文章编号：1007-0249 (2003) 03-0121-04

伪随机数发生器的 FPGA 实现与研究*

束礼宝， 宋克柱， 王砚方

（中国科学技术大学 快电子学实验室，安徽 合肥 230027）

摘要：在很多实际应用中，直接利用 FPGA 产生伪随机序列的方法可以为系统设计或测试带来极大的便利。本文给出了基于线性反馈移位寄存器电路，并结合 FPGA 的特有结构，设计了一种简捷而又高效的伪随机序列产生方法。最后通过统计对比，说明了这种方法所产生的随机序列不仅可具有极长的周期，而且还具有良好的随机特性。

关键词：伪随机数发生器；线性反馈移位寄存器；FPGA

中图分类号：TN918；TN710.9 **文献标识码：**A

1 引言

随机数是虽然具有一定的统计学规律，但它是抽样值不能事先确定的数。实际中产生的随机数不是绝对随机数，是相对的，称为“伪随机数”。伪随机数发生器（Pseudo random Number Generator）在扩频通信、信息加密和系统测试等领域中有着广泛的应用。

在软件中一般都采用含有大数乘法的递推公式所计算出的一组数值来产生伪随机数列。当数列足够长时，这组数值就是近似满足均匀分布的伪随机数列。如在 C 语言下，程序初始化时调用一次 srand( ) 函数设置好随机数“种子”，然后就可以通过调用 rand( )函数来产生随机数了。

随着可编程逻辑器件（FPGA）在电子领域越来越广泛的应用，在很多高速设计和高速测试的场合下，我们希望能够在 FPGA 中直接实现伪随机序列发生器。传统的大数乘法产生伪随机数的方法不但时钟频率不能太高，并且需要消耗 FPGA 内大量的逻辑资源。本文将给出在 FPGA 内利用线性反馈移位寄存器（Linear Feedback Shift Registers）结构实现伪随机数发生器的方法；这种方法不仅结构简单，易于实现，而且所产生的伪随机序列具有周期长，随机特性好的特点。

2 线性反馈移位寄存器

图 1 是 m 级线性反馈移位寄存器（LFSR）的电路结构。其中系数因子 f_i=1 表示有连接，f_i=0 表示无连接；⊕ 表示异或（XOR）运算。

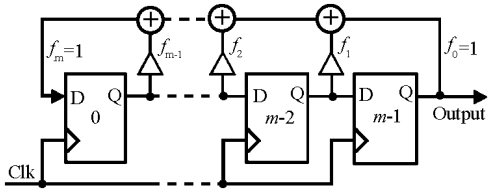

图 1 LFSR 的电路结构

显然，LFSR 的输出序列是有周期性的。因为一旦 m 个寄存器上出现了以前经历过的状态，则以后的状态将周而复始。m 级的 LFSR 最多只有 2^m 个状态，所以重复是不可避免的。假若 m 个寄存器的初始状态全为零，则 LFSR 将一直保持全零状态。因此，在初始状态非全零的前提下，LFSR 的周期 $r \le 2^m$-1。如果选取适当的反馈方式，m 级 LFSR 所产生的序列周期可以达到最大值 2^m-1，这时 LFSR 所产生的伪随机序列也称为最长序列（或者 m 序列）。LFSR 的周期只与其反馈方式有关，而不依赖于其初始状态。根据其反馈方式的不同，可以定义 LFSR 的特征多项式：

$$p(x)=\sum_{i=0}^{m} f_i x^i = x_m + f_{m-1}x^{m-1} + ... + f_1 x + 1 \qquad (1)$$

可以证明，由特征多项式 $p(x)$所对应的 m 级 LFSR 输出最长序列的充要条件[3]是：

$$\begin{cases} \text{(i)} \quad p(x)=x_m + f_{m-1}x^{m-1} + ... + f_1 x + 1 \text{ 是不可化约的;} \\ \text{(ii)} \ \text{不存在} k < 2^m - 1, \ \text{使得} (x^k + 1) \text{能被} p(x) \text{整除} \end{cases} \qquad (2)$$

