高斯白噪声发生器的 FPGA 实现

冯建群，文海明

（江西工业工程职业技术学院，江西 萍乡 337055）

摘 要：分析了几种高斯白噪声发生器的实现途径，提出了一个基于 Box-Muller 算法级联中心极限定理实现高精度高斯白噪声发生器的方法，在 FPGA 中使用 HDL 语言实现。与现有的实现方法相比，本设计速度更快、占用的资源更少，且易于 FPGA 实现，可作为一个快速实用的误码率测试平台的一部分。

关键词：高斯白噪声发生器；FPGA；Box-Muller；CLT

中图分类号：TP391.9 **文献标识码**：B **文章编号**：1674-7720(2012)21-0073-03

FPGA implementation of white Gaussian noise generator

Feng Jianqun, Wen Haiming

(Jiangxi Vocational Technical College of Industrial Engineering, Pingxiang 337055, China)

Abstract: This paper analyses several methods of implementation white Gaussian noise generator (WGNG), then presents a method for designing a high accuracy WGNG suitable for digital communication channel emulation. The proposed solution is based on the Box-Muller algorithm in conjunction with the central limit theorem. Compaed with others methods, this architecture is easy for a digital implementation in FPGA. The proposed present model can be as a component of fast and practical emulation platform for BER simulation.

Key words: WGNG; FPGA; Box-Muller; CLT

加性白高斯噪声（AWGN）通信信道是通信系统设计与测试的主要信道模型之一。其数学模型如图 1 所示[1]，信号 $s(t)$ 加上白高斯噪声 $n(t)$ 得到了加噪后的接收信号 $r(t)$。

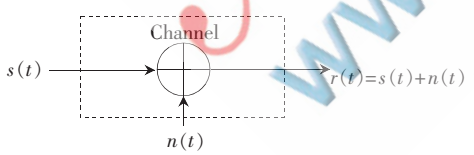

图 1 AWGN 通信信道模型

本文使用通用硬件描述语言（HDL）设计了一个高斯白噪声发生器（WGNG），可以作为一个功能模块集成到基于 FPGA 的数字通信系统设计中，便于在 FPGA 中进行信道编译码的仿真测试工作。

1 方法概述

高斯白噪声产生的现有算法主要包括：中心极限定理 CLT（Central Limit Theorem），Box-Muller 算法、混合方法（Box-Muller+CLT），基于 CA（Cellular Automata）的方法和 Wallace 方法等。

1.1 CLT 方法

CLT 方法的理论根据是中心极限定理，其表述如下：

若 $X_i(i=0\cdots N-1)$ 为 N 个统计独立、同分布随机变量，均值为 m_x，标准差为 σ_x。则定义随机变量 X_N 为：

$$X_N=\frac{1}{\sigma_x\sqrt{N}}\sum_{i=0}^{N-1}(X_i-m_x) \quad (1)$$

当 $N\to\infty$ 时，X_N 服从标准正态分布 $N(0,1)$。对于均值为 0，标准差为 1，则式（1）变为：

$$X_N=\frac{X_1+X_2\cdots+X_N}{\sqrt{N}} \quad (2)$$

CLT 方法通常使用累加器来实现，导致速度慢、输出率低，不合适于高速的应用，如参考文献[2]，最高时钟速率为 98 MHz，而最高输出速率只有 24.5 MHz。可以采用流水线方式提高速率，不过代价是要用大量的寄存器和加法器。

1.2 Box-Muller 算法

Box-Muller 被广泛使用在产生加性高斯白噪声中，其算法步骤如下：

（1）产生两个独立的服从 U(0,1) 均匀分布的随机变量 u_1、u_2

（2）计算 $f(u_1)=\sqrt{-2\ln(u_1)}$ （3）

$g(u_2)=\sin(2\pi u_2)$

（或 $g(u_2)=\cos(2\pi u_2)$） （4）

（3）产生服从高斯分布的随机变量 $x=f(u_1)g(u_2)$

在这个算法中，u_1、u_2 的随机性是保证频谱是“白”的，而 Box-Muller 方法保证 x 的统计特性是高斯分布。

1.3 混合方法

混合方法是 Box-Muller 方法和 CLT 方法的结合。先用 Box-Muller 方法实现高斯白噪声，再用 CLT 方法提高输出的精度。如果 CLT 采用流水线方式实现，可以保证高速率输出。混合方法具有非常好的统计特性，输出精度很高，是目前产生高斯白噪声最好的方法。本文采用混合方法来实现高斯白噪声发生器。图 2 所示为混合方法的 Matlab 仿真结果。

