一种基于 FPGA 的真随机数发生器设计与实现

张 聪，于忠臣
（北京工业大学 北京市嵌入式系统重点实验室，北京 100124）

摘要： 设计并实现了一种基于 FPGA 的真随机数发生器，利用一对振荡环路之间的相位漂移和抖动以及亚稳态作为随机源，使用线性反馈移位寄存器的输出与原始序列运算作为后续处理。在 Xilinx Virtex-5 平台的测试实验中，探讨了振荡器数量以及采样频率等参数对随机序列的统计特性的影响。测试结果表明本设计产生的随机序列能够通过 DIEHARD 测试，性能满足要求。由于仅使用了普通逻辑单元，使得本设计能够迅速移植到 ASIC 设计，大大缩短了开发周期。

关键词： 真随机数发生器；振荡环；相位漂移与抖动；亚稳态；FPGA
中图分类号： TN782 **文献标识码：** A **文章编号：** 1674-6236(2011)10-0176-04

Design and implementation a FPGA-based true random number generator

ZHANG Cong, YU Zhong-chen
(*Beijing Embedded System Key Lab*, *Beijing University of Technology*, *Beijing* 100124, *China*)

Abstract: A FPGA-based true random number generator (TRNG) is presented in this paper. The design utilizes a pair of oscillators that are permitted to free-run. At some point, the free-running oscillators are coerced to match states via a bi-stable device. Metastability and oscillator drift and jitter are two possible causes of randomness, and a linear feedback shift register (LFSR) is used to as post-processing. In the tests on Xilinx Virtex-5 physical platform, the effects of the design parameters, such as the number of oscillators and sampling frequency, are discussed. The result of DIEHARD suite tests for randomness indicates that the performance of the random sequence of the TRNG meets the requirement. Since the TRNG only uses common logic unites, it can be quickly transplanted to the ASIC design and shorten the development cycle.

Key words: true random number generator; oscillation ring; phase drift and jitter; metastability; FPGA

真随机数发生器(TRNG)在统计学、信息安全等领域有着广泛的应用。在这些领域中，不仅要求数据序列分布均匀、彼此独立，而且要求其具有不可预测性，能够抵御针对随机性的攻击。B. Sunar, W.J. Martin 和 D.R. Stinson 提出[1]，真随机数发生器的性能受 3 个因素的影响：熵源(Entropy Source)，采集方式(Harvesting Mechanism)和后续处理(Post-Processing)。在电路系统中最常见的三种真随机数产生方法为[2]：1)直接放大法：放大电路中的电阻热噪声等物理噪声，通过比较器进行比较后获得随机数序列；2) 振荡采样法：用带有抖动的慢振荡器通过 D 触发器采样一个周期固定的快振荡器，输出随机序列；3)离散时间混沌法：利用混沌电路不可预测以及对初始条件敏感的依赖性的特点产生随机序列。基于模拟电路的结构，熵源的统计分布更加理想，且熵源噪声不随采样周期的变化而改变；基于数字电路的结构，集成度高，便于在 FPGA 等通用可编程平台上实现，但熵源的统计特性与模拟电路相比不够理想[3]。

本文尝试了一种用纯数字电路实现的 TRNG 结构，且不使用诸如 PLL 等特殊资源，便于设计由 FPGA 验证移植到芯片设计。其核心思想是使用反相器和延时单元构成两个相互独立的振荡器，由于内部噪声的差异引起的相位偏移作为熵源，经过一段时间振荡后，随机的状态由数字双稳态电路锁存。多组振荡器的输出，经过异或和同步处理后得到随机序列。该 TRNG 在 FPGA 物理平台上实现并进行了测试验证。

1 TRNG 的设计

1.1 相位漂移与抖动

由于受到电路中噪声的影响，数字电路中时钟信号的周期在每个不同的周期上可能缩短或者加长，这就是时钟抖动。抖动可以用许多方法来衡量和表征，它是一个均值为零的随机变量。振荡器起振时刻的差异和电路元件的工艺偏差，使得振荡器间存在相位漂移。因此抖动信号和相位漂移适合在数字电路中作为 TRNG 的随机源。

1.2 亚稳态

锁存器是有逻辑‘1’和‘0’两个稳定状态的双稳态器件，但是在特殊情况下其可能进入亚稳态[4]，此时它的输出是介

收稿日期：2011-02-24 **稿件编号：**201102063
作者简介：张 聪(1985—)，男，辽宁铁岭人，硕士研究生。研究方向：集成电路数字前端设计。

