Box Muller based AWGN IP Design

Datasheet

Rahul Garlapati

***Abstract*—This electronic document contains the approach followed to design the AWGN IP. It contains the 32-bit Tausworthe URNG, Coefficients for pipeline stages Logarithmic unit, Square root unit, Sin/Cos unit and Leading Zero Detector.**

I. Features

Gaussian noise generator based on the Box-Muller method generates

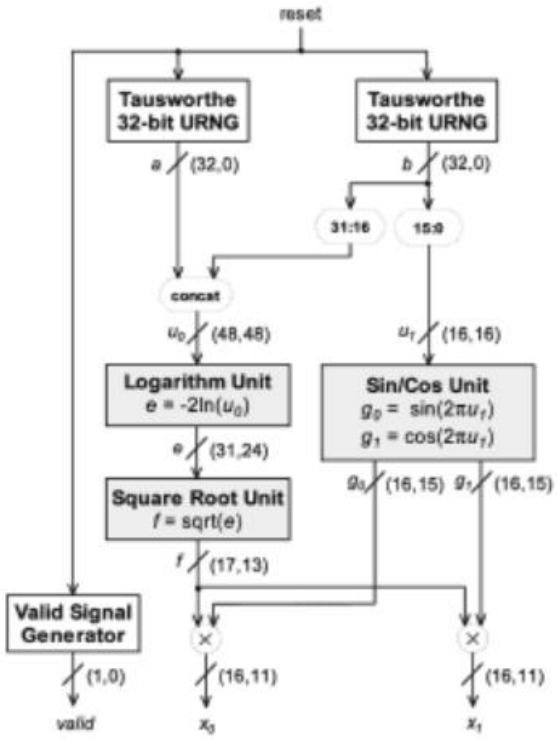
- 16-bit noise samples accurate to one unit in the last place up to 8.2ρ.
- Simulation size of over 10^{15} samples. Generally when evaluating channel codes, one needs 100 to 1000 bits in error.
- Noise is quantized to 16 bits, 5-bits for integer bit width and 11 bits for fraction bit width.
- The noise generator is relatively small, while producing 750 million samples per second at a clock speed of 375 MHz on a Xilinx Virtex-4 XC4VLX100-12 FPGA.

II. Applications

Software-based simulations can take several days to several weeks when the behavior at very low bit error rates (BERs) of such codes is being examined. Hardware-based simulations equipped with a fast and accurate noise generator offer the potential to speed of simulation by several orders of magnitude. Transferring software generated noise samples to the hardware device is highly inefficient and can be a performance bottleneck, hence it is desirable to have the noise generator on the hardware device itself.

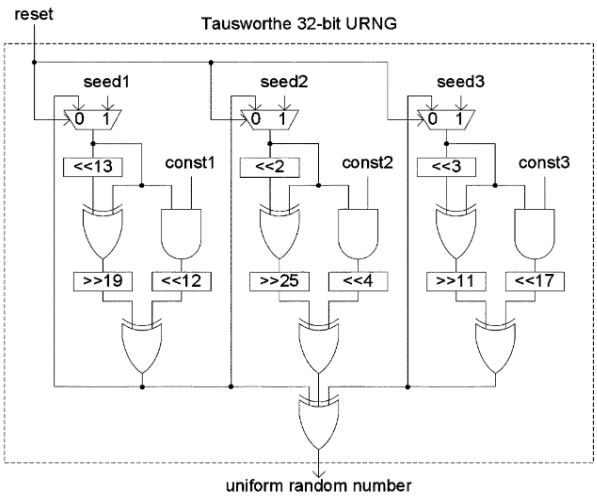
III. Hardware architecture

Box Muller Architecture is depicted in the below figure.



IV. Tausworthe 32-bit URNG

Tausworthe is implemented in the hardware using the below architecture taking the constants as seeds.



A. Error Analysis and Bit-Width Optimization for Noise Generator

B. Error Analysis for Sin/Cos Unit

- In the actual ROM, we store the two coefficients C1g and C0g as integers. The coefficients for C1g all contain six leading zeros in the fraction part, and some values of C0g turn out to be slightly larger than one. Moreover, all values of C1g and C0g are found to have the same sign.
- Hence, the bit-width required for C1g is 12 bits (with the six redundant leading six bits eliminated) and C0g is 19 bits.
- 128 segments are required for a degree one approximation. Hence the total table size needed for the sin/cos unit is (12 + 19)*128 = 3968 bits.