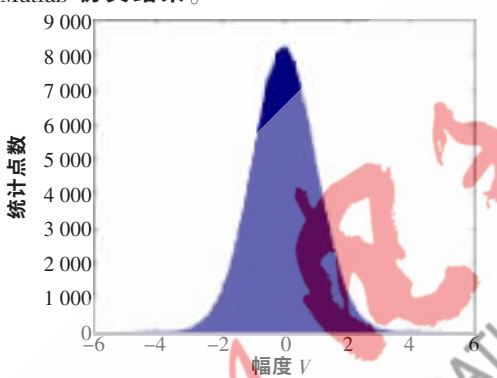

图 2 混合方法统计特性仿真

2 混合方法硬件实现架构

图 3 为本文所使用的混合方法的硬件实现总体结构图。它由 Tausworthe 地址产生模块、Sine/Cosine 查找表、Sqrt_log 模块、2 个乘法器和 2 个 CLT 模块组成。

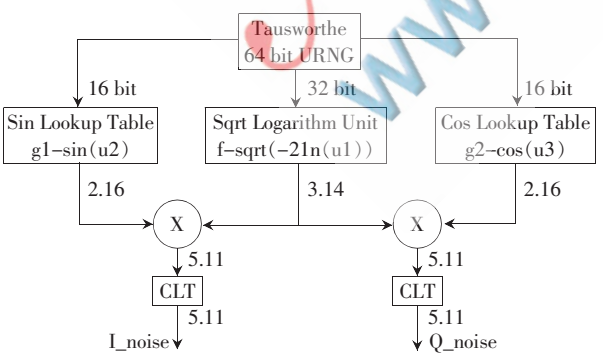

图 3 混合方法总体结构图

由于在 FPGA 中要有效地实现平方根、Sine/Cosine 和 log 等函数，将需要很多硬件资源。为了减少硬件的复杂度，采用量化 Box-Muller 算法，即查找表 Box-Muller 算法，先计算式（1）和式（2）函数值，并存放在 ROM 中，需要计算时用 u_1、u_2 作为地址查找相应函数值。这样可以大量减少运算量，当然这样需要额外的存储器。

2.1 产生服从 $U(0,1)$ 的随机变量

为了实现混合方法，首先要产生服从均匀分布 $U(0,1)$ 的随机变量。而用数字方法实现随机变量十分困难，但可以实现“伪随机变量”，所谓“伪随机变量”就是输出数据的周期非常长。产生“伪随机变量”有多种方法，但很多方法硬件实现比较困难，因为它们要求硬件资源较多。针对 FPGA 实现常用两种方法：LFSR 和 Tausworthe[3]。

LFSR 是利用移位器来实现的，它的周期为 2^m-1，其中 m 为移位器的位数，每个时钟输出一个 bit。假设要求输出为 64 bit，周期为 $2^{200}-1$，则要用 $200\times64=12\ 800$ 个触发器。这样如果周期要求越长，花的硬件资源也越多。

Tausworthe 是对 LFSR 进行改进，这个算法的优点是有比较好的统计特性，在 FPGA 实现中，只要用“异或”运算，硬件实现比较容易，占用资源少。用 C 语言实现产生 64 bit，周期约为 2^{258} 的伪随机变量的 Tausworthe 算法如下：

```
unsigned long long z1, z2, z3, z4, z5;
double lfsr258( )
{/* Generates numbers between 0 and 1. */
    unsigned long long b;
    b=(((z1<<1)^z1)>>53);
    z1=(((z1&0xFFFFFFFFFFFFFFFE)<<10)^b);
    b=(((z2<<24)^z2)>>50);
    z2=(((z1&0xFFFFFFFFFFFFFE00)<<5)^b);
    b=(((z2<<3)^z3)>>23);
    z3=(((z3&0xFFFFFFFFFFFFF000)<<29)^b);
    b=(((z3<<5)^z4)>>24);
    z4=(((z4&0xFFFFFFFFFFFE0000)<<23)^b);
    b=(((z4<<3)^z5)>>33);
    z5=(((z5&0xFFFFFFFFFF800000)<<8)^b);
    return(z1^z2^z3^z4^z5);
}
```

上述算法采用 C 语言，实际用 FPGA 实现只需要“异或”运算很容易实现。该算法的周期为 2^{258}，假设系统的工作频率为 10 GHz，那么 $1.468\ 7\times10^{60}$ 年后才会被重复，所以可以看成随机了。