于‘1’和‘0’之间的中间电平。如图 1 所示锁存器用两个反相器和两个开关表征。当锁存器导通时,采样开关闭合,保持开关打开(图 a);当锁存器关闭时,采样开关打开,保持开关闭合(图 b)。图 c 展示了两个反相器的直流传输特性。当锁存器关闭时 A=B,稳态是 A=B=0 和 A=B=V_{DD},亚稳态为 A=B=V_m,其中 V_m 不是一个合理的逻辑值。因为电平在该点是相互稳定的并且可以无限期停留,所以称该点为亚稳态。但是,任何噪声或者其他干扰都会使得 A 和 B 最终稳定在两个稳态中的一个状态。图 d 非常形象地表征了亚稳态,它就好像处于山顶的小球任何干扰都会使小球滚落到山两端的稳定状态。

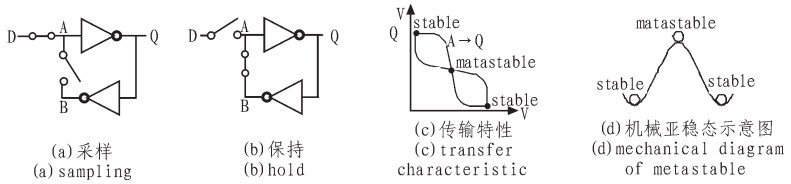

图 1　亚稳态

Fig. 1　Metability

1.3　振荡器的设计

如图 2 所示,二选一复用器既作为延迟单元又作为选通单元使用。当选通信号为‘1’时,形成两个相互独立、自由振荡的环形振荡器。当选通信号为‘0’时,两组反相器交叉相连形成双稳态器件。自由振荡时,两个振荡器之间存在着抖动和相位偏移。在振荡的停止时刻,即振荡环路断开、两组反相器交叉连接时,反相器的瞬时输出电压以及内部噪声的绝对和相对值决定了电路最终稳定在哪个逻辑值上。有时即使反相器跨接在一起,电路也会振荡很长一段时间才能稳定下来,形成亚稳态。综上所述,随机序列的来源用到了抖动和亚稳态两种机制。

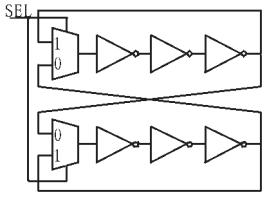

图 2　振荡器结构图

Fig. 2　Oscillator structure

波形如图 3 所示,为了方便数据采集选通信号是由时钟经过分频得到的。在自由振荡阶段,输出信号快速变化不属于任何稳定状态,在图中用斜线表示。在解析阶段,电路是双稳态器件,此时应该保持解析时间足够长,从而使输出电平在大多数情况下稳定在逻辑‘1’或‘0’。

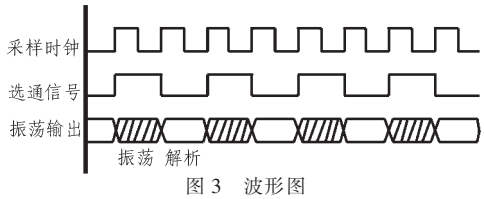

图 3　波形图

Fig. 3　Waveform

1.4　随机源模块的电路设计

各个振荡器的输出经过异或运算可以增加随机性,而亚稳态的传播会造成后续电路的错误动作,因此使用同步器将异或后的随机序列与后续电路隔离开来,同时也方便采集稳定的输出序列做性能分析。此处采用了三级寄存器的同步结构,由 MTBF(Mean Time Between Failure)[5]的定义可知,平均需要经过数百年时间才会发生一次亚稳态通过同步器向下传播的事件,因此是满足设计要求的。该模块电路图如图 4 所示。

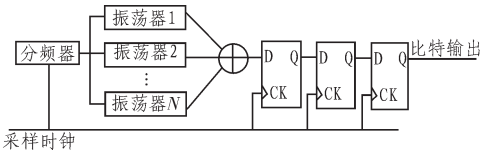

图 4　随机源模块电路

Fig. 4　Circuit of random source

1.5　后续处理模块的设计

理想情况下,D 触发器所采集的信号具有随机的统计特性,可是 FPGA 内部电路不可避免地会受到温度漂移、电压抖动等不良因素影响,从而导致采样得到的随机信号中存在偏置,影响结果的统计特性。所以在采样得到随机序列后要对数据进行消偏处理,使 0 和 1 出现的概率相当。

