Entropy Analysis and System Design for Quantum Random Number Generators in CMOS Integrated Circuits

Scott A Wilber[*]

ABSTRACT: A quantum random number generator is implemented in an integrated circuit without the need for complex, bulky and expensive measurement equipment and circuitry. Quantum entropy, chaotic entropy and pseudo-entropy are defined and their combinations mathematically described. Models and design equations are provided for estimating the quantum entropy in the form of shot noise due to sub-threshold leakage, gate tunneling leakage and junction tunneling leakage in MOS transistors and CMOS IC's.

KEYWORDS: quantum, tunneling, entropy, QRNG, random number, shot noise, MOS transistor, CMOS

INTRODUCTION

In today's electronic information age we store or transfer almost every important or economically valuable document or bit of data in some type of encrypted form to prevent others from compromising privacy or stealing the information for nefarious uses. Random numbers are used in virtually every form of encryption or data security, and the source of these random numbers is a random number generator.

Initially virtually all random numbers for electronic use were produced by pseudorandom number generators (PRNGs). These generators are computer algorithms that are initialized with a seed or starting point and then produce an output sequence precisely determined by the computation of the steps in the algorithm. Every PRNG has a period, or length of bits after which the sequence begins to repeat. PRNG's have been improved over the years to so-called cryptographic or cryptographically secure PRNG's (CSPRNG), which are more secure because predicting future output given a sequence of previous bits is computationally intractable for current computer technology (Ruggeri, 2006). Advances in supercomputers and especially the development of quantum computers continue to chip away at the ultimate security of various encryption methods.

True random number generators (TRNG's), which derive their randomness from a physical entropy source, have been developed partly to enhance the security of PRNG's by providing seeds that are inherently non-deterministic. In addition some algorithms use a TRNG output directly to encrypt information. Some applications require a true random number generator that is theoretically unpredictable or has properties that only exist due to being derived from quantum mechanical measurements. Ideally a TRNG output should

[*] President of The Quantum World Corporation

be completely unpredictable and exhibit virtually perfect statistical properties. This paper will demonstrate the design and implementation of practical QRNG's in standard integrated circuits that provides output sequences with arbitrarily small statistical defects and quantum entropy at a specified target level.

TYPES OF ENTROPY

There are two general types of entropy sources that may be measured to generate true random numbers. The first type includes a physical process that is difficult or impossible to measure or too computationally intense to predict, or both. This is a chaotic entropy source. A common example known to most people is a lottery drawing machine. A set of sequentially numbered balls is placed in a chamber and they are constantly mixed by spinning the chamber or by blowing air through the chamber. Several of the balls are allowed to drop out of the chamber and the numbers marked on the balls represents the lottery drawing. The drawing is random because of the large number of interactions between the balls and the chamber resulting in a rapidly increasing number of possible movements of each ball. Not only is the complexity of these interactions exceedingly high, there is no apparent way of observing or precisely measuring all the internal variables of the balls, chamber and air flow.

A second and very different type of entropy source is quantum mechanical. Many microscopic particles or waves, such as photons, electrons and protons have quantum mechanical properties including spin, polarization, position and momentum. Given the proper setup for producing these particles, the specific values of their spin or polarization for example are not only unknown and theoretically unpredictable, they are physically undetermined until a measurement is performed. In these simple systems a measurement collapses the quantum wave function, producing one of two possible outcomes. According to the highly successful theory of quantum mechanics, the specific outcome is not knowable or predictable prior to the measurement. Only the *probability* of a specific outcome is computable. Therefore, the measurement of quantum entropy can produce the highest degree of non-determinism possible. Some cryptographers believe a quantum random number generator based on measurements of quantum mechanical properties can be used to produce the most secure encryption possible (Stipčević, 2011).

For this discussion a nonstandard definition of entropy is proposed; what may be called pseudo-entropy. Pseudo-entropy is the entropy mathematically calculated or inferred in pseudorandom sequences resulting solely from statistical randomness of the sequences. By definition pseudo-entropy is not real entropy because it has no physical source. This definition disregards any actual entropy contained in a true random seed that may have been used to initialize the PRNG. A PRNG's output bits have no more total entropy than the number of bits of entropy in the seed that was used to initialize the PRNG.

The concept of pseudo-entropy is useful in the context of randomness correction, also called whitening, cryptographic whitening or conditioning. In order to satisfy the statistical requirements for modern TRNGs, their output sequences are typically

subjected to some type of randomness correction to reduce statistical defects below a predetermined or desired level. A typical method of randomness correction is to perform an Exclusive-Or function (XOR) of the bits or words in the true random sequence with bits or words generated by a PRNG. A property of XORing random numbers from independent sources is that the resulting numbers have statistical properties better than or at least equal to the properties of the better, or most statistically random numbers (Davies, 2002). Since it is possible to design a PRNG with very good statistical properties, XORing these pseudorandom numbers with the statistically defective true random numbers will produce numbers with the same very good – or even slightly better – statistical properties. These resulting numbers are still considered nondeterministic or truly random, but the XORing process does not add any true, that is, chaotic or quantum entropy to the output numbers. Rather, the true entropy is supplemented with the pseudo-entropy in the pseudorandom numbers, and the two become statistically inseparable.

