

面向可重构并行化处理的线性反馈移位寄存器 统一架构研究

郑诚玮,戴紫彬,李 伟

(解放军信息工程大学,河南 郑州 450001)

摘 要: 为了解决目前线性反馈移位寄存器实现方式在运算速度、灵活性和安全性三个方面存在的不足,研究了 Fibonacci LFSR 和 Galois LFSR 的结构特点,提取出了两者可重构和并行化实现方式的共同点,研究并实现了两者的统一架构.统一架构的可重构实现方式提高了电路的灵活性,并行化处理方式提高了电路的运算速度,灵活可变的结构增加了电路的安全性.和文献[6]中的设计相比,面积减小了 47.4%,电路延迟缩减了 11.5%.

关键词: Fibonacci LFSR; Galois LFSR; 并行化; 可重构; 统一架构

中图分类号: TP309.7

文献标识码: A

文章编号: 1000-7180(2015)11-0111-05

Research on Linear Feedback Shift Register United Architecture for Parallel and Reconfigurable Processing

ZHENG Cheng-wei, DAI Zi-bin, LI Wei

(PLA Information Engineering University, Zhengzhou 450004, China)

Abstract: The parallel and reconfigurable united architecture of LFSR is represented to improve the speed, flexibility and security of communication system and cipher algorithms. It can be reconfigured to Fibonacci LFSRs and Galois LFSRs according to different applications. The random lengths and feedback taps can be achieved to meet the demands of different applications. The parallel updating method has an obvious advantage on speed and efficiency. Furthermore, the flexibility and complexity increases the security of the communication system and cipher algorithms. Compared with design in Ref [6], the area decreases 47.4% and the delay time shortens 11.5%.

Key words: Fibonacci LFSR, Galois LFSR, Parallel, Reconfigurable, United architecture

1 引言

在通信领域和密码算法中,线性反馈移位寄存器(Linear Feedback Shift Register, LFSR)是一种重要的电路^[1],它广泛应用于 M 序列产生、信道编码、信道串扰、CRC(Cyclic Redundancy Check)校验^[2]、密码算法实现等多个方面.在实际应用中,LFSR 主要有两种类型^[3],即 Fibonacci LFSR 和 Galois LFSR.在目前应用中,LFSR 通常采用固定结构,完成单一功能.这样的实现方式主要有三个方面的不足:首先,它不能根据应用场合的变化来改变

LFSR 的类型、长度和反馈函数,其灵活性较差^[2,3];其次,密码算法硬件实现时,LFSR 采用固定结构更容易被攻击,其安全性较差;此外,大多数 LFSR 采用串行更新的方式^[4],其运算速度较慢.

为了解决上述问题,本文采用并行化和可重构的设计技术,实现 Fibonacci LFSR 和 Galois LFSR 的统一架构.该架构能够根据不同的应用场合,实现不同类型、不同结构的 LFSR.并行更新方式能够在—一个时钟周期完成多步更新,提升了 LFSR 运算的速度和效率.同时,灵活多变的结构保证了电路的安全性.

收稿日期:2015-01-20;修回日期:2015-03-11

基金项目:国家自然科学基金(61404175)

2 LFSR 可重构实现方式研究

2.1 Fibonacci LFSR 可重构实现方式研究

采用可重构方式实现 Fibonacci LFSR 时,其结构包括两个部分^[5],即移位寄存器和线性反馈函数.移位寄存器由 n 个串行连接 D 触发器构成.线性反馈函数由 $n-1$ 个异或门单元构成.其中, n 代表 LFSR 的长度,系数 $f_i \{i=0, \dots, n-1\}$ 用于实现线性反馈函数.当 $f_i=1$ 时,则 x_i 被选中,参与状态更新;反之,当 $f_i=0$ 时, x_i 不参与更新.如图 1 所示.