本设计采用 16 位最大长度二进制伪随机序列(Pseudo Random Binary Sequence)[6] 的输出与采样得到的随机序列进行异或运算作为后续处理,PRBS 产生电路消耗资源少并且使用线性反馈移位寄存器实现,非常适合于在 FPGA 上实现。它的生成多项式是:

$$G_{(16)}=X^{16}+X^{13}+X^{12}+X^{11}+X^{7}+X^{6}+X^{3}+X+1 \quad (1)$$

多项式表示如图 5 所示。

2　TRNG 的 FPGA 实现与测试

整个 TRNG 的实验环境由外部时钟源、FPGA 开发板以及逻辑分析仪组成。TRNG 采用 Xilinx 公司的 Virtex-5 系列中的 XC5VLX110 作为物理实现平台,外部时钟频率为 64 MHz。由 FPGA 产生的随机数据,经逻辑分析仪采集后,使用

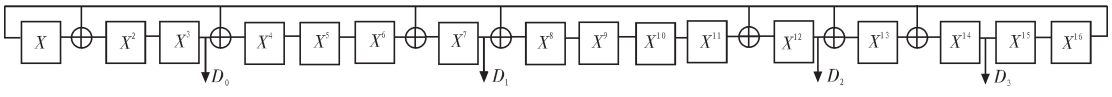

图 5　PRBS 的多项式表示

Fig. 5　Polynomial representation of PRBS

DIEHARD battery of tests of randomness[7]随机数测试程序进行测试,检验随机序列的性能。

DIEHARD 测试是由 16 项测试组成的用来度量随机数发生器性能的一组统计学测试，它由 George Marsaglia 开发并于 1995 年首次发布。DIEHARD 的测试结果叫做 *P*-value，它由方程 *P*-value=$F_i(X)$计算得到,其中 F_i 试图建立样本 X 在 0 和 1 间服从均匀分布的分布函数。因为 F_i 是渐进逼近的，它在尾部的近似效果变差，所以数值接近 0 或 1 的 *P*-value 在真随机序列中极少出现。当被测序列随机性能很差时,会有很多 *P*-value 的值是精确到小数点后数位的 0 或者 1,例如 1.000 000。需要强调的是,*P*-value 等于 1.000 000 或 0.000 000 是序列为真随机序列的充分不必要条件。

2.1 FPGA 位置约束

为保证每个振荡器中的两个独立振荡环的理论振荡周期相同,以便更容易在锁定期间产生亚稳态,加大噪声对输出电平的影响,同时尽量让各个振荡器的输出在进行异或运算前延迟不出现太大偏差。所以对 TRNG 中的振荡环进行位置约束：将振荡环中的反相器约束在左右相邻的逻辑单元(Slice)中,让各个振荡环分别约束在上下相邻的逻辑单元中。

2.2 振荡器数目对统计特性的影响

在 32 MHz 的采样频率下，分别以 15 组、19 组、27 组和 32 组振荡器作为 TRNG 的随机源,随机序列经过同步器后不与 PRBS 运算直接输出。将采集到的随机序列送入测试程序进行测试以后,其结果如表 1 所示。

表 1 不同振荡器数目的测试结果

Tab. 1 Test result of different number of oscillator

Test	*P*–value			
	N = 15	N = 19	N = 27	N = 39
Birthday Spacing	0.728 235	0.442 170	0.230 892	0.582 145
Overlapping 5-permutation	0.723 314	0.573 830	0.150 224	0.653 355
Binary Rank 31×31	0.660 371	0.337 681	0.577 720	0.835 246
Binary Rank 32×32	0.935 259	0.411 223	0.978 785	0.963 681
Binary Rank 6×8	1.000 000	1.000 000	1.000 000	0.813 708
Bit stream	1.000 000	1.000 000	0.890 200	0.580 210
OPSO	1.000 000	1.000 000	0.989 800	0.594 100
OQSO	1.000 000	1.000 000	0.988 200	0.521 200
DNA	1.000 000	1.000 000	0.970 000	0.638 700
Stream of bytes Count-The-1	1.000 000	1.000 000	1.000 000	0.964 383
Specific bytes Conut-The-1	1.000 000	1.000 000	1.000 000	0.746 921
Parking Lot	0.359 118	0.644 878	0.666 851	0.828 668
Minimum Distance	0.410 437	0.375 076	0.149 517	0.743 388
3D Spheres	0.654 918	0.174 606	0.744 797	0.069 999
Squeeze	1.000 000	1.000 000	0.998 682	0.789 090
Overlapping Sums	0.326 859	0.686 135	0.109 606	0.195 026
Runs up	0.868 715	0.352 910	0.200 281	0.319 496
Runs down	0.963 565	0.598 274	0.540 114	0.146 326
Craps	0.922 008	0.626 522	0.407 112	0.450 501