* 收稿日期：2003-01-20 修订日期：2003-03-19
基金项目：国家 863 计划项目资助（2001AA602011-1）

根据（2）式所给条件，可以利用 Mathematica 4.2 软件[5]，编写求解程序，该程序对于给定的 m，可以构造出一些满足（2）式的$[f_{m-1}, \ldots, f_2, f_1]$来。但 m 值不能太大，否则计算时间会急剧增长。如果$[f_{m-1}, \ldots, f_2, f_1]$中恰好有 k 个 f_i=1，就意味着将有 k 个寄存器的输出需要经过异或运算后反馈给第一级寄存器。如果 k 较大的话，一方面会导致 LFSR 占用 FPGA 内部更多的组合逻辑资源；另一方面会导致组合逻辑的延迟增加，使得 LFSR 的工作频率受到限制。所以，在一般情况下，仅仅求解 k=2~6 的情形。

在不同应用场合下，对 LFSR 所产生的随机序列的周期长度有不同要求。随着 m 的增大，LFSR 对 FPGA 内部逻辑资源的消耗量线性增加，而 LFSR 输出的最长序列周期却呈指数增长，这一点对实际应用来说是非常有利的。当 m=63 时，最大长度序列的周期可达 9.22337×10^{18}，如果 CLK 频率为 50MHz，则重复周期超过 5800 年。在大多实际应用中，这样的序列长度是非常充裕的。

3 LFSR 的 FPGA 实现及其片内资源优化

不妨以 m=63，特征多项式为 $p(x)=x^{63}+x+1$ 的 LFSR 为例，说明伪随机数发生器的具体电路实现。因为 $p(x)=x^{63}+x+1$ 满足（2）式所给出的条件，所以在初始状态并非全零的情况下，该 LFSR 可以产生周期为 $2^{63}-1\approx9.22337\times10^{18}$ 的最长序列。

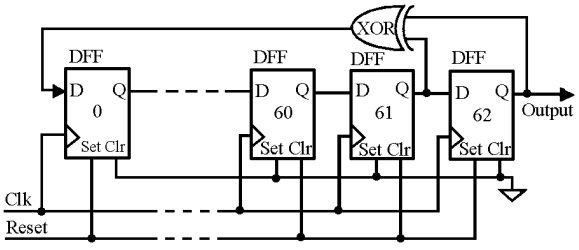

图 2 基于 DFF 的 63 级 LFSR 电路

图 2 是 63 级 LFSR 直接基于 DFF（D 触发器）所实现的电路。图中，63 个 DFF 前后级联，最后两级的输出经过一个异或门后反馈给最前级的 DFF。通过选择各个 DFF 的置 1（Set）端或清 0（Clr）端与系统复位信号（Reset）相连来设置其初始状态。为了保证复位时 LFSR 不进入全零状态，应该避免使所有的 DFF 都选择其清 0 端与 Reset 相连。

在基于 LUT（Look Up Table）的 Xilinx Virtex-II 系列 FPGA 中，核心可编程逻辑由 64~11648 个 CLB（Configurable Logic Blocks）组成，每个 CLB 含 8 个 LUT 和 8 个 DFF。一个 LUT 可以配置成：4 输入 1 输出的任意组合逻辑；16 bit 随机存储器（RAM16）；16 级移位寄存器（SRL16）等三种模式[4]。

一个异或门可以由一个 LUT 实现。所以，图 2 所示的基于 DFF 的 63 级 LFSR 需要占用 63 个 DFF 和 1 个 LUT。因为每个 CLB 只能提供 8 个 DFF，所以 63 级 LFSR 必须占用 8 个 CLB。注意到，这种实现方法所用到的 CLB 数目是由 DFF 的数量所决定的，而 8 个 CLB 内的 64 个 LUT 却只用了 1 个。在很多实际应用中，FPGA 内部 LUT 资源都很富裕，而 DFF 资源却比较紧张。因此，可以考虑将一些 LUT 配置成 SRL16 模式来实现 LFSR。即在 FPGA 配置过程中将一些 LUT 配置成 16 级移位寄存器的模式，再基于这些 16 级移位寄存器单元构造出完整的 LFSR 来。图 3 给出了 63 级 LFSR 基于 SRL16 的实现方式。

