s_n)$ 为非线性的组合电路,以"与"和"异或"表示,反馈函数可设为:

$$f(s_1, s_2, \cdots, s_n) = \sum_{r=0\,1}^{n} \sum_{\le i_1 < i_2 < \cdots < i_r \le n} c_{i_1 i_2 \cdots i_r} s_{i_1} s_{i_2} \cdots s_{i_r} \quad (8)$$

其中 $c_{i_1 i_2 \cdots i_r} = 0$ 或 1;当 $r=0$,约定 $c_{i_1 i_2 \cdots i_r} = c_0$,$s_{i_1} s_{i_2} \cdots s_{i_r} = 1$. 定义对应的生成多项式为 $g(s) = 1 + \sum_{r=0\,1}^{n} \sum_{\le i_1 < i_2 < \cdots < i_r \le n} c_{i_1 i_2 \cdots i_r} s^{i_1} s^{i_2} \cdots s^{i_r}$,其中 $s^{i_1} s^{i_2} \cdots s^{i_r}$ 表示各项的"与",即不能写成 $s^{i_1 + i_2 + \cdots + i_r}$ 的形式.

记反馈函数给出的序列(即寄存器 s_1 的输入)为 $\{a_m\}$,则 m 时刻非线性移位寄存器的状态为 $\{a_{m-1}, a_{m-2}, \cdots, a_{m-n}\}$. 由反馈函数或生成多项式写出序列 $\{a_m\}$ 的递推关系式为:

$$a_m = \sum_{r=0\,1}^{n} \sum_{\le i_1 < i_2 < \cdots < i_r \le n} c_{i_1 i_2 \cdots i_r} a_{m-i_1} a_{m-i_2} \cdots a_{m-i_r} \quad (9)$$

参照文献[3],引入延时因子 D,即 $a_{m+1} = \mathrm{D}a_m$,$a_{m-1} = \mathrm{D}^{-1} a_m$,$\mathrm{D}^{-1}$ 为逆因子,且 $\mathrm{D}^0 = 1$,$\mathrm{D}^i \mathrm{D}^j = \mathrm{D}^{i+j}$,$i$,$j$ 为整数. 由于非线性移位寄存器的反馈函数中可能存在"与"项和"+1"(即与 1 异或,为避免与 $\mathrm{D}^0 = 1$ 中的 1 混淆,用"1"表示),因此需要在延时因子 D 的作用下对它们的操作性质作一说明. 为避免符号混乱,定义 $\mathrm{D}^i \cdot \mathrm{D}^j$($i$,$j$ 为整数)为两者的"与"操作,其满足一般"与"的性质;且有 $\mathrm{D}(\mathrm{D}^i \cdot \mathrm{D}^j) = \mathrm{D}^{i+1} \cdot \mathrm{D}^{j+1}$,$\mathrm{D}^{-1}(\mathrm{D}^i \cdot \mathrm{D}^j) = \mathrm{D}^{i-1} \cdot \mathrm{D}^{j-1}$;并约定 $\mathrm{D}1 = 1$,$1 \cdot \mathrm{D}^i = \mathrm{D}^i$;$\mathrm{D}^i + \mathrm{D}^j$ 仍满足"模二和"的性质. 故有

$$a_m = \sum_{r=1\,1}^{n} \sum_{\le i_1 < i_2 < \cdots < i_r \le n} c_{i_1 i_2 \cdots i_r} \mathrm{D}^{-i_1} \cdot \mathrm{D}^{-i_2} \cdot \cdots \cdot \mathrm{D}^{-i_r} a_m + c_0 1, \quad (10)$$

$$a_{m+1} = \mathrm{D}a_m = \Big(\sum_{r=1\,1}^{n} \sum_{\le i_1 < i_2 < \cdots < i_r \le n} c_{i_1 i_2 \cdots i_r} \mathrm{D}^{-i_1+1} \cdot \mathrm{D}^{-i_2+1} \cdot \cdots \cdot \mathrm{D}^{-i_r+1} \Big) a_m + c_0 1, \quad (11)$$