可以看出,振荡器的数目直接影响随机源模块产生序列的统计性能，振荡器数目越多,TRNG 输出序列的随机性越好。但是如果振荡器的数目太多,会消耗过多的硬件资源,功耗也过大。因此,不宜通过单纯地增加振荡器数目的方法提高随机序列的性能。

定性分析如下:将序列的每位看作是一个随机的二进制变量 X,定义 b 是序列的偏置。即

$$b=|P(X=1)-1/2|=|P(X=0)-1/2| \tag{2}$$

根据 Piling-up 引理[8],输出序列的偏置是:

$$b=2^{n-1}\prod_{i=1}^{n}b_i \tag{3}$$

其中 n 是输入序列的个数,b_i 是每个序列的偏置。容易看出 $b\leqslant b_i(1\leqslant i\leqslant n)$,等式当且仅当在 $b_i=0(\forall i=1,2,\cdots,n)$或者 $b_j=1/2(\forall j\neq i)$时成立。简而言之,异或运算显著地减小了独立输入序列的偏置。假设 n=16 且所有 b_i=1/3,那么 b=0.000 761 可以忽略不计。

2.3 后续处理模块对统计性能的改善

由上一节的分析可知，增加振荡器数量是改善序列统计特性的有效方法。但前提条件是各个振荡器相互独立,当振荡器数量过多时位置约束很可能与相互独立的要求相互矛盾。因为高速的振荡信号往往发生相互串扰的情况，并且消耗更多资源和功耗，所以有必要在保证 TRNG 包含一定数量的振荡器的前提下,引入后续处理模块。从而达到消耗资源较少,序列性能较好的目的。

本项测试以 19 组振荡器作为 TRNG 的随机源,输出序列与 PRBS 模块输出进行异或运算，然后分别使用 32,16,8,2 MHz 的采样时钟采集数据。将数据送入测试程序进行测试以后,结果如表 2 所示。

表 2 不同采样频率的测试结果

Tab. 2 Test result of different sampling frequency

Test	*P*–value			
	f_s=32 MHz	f_s=16 MHz	f_s=8 MHz	f_s=2 MHz
Birthday Spacing	0.307 091	0.479 600	0.360 894	0.520 010
Overlapping 5-permutation	0.138 627	0.912 908	0.460 020	0.159 915
Binary Rank 31×31	0.998 594	0.702 277	0.689 345	0.366 537
Binary Rank 32×32	0.478 827	0.495 350	0.638 612	0.340 895
Binary Rank 6×8	0.832 741	0.833 368	0.983 740	0.140 908
Bit stream	0.554 520	0.673 710	0.319 870	0.125 970
OPSO	0.299 700	0.377 700	0.601 600	0.688 600
OQSO	0.193 400	0.386 200	0.377 100	0.674 800
DNA	0.387 000	0.259 800	0.562 900	0.216 000
Stream of bytes Count-The-1	0.549 834	0.620 547	0.366 430	0.262 596
Specific bytes Conut-The-1	0.408 254	0.665 905	0.467 708	0.673 911
Parking Lot	0.485 422	0.857 314	0.436 712	0.880 404
Minimum Distance	0.185 601	0.113 618	0.598 913	0.814 288
3D Spheres	0.336 313	0.433 047	0.022 821	0.564 009
Squeeze	0.846 001	0.482 508	0.290 779	0.947 326
Overlapping Sums	0.175 234	0.216 132	0.559 449	0.590 358
Runs up	0.338 215	0.079 644	0.466 560	0.403 556
Runs down	0.291 810	0.697 394	0.758 464	0.801 223
Craps	0.817 611	0.611 785	0.312 742	0.375 999

可以看出,TRNG生成的随机序列全部达到了预定的性能指标。同时不难发现,采样时钟频率对TRNG的输出统计特性是有影响的,当采样频率逐渐降低时,TRNG的随机性能逐步提高。出现这种现象是由于采样频率越高,就与振荡频率越接近,二者的相位偏移干扰了随机信号的获取影响了统计特性[9]。