2.2 实现 $\sqrt{-2\ln(x)}$

$\sqrt{-2\ln(x)}$ 是一个复合函数，复合函数基本上都可以分解成几个基本函数分级计算。但是这种方法对于用硬件来实现不仅会带来很大的延时和舍入误差，而且可能会让计算无法进行。图 4 绘出了该函数图形，从图中可以看出在 0 和 1 附近函数呈现为极大的非线性，所以对于本函数的计算也不能采用传统的基于均匀地址的查找表方法，因为对于高度非线性区域可能要使用非常小的分段才能达到线性区域的精度。在本文中采用了分级分段方法[3]（HSM），可以使用更少的存储资源达到同样的精度。

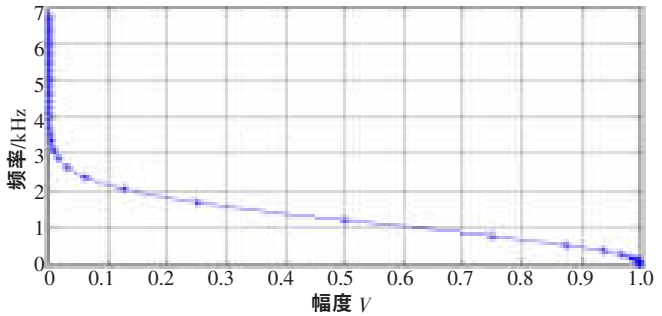

图 4 $\sqrt{-2\ln(x)}$ 在区间(0,1]绘图及分段

具体做法是，把 (0,1] 分成两个区间：(0,0.5)和[0.5,1]。对 (0,0.5)，用它的中点来分段，然后对(0,0.25)再用它的中点来分段。对[0.5,1]做同样的处理，得到的分段结果如图 4 所示。这样在每个区间，可以用一条直线 $y=ax+b$ 逼近，即每个区间用 2 个系数 a、b 来代替，可以用查找表的方式来实现，查找表的长度是输入变量 x 的位数的一倍，而表中每个单元是 a、b 需要的位数之和。x 的位数为 32，所以表长为 64 bit，从仿真结果可计算 a 需要 33 位来表示，b 需要 31 位来表示，数据格式函数实现框图如图 5 所示。

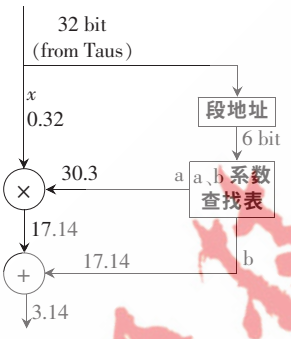

图 5 $\sqrt{-2\ln(x)}$ 实现框图

图中有两个重要的小模块：段地址模块和 a、b 系数查找表模块。对于段地址模块在参考文献[4]中给出的寻址方法需要更多资源，因为它涉及到 32 个 1 比特相加，速率非常低，并要用很多加法器，对 32 bit 的输入至少要 4 级流水线。而采用直接寻址，并行计算[5]，这样不用加法器，只需要 2 级流水线。

对于 a、b 系数查找表模块，要进行多次计算、仿真，检查结果的正确性和精度，以此决定每小区间的系数 a、b。然后把它们存入 ROM，寻址后直接读出。图 6 为该函数的 ModelSim 仿真图。

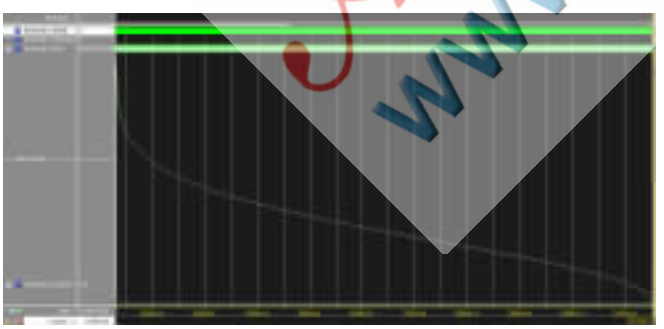

图 6 $\sqrt{-2\ln(x)}$ 函数 ModelSim 仿真

2.3 Sine/Cosine 的计算

Sine/Cosine 的计算通过查找表实现，将 Tausworthe 模块产生的 16 位数据作为地址寻址，输出结果用 18 位表示。

2.4 CLT 模块

由 CLT 算法可知，由于要用到累加器，如果不采用流水操作就会使数据的输出速度降为原来的 $1/N$。在本模块中，使用了流水作业，使得数据的输出速度能够保持不变。图 7 所示为当 N 等于 4 时的 CLT 算法实现图，只要把输出的数据右移一位就实现了除 $\sqrt{4}$ 即 2，用 FPGA 实现起来很方便。