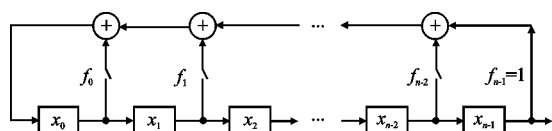


图 1 Fibonacci LFSR 可重构实现方式

在 Fibonacci LFSR 可重构实现方式中, Fibonacci LFSR 的状态更新函数可以描述如下:

$$X(k+1) = \underbrace{\begin{pmatrix} f_0 & f_1 & \cdots & f_{n-2} & f_{n-1} \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}}_A \cdot \underbrace{\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{pmatrix}}_{X(k)} \quad (1)$$

式中, n 阶矩阵 A 叫做 n 级 Fibonacci LFSR 的状态转移矩阵, $a_{i,j}$ 为矩阵 A 中的元素,其中 $a_{i,j} \in \text{GF}(2)$, $1 \leq i, j \leq n$. 特别是, $A^0 = I_n, I_n$ 为单位矩阵.

2.2 Galois LFSR 可重构实现方式研究

Galois LFSR 的可重构实现方式与 Fibonacci LFSR 类似,同样由移位寄存器和线性反馈函数两部分构成.如图 2 所示.

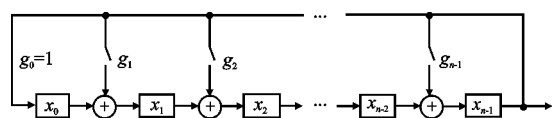


图 2 Galois LFSR 可重构实现方式

在 Galois LFSR 可重构实现方式中, Galois LFSR 的状态更新函数可以描述如下:

$$X(k+1) = A \cdot X(k) = \begin{pmatrix} 0 & 0 & \cdots & 0 & g_0 \\ 1 & 0 & \cdots & 0 & g_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & g_{n-2} \\ 0 & 0 & \cdots & 1 & g_{n-1} \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{pmatrix} \quad (2)$$

式中, n 阶矩阵 A 叫做 n 级 Galois LFSR 的状态转移

矩阵, $a_{i,j}$ 为矩阵 A 中的元素,其中 $a_{i,j} \in \text{GF}(2)$, $1 \leq i, j \leq n$. 特别是, $A^0 = I_n, I_n$ 为单位矩阵.

2.3 LFSR 统一架构可重构实现方式研究

通过对 Fibonacci LFSR 和 Galois LFSR 可重构实现方式的研究,分析得到了两者状态更新函数的共同点.由表达式(1)和(2),可以得到以可重构方式实现统一架构时的状态更新函数为

$$\begin{aligned} X(k+1) &= A \cdot X(k) \\ &= \begin{pmatrix} f_0 & f_1 & \cdots & f_{n-2} & f_{n-1} + g_0 \\ 1 & 0 & \cdots & 0 & g_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & g_{n-2} \\ 0 & 0 & \cdots & 1 & g_{n-1} \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{pmatrix} \quad (3) \end{aligned}$$

根据表达式(3),可以得到可重构 LFSR 统一架构的结构如图 3 所示.

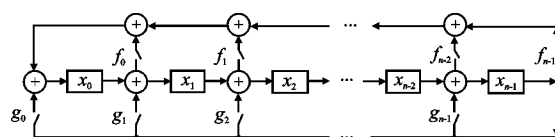


图 3 LFSR 统一架构可重构实现方式

3 LFSR 统一架构并行更新方式研究

基于对 Fibonacci LFSR 和 Galois LFSR 可重构统一架构的研究,通过表达式(3),可以推导出并行完成 d 步更新时的状态更新过程如下:

$$X(k+d) = A \cdot X(k+d-1) = \cdots = A^d \cdot X(k).$$

由上述推导式可知,要计算 $X(k+d)$,需要先计算 A^d , A^d 为 d 步更新的状态转移矩阵.由于 A^d 的计算过程比较复杂,为了方便研究,本文先分两种情况进行研究:(1) $f_i = 0 (0 \leq i \leq n-1)$,即统一架构实现 Fibonacci LFSR;(2) $g_i = 0 (0 \leq i \leq n-1)$,即统一架构实现 Galois LFSR.在基于上述两种情况研究基础上,再提取出 A^d 的计算方法.