SRL16 的输入信号有 D，A[3..0], CE，CLK；输出信号有 Q 和 Q15。D 是第一级移位寄存器的输入端；A[3..0]是输出地址信号，选择 Q 端输出哪一级寄存器的内容；Q15 是第 16 级寄存器的输出；CLK 是时钟信号，CE 是时钟使能（Clock Enable）信号。

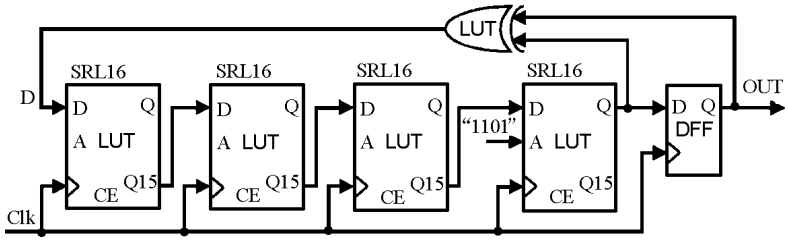

图 3 基于 SRL16 的 63 级 LFSR 电路结构

图 3 中，第四个 SRL16 的输出地址信号 A 接“1101”，选择 Q 输出第 14 级寄存器的内容。其它的 SRL16 的 A 端可接任意值。第四个 SRL16 的 Q 端接整个 LFSR 的最后一级 DFF，并与最后一级 DFF 的输出异或后反馈给第一个 SRL16 的 D 输入端。这样，整个 63 级 LFSR 就可以仅由 5 个 LUT 和 1 个 DFF 组成，从而在 Virtex-II 系列 FPGA 的单个 CLB 即可实现。一般地，基于 SRL16 的 LFSR 所占用的资源仅仅是基于 DFF 的 LFSR 所占用资源总量的 1/16~1/8；并且，基于 SRL16 结构的 LFSR 大大减少了 DFF 的使用数量。

值得一提的是：基于 DFF 和基于 SRL16 所实现的 LFSR 在初始状态的设定方法上有所不同。基于 DFF 的 LFSR 在每次复位信号（Reset）有效后，都将进入初始状态。而基于 SRL16 的 LFSR，其初始状态在系统上电过程中，在 FPGA 配置时设置，复位信号（Reset）并不影响 LFSR 的状态。

在实际应用中往往需要在 FPGA 内部产生 n 位并行伪随机序列，这可由以下三种方法之一来实现：

1）由 n 个并行的结构相同但初始态互不相同的 LFSR 产生。不难明白，结构相同的 LFSR 在不同初始状态下所产生的序列之间存在着一种移位关系。因此，这种方式会导致 n 个序列并非相互独立。如果所期望的序列长度至少为 k，那么只有在保证任一状态在 k 次移位操作之内都不会与其它状态发生重复后，这种方法才是可取的；

2）由同一 LFSR 工作在 n 倍 CLK 下将相邻的 n 次输出合并生成。这种方法需要使用 FPGA 内部的 PLL（Phase Locked Logic）或 DCM（Digital Clock Manager）模块[4]，以产生 n 倍频的时钟信号。显然，n 位随机序列是相互独立的，但有效周期变为原来的 1/n。由于 LFSR 的周期可以达到很长，这也不成问题；但这种方法需要占用 FPGA 内部的 PLL/DCM 资源，并且可能会束缚系统的最高工作频率，所以适用的场合有限；

3）由反馈方式互不相同的 n 个 LFSR 实现。显然，这种方法所产生的 n 位序列之间是相互独立的。如果 n 较大，m 级 LFSR 中适用的反馈方式不足时，应该使用 m+1，m+2，..级的 LFSR。可从参考文献[1]或[2]中查到数百种可选的反馈方式。