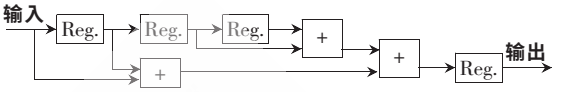

图 7 CLT 模块图(N=4)

3 结果分析

3.1 统计特性

通过 Matlab 对产生的噪声进行分析，把 ModelSim 仿真产生的噪声输出到一个文本文件中。图 8 为对一百万个噪声样点统计结果，从图中可以看出统计结果接近理想效果。图 9 为对两个连续的 10 000 个点进行的相关性测试，可以看出这些点没有明显的相关性。

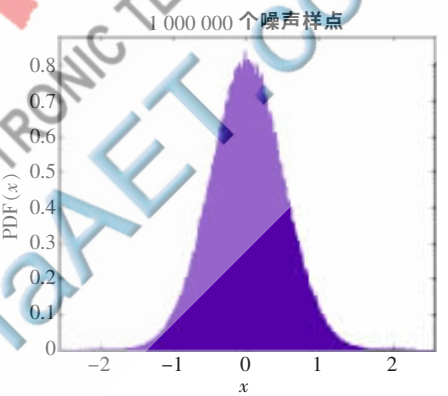

图 8 1 000 000 个噪声样点的 PDF 统计

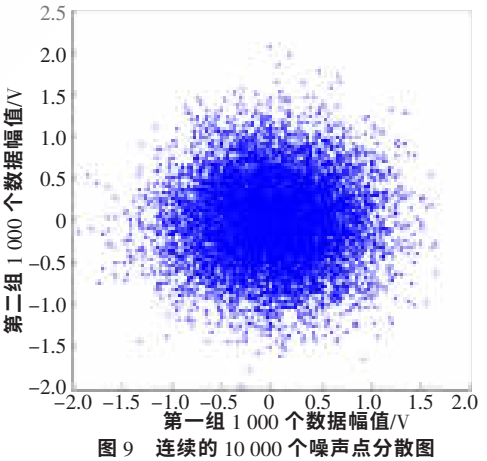

图 9 连续的 10 000 个噪声点分散图

3.2 综合结果及比较

本设计在 Xilinx 公司 ISE9.1 开发环境下完成，其中综合工具使用 Synplify Pro 8.1，器件使用 XC2V4000-6，占用了 1%的 SLICEs，资源利用情况和达到的性能如表 1 所示。

表 1 综合结果及比较

	SLICEs	Block RAMs	MULT 18X18s	Clock/MHz
Xilinx[4]	653	4	8	168
Lee[5]	770	6	4	155
本设计	710	3	2	228

从表 1 可以看出本设计和其他两个设计相比，在使用更少的设计资源的情况下获得了更快的速度。

本文中实现的 WGNG，只占用了 XC2V4000-6 中的 1% 的 SLICEs，2% 的 Block RAMs，1% 的 18×18 乘法器，最高时钟速度可达 228 MHz。且可以同时产生两路独立高斯噪声，特别适合于 I/Q 两路信号的仿真。已用于某项目 LDPC 编解码性能测试，测试结果和预期的结果相符。使用 WGNG 产生的加性白高斯噪声，通过滤波和适当数学处理可以得到更复杂的信道模型（如 Rayleigh 信道）。

参考文献

[1] PROAKIS J G. Digital communications [M]. McGraw-Hill High Education, Electrical and Computer Engineering Series, 2001.

[2] GHAZEL A. Design and performance analysis of a high speed AWGN communication channel emulator [C]. IEEE PACRIM Conference, Victoria, B. C., Aug. 2001:374-377.

[3] LEE D. Non-uniform segmentation for hardware function evaluation[C]. In Proc. Int'l Conference on Field Programmable Logic and its Applications, LNCS 2778, Springer-Verlag, Lisbon, Portugal, Sep 2003: 796-807.

[4] Xilinx Data Sheet. Additive White Gaussian Noise (AWGN) Core v1.0. Xilinx[R]. Inc. October 2002.

[5] LEE D, LUK W. A hardware gaussian noise generator using the wallace method [J]. IEEE Transaction VLSI Systems, 2005, 13(8):911-920.

（收稿日期：2012-08-15）

作者简介：

冯建群，男，1968 年生，硕士，副教授，主要研究方向：嵌入式系统开发。

文海明，男，1980 年生，讲师，硕士研究生，主要研究方向：嵌入式系统开发。