并将式(11)右端出现的 $\mathrm{D}^0 a_m = a_m$ 用式(10)带入,同样 $a_{m+2} = \mathrm{D}^2 a_m = \cdots, \cdots$,并依此类推. 即后续序列 a_{m+k} 均可用当前状态 $\{a_{m-1}, a_{m-2}, \cdots, a_{m-n}\}$ 表示. 省略式(10)、(11)等两边的 a_m,即有如下递推关系式

$$\mathrm{D}^0 = 1 = \sum_{r=1\,1}^{n} \sum_{\le i_1 < i_2 < \cdots < i_r \le n} c_{i_1 i_2 \cdots i_r} \mathrm{D}^{-i_1} \cdot \mathrm{D}^{-i_2} \cdot \cdots \cdot \mathrm{D}^{-i_r} + c_0 1; \quad (12)$$

$$\mathrm{D}^0 = \sum_{r=1\,1}^{n} \sum_{\le i_1 < i_2 < \cdots < i_r \le n} c_{i_1 i_2 \cdots i_r} \mathrm{D}^{-i_1} \cdot \mathrm{D}^{-i_2} \cdot \cdots \cdot \mathrm{D}^{-i_r} + c_0 1 = \cdots \quad (13)$$

正如文献[3]所述,以上的延时因子的递推关系代表了序列的递推关系. 类似的,设序列的周期为 T,则 $\mathrm{D}^T = 1$,即以上递推关系式最多有 T 个. 同理,当求非线性移位寄存器的 N 路并行实现时,设电路的当前状态为 $\{\mathrm{D}^i, i = -\max(n, N), \cdots, -1\}$,一个并行周期(即 N 个串行时钟周期)过后,电路应跳转 N 个状态,成为 $\{\mathrm{D}^{N+i}, i = -\max(n, N), \cdots, -1\}$,状态转移过程为 $\mathrm{D}^{N+i} \to \mathrm{D}^i$,$i = -\max(n, N), \cdots, -1$. 按照状态转移关系和递推关系将电路连接起来即可得到 N 路的并行实现.

下面以图 4(a)中输出序列 s_3 的 5 路并行实现为例对推广延时因子法进行具体说明. 反馈函数 $f(s_1, s_2, s_3) = s_1 \bar{s}_2 + \bar{s}_3 = 1 + s_1 + s_3 + s_1 s_2$,$c_0 = 1$;生成多项式 $g(s) = 1 + 1 + s^1 + s^3 + s^1 s^2$. 延时因子 D 的递推关系式为:

$$\begin{aligned}
1 &= \mathrm{D}^0 = 1 + \mathrm{D}^{-1} + \mathrm{D}^{-3} + \mathrm{D}^{-1} \cdot \mathrm{D}^{-2} \\
\mathrm{D} &= 1 + \mathrm{D}^0 + \mathrm{D}^{-2} + \mathrm{D}^0 \cdot \mathrm{D}^{-1} = \mathrm{D}^{-1} + \mathrm{D}^{-2} + \mathrm{D}^{-3} + \mathrm{D}^{-1} \cdot \mathrm{D}^{-3} \\
\mathrm{D}^2 &= \mathrm{D}^0 + \mathrm{D}^{-1} + \mathrm{D}^{-2} + \mathrm{D}^0 \cdot \mathrm{D}^{-2} = 1 + \mathrm{D}^{-3} + \mathrm{D}^{-1} \cdot \mathrm{D}^{-2} + \mathrm{D}^{-2} \cdot \mathrm{D}^{-3} \\
\mathrm{D}^3 &= 1 + \mathrm{D}^{-2} + \mathrm{D}^0 \cdot \mathrm{D}^{-1} + \mathrm{D}^{-1} \cdot \mathrm{D}^{-2} = 1 + \mathrm{D}^{-2} + \mathrm{D}^{-1} \cdot \mathrm{D}^{-3} \\
\mathrm{D}^4 &= 1 + \mathrm{D}^{-1} + \mathrm{D}^0 \cdot \mathrm{D}^{-2} = 1 + \mathrm{D}^{-1} + \mathrm{D}^{-2} + \mathrm{D}^{-2} \cdot \mathrm{D}^{-3}
\end{aligned} \quad (14)$$