C. Error Analysis for Square Unit

- In the sin/cos units when we store the coefficients as integers, we need 12 bits and 20 bits for C1f and C0f respectively. We need to store two tables for the square root unit.
- Recalling that 64 entries are required for each table, the total table size is (12 + 20)*64*2 = 4096 bits.

D. Error Analysis for Logarithmic Unit

The minimal bit-widths for the three polynomial coefficients C2e,C1e,C0e are found to be 13, 22 and 30 bits, respectively. The approximation requires 256 segments, hence the total table size for the logarithm unit is (13 + 22 + 30)*256 = 16640 bits.

V. Leading Zero Detector

LZD is to count number of leading zeros of 48-bit number.

Where the logic of LZD4 is as below

$$V = d[0] \mid d[1] \mid d[2] \mid d[3]$$

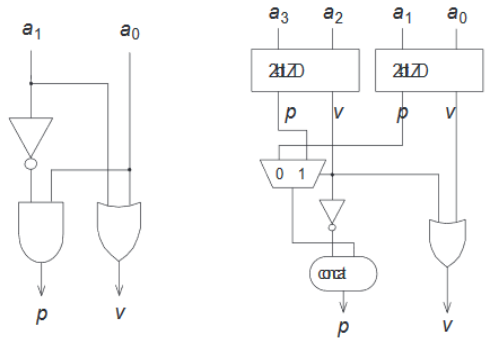
$$P = \{\sim(d[2] \mid d[3]), ((d[2] \mid d[3]) \ ? \sim d[3] : \sim d[1])\}$$

And the logic of LZD8 is

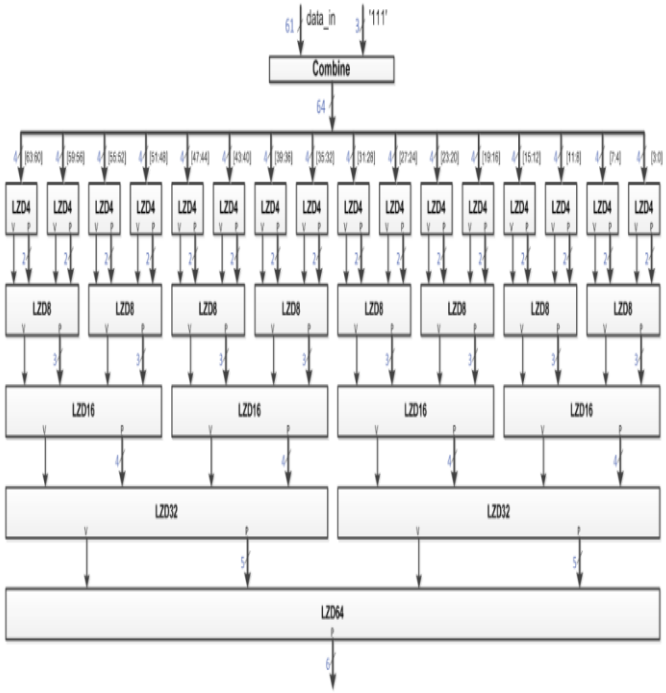
$$V = V[0] \mid V[1]$$

$$P = \{\sim V[1], ((V[1]) \ ? P[1] : P[0])\}$$

And so on. Note that in the last level LZD64 only P is used.



(a) 2-bit LZD (b) 4-bit LZD



Acknowledgment

I would like to thank TexasLPDC and Osso for giving me an opportunity to work on this project which helped me delve deeper into the requirements of a complete IP Design and it was a great learning experience.

References

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] Lee, D.-U.; Villasenor, J.D.; Luk, W.; Leong, P.H.W., "A hardware Gaussian noise generator using the BoxMuller method and its error analysis," in Computers, IEEE Transactions on , vol.55, no.6, pp.659-671, June 2006, doi: 10.1109/TC.2006.8

[2] http://www.iasj.net/iasj?func=fulltext&aId=87322

[3] http://www.ee.cityu.edu.hk/~rcheung/papers/tvlsi07_inv.pdf.