Evaluation Project for ASIC/FPGA Design Engineer

Osso Vahabzadeh
TexasLDPC Inc.

Please design an AWGN IP core by following the reference

Lee, D.-U.; Villasenor, J.D.; Luk, W.; Leong, P.H.W., "A hardware Gaussian noise generator using the Box-Muller method and its error analysis," in Computers, IEEE Transactions on , vol.55, no.6, pp.659-671, June 2006, doi: 10.1109/TC.2006.81

http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1628955

Read the paper thoroughly with emphasis on being able to implement the Matlab and RTL code for pseudo-code given in Fig 6. A simple block diagram is given in Fig 2.

Deliverables

1) You are expected to write program on these aspects on arriving at specification, bitwidths and error analysis.
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 (Links to an external site.)
3) Synthesizable Verilog RTL design for AWGN. The RTL IP core should have the following input interface pins
   Clock (1 bit), reset (1 bit), urng_seed1/2/3/4/5/6 (each seed is 32-bit). The output interface pins should be awgn_out (16 bits). urng_seed1/2/3/4/5/6 supplied as seeds to two URNGs used at the initialization stage.
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 an 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) Prepare a datasheet of your design. Use this as an example:
   http://www.xilinx.com/support/documentation/ip_documentation/awgn.pdf
6) Please upload your developed program via a github. Create a public repository of your code and documentation at https://github.com and **submit the link by email by June 17th 2016 (5pm central time).**

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 according to 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.**

Rules of Engagement

Please do not ask your friends or professors to help you in the coding. Do not copy the code written by others. It is OK to search or ask around if you have conceptual questions.

Next Step after this

If you are able to do this exercise well, you will be asked for 8-hour interview (either in-person or Skype video interview, scheduled some time between June 20th and June 23rd) that consists of coding exercises, algorithms, hardware architectures.