3.1 Fibonacci LFSR 并行更新方式研究

统一架构实现 Fibonacci LFSR 时, $g_i = 0, (0 \leq i \leq n-1)$, f_i 正常工作.此时:

$$A = \begin{pmatrix} f_0 & f_1 & \cdots & f_{n-2} & f_{n-1} \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

为了得到完成 d 步更新的状态转移矩阵 A^d ,本

文需逐步进行推导.若用 A_{rj}^i 代表 A^i 的第 j 行,则状态转移矩阵 A 可以表示如下:

$$A = A^1 = (A_{r1}^1 \ A_{r2}^1 \ \cdots \ A_{r_{m-1}}^1 \ A_{rm}^1)^T.$$

状态转移矩阵 A^d 的并行更新过程如下:

$$A^2 = A^1 \cdot A^1 =$$

$$(A_{r1}^2 \ A_{r1}^1 \ A_{r2}^1 \ \cdots \ A_{r_{m-1}}^1)^T,$$

$$A^3 = A^1 \cdot A^2 =$$

$$(A_{r1}^3 \ A_{r1}^2 \ A_{r1}^1 \ A_{r2}^1 \ \cdots \ A_{r_{m-1}}^1)^T$$

$\vdots$

$$A^d =$$

$$A^1 \cdot A^{d-1} =$$

$$(A_{r1}^d \ A_{r1}^{d-1} \ \cdots \ A_{r1}^1 \ A_{r2}^1 \ \cdots \ A_{r_{m-1}}^1)^T.$$

其中,当 $1 \leq i \leq d$ 时,

$$A^i = (A_{r1}^i \ \cdots \ A_{r1}^1 \ A_{r2}^1 \ \cdots \ A_{r_{m-i+1}}^1)^T.$$

由上述推导过程可知,当计算 A^d 时:

(1) $A_{ri}^d (d+1 \leq i \leq n)$ 由 $A_{ri}^1 (1 \leq i \leq n-d+1)$ 直接得到;

1) 直接得到;

(2) $A_{ri}^d (1 \leq i \leq d)$ 由 $A_{ri}^1 (1 \leq i \leq d)$ 得到,而

A_{ri}^i 需要进行计算.

由结论可知,统一架构实现 Fibonacci LFSR 功能时,计算 A^d 的关键在于计算 $A_{ri}^i (1 \leq i \leq d)$. 若用 $a_{p,q}^i$ 表示矩阵 A^i 的第 p 行、第 q 列的元素,其中 $1 \leq i \leq d, 1 \leq p, q \leq d$. 则 A_{ri}^i 的计算方法如下:

$$\therefore A^i = A^{i-1} \cdot A^1$$

$$\therefore A_{ri}^i = A_{ri}^{i-1} \cdot A^1 = (a_{1,1}^{i-1} \ a_{1,2}^{i-1} \ \cdots \ a_{1,n}^{i-1}) \cdot$$

$$A = (A_{ri}^{i-1} \ll 1) + a_{1,1}^{i-1} \cdot A_{ri}^1$$

因此, A_{ri}^i 计算表达式为

$$A_{ri}^i = (A_{ri}^{i-1} \ll 1) + a_{1,1}^{i-1} \cdot A_{ri}^1 \quad (4)$$

其中,元素 $a_{1,1}^1, a_{1,1}^2, \dots, a_{1,1}^d$ 的计算方法为

$$a_{1,1}^i = \sum_{j=1}^i a_{1,j}^1 \cdot a_{1,1}^{i-j} \quad (0 \leq i \leq d) \quad (5)$$

3.2 Galois LFSR 并行更新方式研究

统一架构实现 Galois LFSR 时, $f_i = 0, (0 \leq i \leq n-1), g_i$ 正常工作. 此时:

$$A = \begin{pmatrix} 0 & 0 & \cdots & 0 & g_0 \\ 1 & 0 & \cdots & 0 & g_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & g_{n-2} \\ 0 & 0 & \cdots & 1 & g_{n-1} \end{pmatrix}.$$