4　两种伪随机数的质量比较

在 1 bit 序列中，连续出现 c 个 0，称为 0 的 c 游程，连续出现 c 个 1，称为 1 的 c 游程。一个随机性很好的序列应该满足 0 的 c 游程数目与 1 的 c 游程数目基本相等。并且 2 游程的总数是 1 游程总数的 1/2，3 游程的总数是 2 游程总数的 1/2，依次类推。

假定$\{a_i\}$是周期为 r 的随机序列，对于整数 t，统计 $a_i = a_{i+t}$，i=1,2, ..,r 的数目记为 n_t；$a_j \neq a_{j+\delta}$，j=1,2, ..,r 数目记为 d_δ，则定义$\{a_i\}$的自关函数 $R(t) = (n_\delta - d_\delta)/r$。一个随机性很好的随机序列应该满足 $R(0)$=1；$R(1) \to 0$，$R(2) \to 0$，.。

4.1　LFSR 所产生最长序列的随机性质

如果 m 级 LFSR 输出的序列$\{a_i\}$是周期为 r 的最长序列，则该序列有如下性质[3]：

1）$\{a_i\}$的周期 $r=2^m-1$；

2）$\{a_i\}$在一个周期内，0 的个数比 1 的个数仅少一个；

3）$\{a_i\}$在一个周期内，0 的 c 游程数和 1 的 c 游程数相等，并且 1 游程数目占游程总数的 1/2，2 游程数目占游程总数的 1/4，..，c 游程数目占游程总数的 $1/2^c$，c=1,2,..,m-2；

4）$R(0)$=1, $R(t)$=1/r, t =1,2,..,r-1。

可见，从理论上说，LFSR 在一个周期内所输出的最长序列具有良好的随机特性。

4.2　LFSR 所产生的最长序列与 rand( )函数的随机特性的统计与比较

因为最长序列的周期随 m 的增大呈指数增长，所以由 LFSR 所产生的伪随机序列很容易即可达到很长的周期。图 2 中的 63 级 LFSR 就可以产生周期超过 9×10^{18} 的伪随机序列。

在 C 语言中，随机数种子相同时，rand( )函数就会产生相同的数列。当产生随机数的递推运算中的种子变量回到以前所经历过的数值时，数列就重复了。经过验证，在 gcc-3.0.2 或 Visual C++ 6.0 编译环境下，rand( )函数周期正好是 $2^{31} \approx 2.15\times10^9$；由于 rand( )函数的返回值中低 15 bit 是有效的随机序列（RAND_MAX=32767=2^{15}-1），所以等效于 1bit 随机序列的长度为 $15\times2^{31} \approx 3.2\times10^{10}$。

我们以图 2 中的 63 级 LFSR 为例，将其所产生的随机序列与 rand( )函数所衍生的随机序列进行了统计对比。其中，每次 rand( )函数调用后，可以按照从低有效位到高有效位的顺序提供 15 个 1 bit 序列。图 4 和图 5 分别是对 LFSR 和 rand( )函数所产生的长为 2^{32}($\approx 4.3\times10^9$)的随机序列所进行的游程总

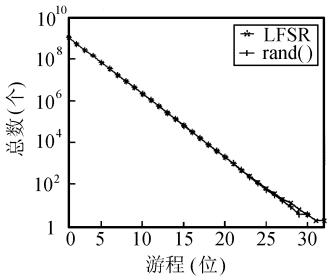

图 4 游程总数统计曲线（仿真长度 2^{32}）

数和自相关统计的结果。

由图 4 的游程总数统计曲线可见，当游程 $c<25$ 时，由 LFSR 和 rand( )函数所产生的随机序列的游程总数曲线基本重合，并且与理想的按 2^{-c} 指数衰减特性相吻合。当 $c>25$ 时，因为游程总数较小，由于统计的涨落而导致了曲线有所偏离。