根据 $\mathrm{D}^{N+i} \to \mathrm{D}^i$ 的状态转移过程可得对应的状态转移图(图 4(b)). 按照图 4(b)和式(14)将寄存器重新连接起来,即可得图 4(c)的电路,$\mathrm{D}^{-1} \sim \mathrm{D}^{-5}$ 分别对应图中的 $P_1 \sim P_5$,其均为寄存器采样输出;若以 D^1、D^0、D^{-1}、D^{-2}、D^{-3} 作为并行输出,即 P'_4、P'_5、$P_1 \sim P_3$,则可省略寄存器 s_4 和 s_5. 因此,用推广延时因子法实现并行化时,所需寄存器个数为 n,与原非线性移位寄存器相同;当并行输出为寄存器采样输出时,所需寄存器个数为 $\max(n, N)$.

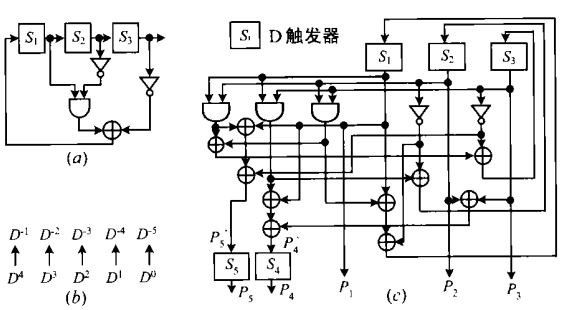

图 4　非线性移位寄存器的并行实现

5 结语

本文对一般 LRS 提出的具有统一计算公式的(1, N)并行

化方法有更广泛的应用范围. 对某些情况下的 (M, N) 并行提出的实现方法也可广泛用于诸如几路信号通过字节或字间插复用为一路信号(如 SDH 中 STM-n 各速率等级的复用)的并行扰码等电路中. 对 NLRS 的并行化给出的线性化矩阵法的最大计算复杂度随 n(系统中寄存器的个数)的增加呈指数增长;而对非线性移位寄存器提出的推广延时因子法,运算量随 n 或 N(并行路数)的增大而增加很小,特别适于 n 或 N 均较大的情况.

参考文献:

[1] 丁石孙. 线性移位寄存器序列[M]. 上海:上海科学技术出版社,1982.

[2] Choi S W. Parallel scrambling techniques for digital multiplexer [J]. AT&T Tech J,1986,9/10:123 - 136.

[3] 张晓如,曾烈光. 帧同步扰码器的并行化技术[J]. 通信学报,1996,17(2):126 - 130.

[4] Albertengo G,Sisto R. Parallel CRC generation[J]. IEEE Micro,Oct 1990,10(5):63 - 71.

[5] Kim SC,Lee BG. Parallel shift register generators-theory and applications to parallel scrambling in multibit-interleaved multiplexing environment[J]. IEEE Transactions on Communications,1995,43(2 - 4):1844 - 1853.

[6] Kim SC,Lee BG. Realizations of parallel and multibit parallel shift register generators[J]. IEEE Transactions on Communications,1997,45(9):1053 - 1060.

[7] 万哲先,代宗铎,刘木兰,等. 非线性移位寄存器[M]. 北京:科学出版社,1978.

作者简介:

秦晓懿 女,1974 年 3 月生于重庆,1996 年和 2001 年获清华大学电子工程系学士学位和博士学位,主要从事 SDH、以太网接入技术研究与集成化设计.

王瀚晟 男,1971 年 5 月生于北京,1994 年和 1998 年获清华大学电子工程系学士学位和博士学位,主要从事光通信系统与网络的研究及 ASIC 实现.

曾烈光 男,1947 年 12 月生于四川南部县,教授,博士生导师,1970 年毕业并任教于清华大学电子工程系,主要从事通信网络与光通信系统的研究,包括 PDH、SDH、ATM、接入网等的研究与 ASIC 实现,曾获电子部科技进步和国家发明一、二等奖多项,专利多项,全国青年科学家提名奖.