若用 A_{gj}^i 代表 A^i 的第 j 列,则状态转移矩阵 A 可以表示如下:

$$A = A^1 = (A_{c1}^1 \ A_{c2}^1 \ \cdots \ A_{c_{n-2}}^1 \ A_{c_{n-1}}^1).$$

当 $1 \leq i \leq d$ 时,状态转移矩阵的并行更新方法

可以表示如下:

$$A^2 = A^1 \cdot A^1 = (A_{c2}^1 \ A_{c3}^1 \ \cdots \ A_{cn}^1 \ A_{cn}^2)$$

$$A^3 = A^2 \cdot A^1 =$$

$$(A_{c3}^1 \ A_{c4}^1 \ \cdots \ A_{cn}^1 \ A_{cn}^2 \ A_{cn}^3)$$

$\vdots$

$$A^d = A^{d-1} \cdot A^1 =$$

$$(A_{cd}^1 \ \cdots \ A_{cn-1}^1 \ A_{cn}^1 \ A_{cn}^2 \ \cdots \ A_{cn}^d).$$

由上述推导过程可知,当计算 A^d 时:

(1) $A_{ci}^d (1 \leq i \leq n-d)$ 由 $A_{ci}^1 (d \leq i \leq n-1)$ 直接得到;

(2) $A_{ci}^d (n-d+1 \leq i \leq n)$ 由 $A_{ci}^1 (1 \leq i \leq d)$ 得到,而 A_{ci}^i 需要进行计算.

由结论可知,统一架构实现 Galois LFSR 功能时,计算 A^d 的关键在于计算 $A_{ci}^i (1 \leq i \leq d)$. 若用 $a_{p,q}^i$ 表示矩阵 A^i 的第 p 行、第 q 列的元素,其中 $1 \leq i \leq d, 1 \leq p, q \leq d$. 则 A_{ci}^i 的计算方法如下:

$$\therefore A^i = A^1 \cdot A^{i-1}$$

$$\therefore A_{ci}^i = A^1 \cdot A_{ci}^{i-1} = ((A_{ci}^{i-1})^T \ggg 1)^T + a_{n,n}^{i-1} \cdot$$

$$A_{ci}^1$$

因此, A_{ci}^i 计算表达式为

$$A_{ci}^i = ((A_{ci}^{i-1})^T \ggg 1)^T + a_{n,n}^{i-1} \cdot A_{ci}^1 \quad (6)$$

其中,元素 $a_{n,n}^1, a_{n,n}^2, \dots, a_{n,n}^d$ 计算方法如下:

$$a_{n,n}^i = \sum_{j=1}^i a_{n,n}^{i-j} \cdot a_{n-j+1,n}^1 \quad (1 \leq i \leq d) \quad (7)$$

3.3 LFSR 统一架构并行更新方式研究

结合 3.1 节和 3.2 节中的结论,可以实现 LFSR 可重构统一架构的并行化更新. 用信号 Type 表示 LFSR 的类型, Type = 0 表示实现 Fibonacci LFSR, Type = 1 表示实现 Galois LFSR. 结合函数表达式(4)、(5)、(6)、(7),可以得到统一架构的配置信息生成算法和状态更新算法. 其中,配置信息生成算法表述如下:

算法 1: 统一架构配置信息生成算法

输入: $A; d; \text{Type} \in (0, 1)$

输出: $A_{r1}^2, A_{r1}^3, \dots, A_{r1}^d; A_{cn}^2, A_{cn}^3, \dots, A_{cn}^d$

1. for $i = 1$ to $d-1$

2. $a_{\text{temp}} = 0$;

3. for $j = 1$ to i

4. $a_{\text{temp}} = a_{\text{temp}} + \text{!Type} \cdot a_{1,j}^1 \cdot a_{1,1}^{i-j} + \text{Type} \cdot a_{n-j+1,n}^1 \cdot a_{n,n}^{i-j}$;

5. endfor j

6. if Type = 0

7. $a_{1,1}^{i+1} = a_{\text{temp}}; A_{r1}^{i+1} = (A_{r1}^i \ll 1) + a_{1,1}^i \cdot A_{r1}^1$;

8. else

电路面积和延迟是衡量电路设计性能的重要参数. 本文设计的 LFSR 统一架构与文献[4]、文献[5]、文献[6]中的可重构 LFSR 进行性能对比, 其性能对比结果如表 1 所示.