由图 5 的自相关统计图可见，当位移 $t>0$ 时，由 LFSR 和 rand( )函数所产生的随机序列的自相关系数 $R(t)$ 在 0 附近振动，并且振幅很小，一般不超过 0.5×10^{-4}。与 R(0)=1 相比，R(1),R(2), ..越小，随机序列的前后独立性越好。所以，从自相关的角度来说，LFSR 和 rand( )函数所产生的随机序列的前后独立性都非常好。大多情况下，因为 LFSR 的平均振幅比 rand( )函数的要稍大一些，所以 rand( )函数所产生的随机序列比 LFSR 所产生的随机序列要更好一些；但经过多次和不同长度的仿真统计，发现当 t =165 时，rand( )函数所衍生序列的自相关系数比其它情况突然增大了 15 倍左右，其原因应该与 rand( )函数产生随机数的算法有关。

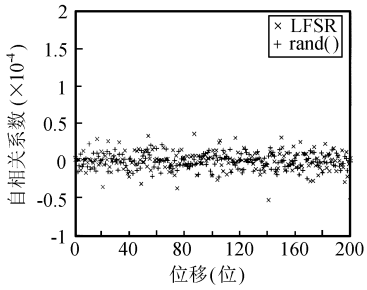

图 5 自相关统计图（仿真长度 2^{32}）

5 结论

本文讨论了利用 LFSR 结构在 FPGA 内部简捷而又高效地实现伪随机序列发生器的方法。并且结合了 FPGA 的特有结构，以基于 SRL16 的结构取代了基于 DFF 的 LFSR 结构，非常有效地节约了逻辑资源，使得最终 CLB 的使用数量可以减少到 1/16~1/8。并且，本文还从统计独立性的角度，讨论了三种利用 LFSR 结构产生 n 位并行伪随机序列方案的优劣性。最后，本文还通过对 LFSR 和 C 语言中 rand( )函数产生的随机序列所进行的游程总数和自相关系数的统计和对比，验证了 LFSR 方法所产生的随机序列不但周期很长，而且随机特性和统计独立性也非常好。

参考文献：

[1] New Wave Instruments. Linear Feedback Shift Registers[OL]. 2002-6, http://www.newwaveinstruments.com /resources/ articles / _sequence _ linear_feedback_shift_register_lfsr.htm

[2] Xilinx Inc. Linear Feedback Shift Register[EB/OL]. 2001-10. http://www.xilinx.com/ipcenter/ catalog/logicore/docs/lfsr.pdf

[3] 卢开澄. 计算密码学[M]. 长沙: 湖南教育出版社, 1993.

[4] Xilinx Inc. Virtex-II 1.5V Field-Programmable Gata Array[EB/OL]. 2001-10. http://www.xilinx.com/partinfo/ds031-1.pdf

[5] 张韵华. 符号计算系统 Mathematica 教程[M]. 北京: 科学出版社, 2001-11.

作者简介：束礼宝（1977-），男，中国科技大学近代物理系快电子学实验室博士研究生，主要研究方向为物理电子学，高速路由与高速通讯；宋克柱（1966-），男，博士，主要研究方向为物理电子学,时移地震采集技术；王砚方（1937-），男，教授、博士生导师，主要研究方向为物理电子学，数据采集与处理。

The Implementation and Research on Pseudo Random Number Generators with FPGA

SHU Li-bao, SONG Ke-zhu, WANG Yan-fang

(Fast Electronics Laboratory, University of Science and Technology of China, Hefei 230027, China)

Abstract: In many projects, it is a great advantage for designing and debugging system to generate pseudo random numbers directly by FPGA. This article presents a simplified and efficient way based on linear feedback shift registers and the specific architecture of FPGA to generate pseudo random numbers. Compared with other methods, such a generator not only has a longer period but also has better statistical characteristics.

Key words: Pseudo random number generators; Linear feedback shift registers; FPGA