Evaluation Project for Design engineers for hardware implementation of algorithms.

Please design a RTL module for GNG/AWGN by following the reference

R. Gutierrez, V. Torres, J.Valls, "Hardware Architecture of a Gaussian Noise Generator Based on Inversion Method, " IEEE Transactions on Circuits & Systems II, Vol.59, No.8, pp.501-505, 2012

http://ieeexplore.ieee.org/document/6236107/

A copy is included as awgn_inversion_method_06236107.pdf

Read the paper thoroughly with emphasis on being able to implement the MATLAB and RTL code for the proposed system in Fig 1. URNG bitwidth is 64-bit and AWGN output is 16-bits.

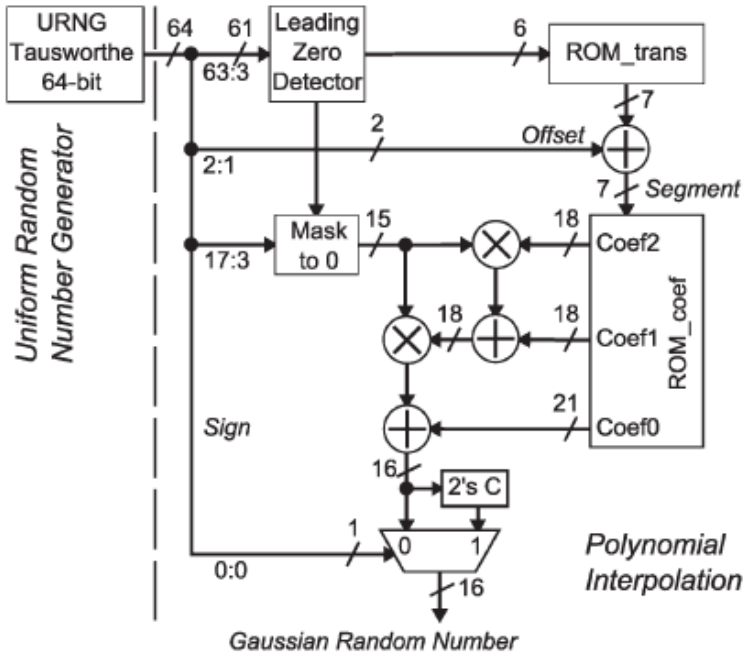

Fig. 1. Proposed 64-bit GNG architecture. Signal widths are shown for a design that uses a single Tausworthe and reaches 9.1 σ.

Deliverables

1) You are expected to understand the paper and write program on arriving at ROM coefficients that are needed. Use the MATLAB preferably without the toolboxes. For fixedpont use round, max, min commands.
2) Bit accurate quantized MATLAB Model for AWGN based on the provided reference. Your MATLAB model should be able to generate the test vectors needed for automated verification of your Verilog RTL design. Also you should check your "golden" model using the following:
   You can use the kstest command to test the normality of the distribution generated from your MATLAB model. Please see http://www.mathworks.com/help/stats/kstest.html#btnf4t4-2

3) Synthesizable Verilog RTL design for AWGN. The RTL IP core should have the following input interface pins: clock (1 bit), reset (1 bit). The output interface pins should be awgn_out (16 bits). The design needs to have URNG as shown in Fig 1. The URNG can be initialized with a 64-bit seed that can be defined as a constant in RTL.
4) Make sure to have a Verilog/System Verilog test bench to compare the RTL results and MATLAB results. You can choose to generate all the vectors from MATLAB model for 10,000 samples and then run a RTL test bench with RTL model to read the test vectors and compare the intermediate values from your RTL model.
   Provide well commented code along with a Model Sim project to simulate the RTL code and read the test vectors generated by MATLAB model and do automatic comparison of key intermediate results. Your simulation should be able to give warnings on any bit-level errors. Functional simulation is required while you are not required to test your design on an FPGA.
5) Perform FPGA(Altera/Xilinx) design synthesis. For the design synthesis, you can choose any latest FPGA device and you need to set the constraints on clock, inputs and outputs.
6) Documentation: Steps to recreate your results for MATLAB simulation, RTL simulation and synthesis.
   Prepare a summary of your design capturing the area, clock frequency and throughput.
   Show a simple snapshot of RTL waves that has reference data and RTL data in the same waveform.
   Show the kstest command output as well as histogram output for the MATLAB model.
7) Please track the time you are spending on this project based on the above tasks 1 to 6 on daily basis in a simple spreadsheet. It is important that you do this accurately so that you and the company understands your strengths in a reliable fashion.
8) [Optional] You may want to add a GNU license per http://www.gnu.org/licenses/gpl-3.0.en.html, if you would like your code to be used by others for free**. The company does not plan to use your software for any purposes other than for estimation of your skill set as part of the interview process. You will not be paid for any of your time spent on this evaluation.**

Tools you need:

1) Modelsim https://www.mentor.com/products/fv/resources/overview/modelsim-pe-evaluation-software-21-day-license--75279295-fcb3-47b4-ab12-7b129bc4cb39
2) MATLAB https://www.mathworks.com/programs/trials/trial_request.html?
3) Vivado Webpack https://www.xilinx.com/products/design-tools/vivado/vivado-webpack.html OR Quartus webpack http://dl.altera.com/13.1/?edition=web

Rules of Engagement

Please do not ask your friends or professors to help you in the coding. Please use your own code. It is OK to ask to search or ask around if you have conceptual questions.

Communication of Developed Package

Please upload your developed package via a github. Create a public repository of your code and documentation at https://github.com and **submit the link by email within 15 days after receiving the problem. If you need extension because you do not have time to work on this problem, please request in-advance.** If you are familiar with RTL design, MATLAB, fixedpoint modeling, hardware architecture, verification aspects, you should be able to complete this in 3 to 4 days.