表1 性能对比结果

设计对象	文献[4]	文献[5]	文献[6]	本设计
位宽/bit	256	256	256	256
并行步数	1	1	1~8	1~8
面积/ μm^2	37 700	36 900	98 300	51 700
延迟/ns	5.46	1.50	1.31	1.16

根据表1中统计数据,可以得到不同设计对象的性能对比结果图如图7所示。

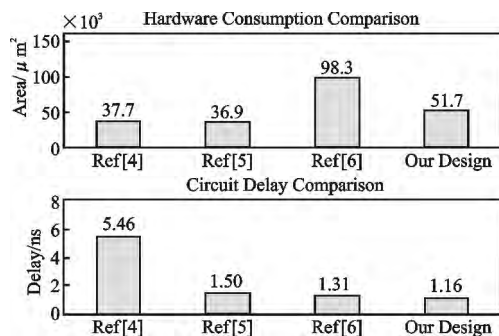


图7 不同设计性能对比

从图7可以看出,文献[4]中的LFSR采用串行更新的方法来实现反馈函数,因此相比其他设计,文献[4]中设计的电路关键路径最长,延迟最大。文献[5]中采用并行更新的方法实现反馈函数,相比文献[4],大大缩短了关键路径和电路延迟。文献[6]相比文献[5],进一步优化了关键路径,但是由于文献[6]中LFSR为了适用于VLIW(Very Long Instruction Word)系统,增加了配置信息寄存器,因此,其硬件电路面积急剧增加。本设计采用可重构并行化的设计方法,同时优化了电路面积和速度。相比文献[6],电路面积缩减了47.4%,电路延迟缩短了11.5%。此外,文献[4-6]只实现了Fibonacci LFSR一种类型的LFSR,而本架构可实现了Fibonacci LFSR和Galois LFSR两种类型的LFSR,进一步节约了电路资源,增加了电路的灵活性。

5 结束语

本文提出了面向可重构和并行化处理的LFSR统一架构,根据不同应用场合,能够实现Fibonacci LFSR和Galois LFSR两种类型的LFSR。可重构设计支持生成任意长度和反馈函数的LFSR,并行化处理方式能够提高电路的速度和效率。同时,架构灵

活可配置的设计,能够大幅增加硬件电路的安全性。

参考文献:

- [1] Zeng G, Dong X, Bornemann J. Reconfigurable feedback shift register based stream cipher for wireless sensor networks [J]. IEEE Wireless Communications Letters, 2013, 2(5): 559-562.
- [2] Campobello G, Patane G, Russo M. Parallel CRC realization [J]. Computers IEEE Transactions on, 2003, 52(10): 1312-1319.
- [3] Goresky M, Klapper A M. Fibonacci and galois representations of feedback-with-carry shift registers [J]. Information Theory IEEE Transactions on, 2002, 48(11): 2826-2836.
- [4] Yingjie Q. Concept and design principle of reconfigurable cipher coprocessor [J]. Computer Engineering and Applications, 2003, 12(2): 7-19.
- [5] Dai Z, Li W, Chen T, Qiao Ren. Design and implementation of a high-speed reconfigurable feedback shift register [C] // proceedings of the 4th IEEE International Conference on Circuits and Systems for Communications. Shanghai: IEEE, 2008: 338-342.
- [6] Nan L, Dai Z, Yang X. Design of configurable LFSR instructions targeted at stream cipher processing [C] // (EDSSC), Electron Devices and Solid-State Circuits. Tianjin: IEEE, 2011: 1-2.
- [7] Wei L, Zibin D, Longmei N. Research and Implementation of A High-speed Reconfigurable A5 algorithm [C] // Pacific Asia Workshop on Computational Intelligence and Industrial Application. Wuhan: IEEE, 2008: 93-97.
- [8] Krivenko S S, Krivenko S A. Many-to-many linear-feedback shift register [C] // 2014 IEEE 34th International Conference on Electronics and Nanotechnology. Kyiv: IEEE, 2014: 176-178.

作者简介:

郑诚玮 男, (1990-), 硕士研究生。研究方向为专用集成电路设计。E-mail: cwzheng660@163.com.

戴紫彬 男, (1966-), 教授, 博士生导师。研究方向为专用集成电路设计、芯片可重构设计。

李伟 男, (1983-), 博士, 讲师。研究方向为专用集成电路设计。