3　结束语

本文尝试了一种纯数字形式的真随机数发生器结构,规模较小、易于移植。电路包含两个振荡环,分为自由振荡和锁存至双稳态两个工作状态。利用振荡环之间的相位偏移和抖动以及双稳态器件的亚稳态作为随机源。本文探讨了振荡器数量对序列统计特性的影响,并在加入后续处理模块的情况下试验了多种采样频率,经测试随机序列完全符合预定指标。

参考文献:

[1] Sunar B, Martin W J, Stinson D R. A provably secure true random number generator with build-in tolerance to active attacks [J].IEEE Transactions on Computer, 2007, 56(1): 234–236.

[2] 吴燕雯,戎蒙恬,诸悦,等.一种基于噪声的真随机数发生器的ASIC设计与实现[J].微电子学, 2005, 35(2): 213–216.
WU Yan-wen, RONG Meng-tian, ZHU Yue, et al. Design and implementation of a noise-based true random number generator ASIC[J]. Microelectronics, 2005, 35(2):213–216.

[3] 张润捷.一种基于FPGA实现的真随机数发生器[J].中国集成电路, 2008, 2(114):52–55.
ZHANG Run-jie. FPGA-based true random number generator [J]. China Integrated Circuit, 2008, 2(114):52–55.

[4] EWister N H, Harris D. CMOS大规模集成电路设计[M].北京:机械工业出版社, 2005.

[5] Kaeslin H. Digital integrated circuit design [M].Cambridge: Cambridge University Press, 2008.

[6] Proakis J G. Digital communications[M].北京:电子工业出版社, 2006.

[7] Marsaglia G. DIEHARD: A Battery of Tests of Randomness [EB/OL]. (1996).http://stat.fsu.edu/~geo.

[8] Epstein M, Hars L, Krasinski R, et al. Design and implementation of a true random number generator based on digital circuit artifacts[EB/OL]. (2003)[2011–02–01].http://citeseerx.ist.psu.edu/viewdoc.

[9] 霍文捷,刘政林,陈毅成,等.一种基于FPGA的真随机数发生器的设计[J].华中科技大学学报, 2009,37(1):73–76.
HUO Wen-jie, LIU Zheng-lin, CHEN Yi-cheng, et al. Design of a true random number generator using FPGA[J]. Journal of Huazhong University of Science and Technology, 2009, 37(1):73–76.

(上接第175页)

现。首先结合IIR滤波器的基本结构,针对分布式算法中查找表规模过大的缺点,采用级联或并联结构,利用多块查找表使得硬件规模极大地减小,提出了并行和串行相结合的设计方案,然后在Quartus Ⅱ软件平台上,对设计的滤波器进行了仿真验证,然后对Matlab理论值和仿真值进行了比较分析,验证了设计的IIR滤波器的正确性。最后还做了硬件测试,测试结果表明,本文所设计的滤波器硬件规模较小,系统最高时钟频率达到了80 MHz以上,体现了设计的实时性。同时,只要将查找表进行相应的改动,就能分别实现低通、高通、带通IIR滤波器,体现了设计的灵活性。

参考文献:

[1] 胡广书.数字信号处理——理论、算法与实现[M].北京:清华大学出版社,2003.

[2] 赵红怡,张常年.数字信号处理及其MATLAB实现[M].北京:化学工业出版社,2002.

[3] 邹彦,庄严.EDA技术与数字系统设计[M].北京:电子工业出版社,2008.

[4] 黄晓红,蔡江利.基于FPGA的改进型FIR滤波器的实现[J].电子技术应用,2009(5):32–34.
HUANG Xiao-hong,CAI Jiang-li.Design of improved FIR filter based on FPGA[J]. Application of Electronic Technique, 2009(5):32–34.

[5] 屈星,唐宁,等.基于FPGA的IIR数字滤波器的设计与仿真[J].计算机仿真,2009,26(8):304–307.
QU Xing,TANG Ning et al.Design of IIR digital filter based on FPGA[J].Computer Simulate,2009,26(8):304–307.

[6] 魏灵,杨日杰,等.基于分布式算法的数字滤波器设计[J].仪器仪表学报,2008,29(10):2100–2104.
WEI Ling,YANG Ri-jie,et al. Design of FIR filter based on distributed arithmetic and its FPGA implementation[J]. Chinese Journal of Scientific Instrument,2008,29 (10): 2100–2104.