Additive White Gaussian Noise using box muller method

June 29, 2016 **Product Specification**

Features

- Designed for Virtex™-II and Virtex-II Pro™ using structural Verilog.
- Probability density function (PDF) deviates less than 0.2 percent from the Gaussian PDF for |x| < 4.8 and is obtained from a closed-form expression
- Based on the Box-Muller algorithm and the central limit theorem.
- Period of generated noise sequence is ~ 2^{190} = 1.57×10^{57} samples
- Power spectral density is flat.
- SNR input ranges from 0.0 to 15.9 dB in steps of 0.1 dB and provides scaling to obtain desired variance
- Noise is quantized to 16 bits with 5 bits of integer and 11 bits of fraction
- 760-bit internal seed selectable through top-level generics
- Core returns to its initial state upon reset
- Bit-true Simulink model and MATLAB programs included
- Uses relationally placed macro (RPM) mapping and placement technology, for maximum and predictable performance
- Requires 480 slices (40 rows, 12 columns), five block RAMs and five hardware multipliers
- Maximum clock rate and output sample rate of 245 MHz in Virtex-II
- Maximum clock rate and output sample rate of 300 MHz in Virtex-II.

Facts

Core Specifics				
Supported Device Family	Virtex-II, Virtex-II Pro			
Resources Used	I/O	LUTs	FFs	Block RAM
	28	733	902	5
Special Features	RPM Core			

Provided with Core	
Documentation	Product Specification
Design File Formats	Verilog HDL
Constraints File	UCF
Verification	MATLAB + ModelSim + Hardware
Instantiation Template	VerilogWrapper
Reference Designs & application notes	BER Measurement of Uncoded BPSK in Hardware
Additional Items	Bit-True Simulink Model; MATLAB Analysis programs

Design Tool Requirements	
Xilinx Implementation Tools	ISE 4.2.03i or later
Verification	MATLAB R12
Simulation	ModelSim 10.4b
Synthesis	Synplify Pro 7.1

Functional Description

Gaussian noise sample is generated: quantized version of box muller method is performed to obtain good approximation of the Gaussian distribution. The aim is to smooth the fluctuation of the distribution obtained with the quantized box muller method. Box muller variable is generated a truncation that can be done according to the needed accuracy to keep only b bits after the decimal point.When the sign bit is one, one's complement is used to get –ve values.

Mean value is -2^-b-1. 1. Turbo code and low density parity check. It generates 16 bit noise samples accurate to one unit in the last place ((ulp_) up to 8.2σ which models the true gaussian pdf for a simulation size of over the 10^15 samples. Actually one need 100 to 1000 bits in the error. Proposed design flow and how we evaluate elementary functions associated with box muller method. Matlab and c model of noise generator. These softare are programmed to be bit accurate. Analog Truly random but sensitive to changes, low throughput. Digital : based on the transformation of uniform random

variables. Popular methods include 1. Box muller method transforms two uniformly distributed random variables into two normally distributed variables through a series elementary function evaluations. 2. Central limit method overcomes approximation errors. Method inlcudes highly accurate elementary function evaluation technique eliminating the need for the central limit theorem altogether.1. Evaluated 2. Noise quality in the tails. This resulted in an output rate of one samples per clock. Ϭ=6.7 1. Slices 2. Block ram 3. Dsp slice (multiply and add unit) A logic cell is 2.25 logic cells :comprised of a 4 input lookup table , 16*1 ram, 16 bit shift register, mux and register. A slice contains additional resources : multiplexers and carry logic. A slice is counted as being equivalent to 2.25 logic cells. Each block ram can store 18kb of data and a dsp slice can perform 18 bit by 18 bit multiplication followed by 48 bit addition. Design flow: periodicity of samples. Polynomial co efficient table using matlab. Matlab symbolic toolbox. Matlab and c models of the noise generator. U0 and u1, uniforma random number variables and x0 and x1 : two samples.

Here, toursworthe urng are used. Periodicity of 10^15 and 16 bit noise samples. Urng should have at least period of 10^15. Normal distribution using maple : we observe that we need to be able to represent up to 8.1σ for a population of 10^15 samples. By examining σvalue is determined by the smallest value of f, which in turn is determined by the E_b/N_0 of 0 dB for rate ½ QPSK is as simple as performing a left-shift of the noise by one bit and setting the SNR equal to 3 dB.

smallest value of u0. 48 bits for u0, maximum σ=8.26 bits for u1, Ulp : 2^-11. Table maker's dilemma and large area penalty.We opt for faithful rounding in this work. Maximum absolute error compared against infinite precision should be less than or equal to 2^-11. Bit width : bx Integer bit width ibx. Fraction bit width fbx. Maple minimizes the maximum error. Main challenge is : mini bit bit width optimization. A technique for optimizing fixed point signals using analytical errors expressions with a guaranteed maximum error bound. Quantize the signals : truncation and round to nearest. Minibit error expression : 128 bits for one polynomial and 16 segments for degree two polynomial.

The AWGN core generates white Gaussian noise using a combination of the Box-Muller algorithm and the central limit theorem, following the general approach. The Box-Muller algorithm generates a unit normal random variable via a transformation of two independent random variables that are uniformly distributed over [0,1]. The uni-form random variables internal to the core are generated using multiple-bit leap-forward Tausworthe. The outputs of the Tausworthes are used to index the contents of ROMs that store the function values used in the Box-Muller algorithm. The outputs of multiple parallel Box-Muller designs are then averaged to obtain a PDF that is Gaussian to within 0.2% out to 4.8. Finally, scaling is performed based on the value of the SNR input to achieve the desired noise variance.

Design Verification

Design Verification is was done by Modelsim 10.4b creating testbench of awgn verilog file in Verilog HDL. The testbench consists of AWGN core, a bit counter.