Cryptographic whitening is usually accomplished by passing statistically imperfect true random numbers through a cryptographic hash function, such as SHA-1. This has at best a similar effect on the entropy as the XOR processing in that the input random numbers are transformed by an algorithm so their statistical properties are greatly improved, but the total amount of true entropy per bit cannot be increased unless more numbers are fed in than taken out of the hash function. In the worst case the hash function does not entirely preserve the entropy provided at its input and the output numbers contain less true entropy and additional pseudo-entropy.

Conditioning a TRNG sequence does not always make it entirely unpredictable. If a TRNG output is predictable to a certain percentage due to a statistical defect prior to randomness correction or conditioning, the resultant numbers after conditioning may still be theoretically predictable to that same percentage. To make such a prediction requires knowledge of the algorithm used to perform the conditioning and sufficient computing power to reverse the process. This potential security loophole is most pronounced when the raw true random sequence has significant statistical defects or relatively low entropy, and an insufficient number of these low-entropy bits is used to produce each conditioned bit. This would be particularly problematic where the conditioning is a bit-by-bit XORing of deficient TRNG bits with PRNG bits.

Sometimes randomness correction methods are also used to extend or increase the number of output random numbers relative to the number of true random input numbers. This is normally accomplished by a deterministic algorithm such as a PRNG that is periodically reseeded by the true entropy source. Provided the algorithm is appropriately designed, the amount of true entropy per output bit is equal to the number of bits of true entropy input to the algorithm divided by the number of output bits that are actually used or observed. This is another way of saying the total entropy out is less than or equal to the total entropy in.

Statistical tests performed on random sequences cannot distinguish the various types of entropy used to produce the sequences, nor can the proportion of the different types be determined given a mixture of two or more types.

COMBINING ENTROPY

Entropy and predictability are terms with different meanings in different fields. Shannon entropy, H, defined for the specific case when only the binary possibilities of 1 and 0 exist, is $H = -(p(1)Log_2 p(1) + p(0)Log_2 p(0))$, where $p(1)$ and $p(0)$ are the probabilities of a one and a zero occurring respectively, and Log_2 is the logarithm in base 2, used when entropy is quantified in bits. For this discussion, $p(1)$ will generally be replaced with predictability, P, defined as the probability of correctly predicting the next number in a sequence of numbers, and $p(0)$ will be replaced by $1-P$, where $0.5 \leq P \leq 1.0$. Substituting predictability for the probabilities, H_P becomes $H_P = -(PLog_2 P + (1-P)Log_2(1-P))$. Sometimes the value of H is measurable or theoretically calculable and the value of P is required. Then $P = h^{-1}$, where h^{-1} is the mathematical inverse of the H_P equation. The inverse is normally performed numerically since there is no closed-form equation for it. The inverse of the entropy equation has two solutions, but only the one where $P \geq 0.5$ is to be taken.

In addition to the definitions of entropy and predictability it is essential to know what happens to the entropy content in resultant numbers when numbers containing various types and amounts of entropy are combined. For binary sequences, entropy is combined by applying the exclusive-or (XOR) function[†] to all the bits to be combined to produce a single resultant number. To calculate the predictability and entropy in resultant numbers, the predictability of each bit to be combined must be known. Sometimes it may be easier to estimate or measure the entropy and use the inverse function to get the predictability. The predictability of each number is then converted to a fractional predictability, $P_F = 2P - 1$. All the fractional predictabilities are then multiplied to produce their product, P_{PF}, and the combined predictability, P_C is finally calculated as $P_C = (P_{PF} + 1)/2$. The combined or total entropy, H_T, is calculated using the equation for H_P. Here is an example to illustrate the calculation. The per-bit entropy in each of three independent sequences of binary bits is $H_1 = 0.60984$, $H_2 = 0.97095$ and $H_3 = 0.88129$. The sequences are combined by performing the XOR function on each of the three bits corresponding to one bit from each sequence to produce a resultant sequence. The inverse entropy calculation is used to find $P_1 = 0.85$, $P_2 = 0.60$ and $P_3 = 0.70$. Their fractional predictabilities are $P_{F1} = 0.7$, $P_{F2} = 0.2$ and $P_{F3} = 0.4$. Their product is, $P_{PF} = P_{F1} \cdot P_{F2} \cdot P_{F3} = 0.056$, and their combined predictability is, $P_C = (P_{PF} + 1)/2 = 0.528$. Finally, using the entropy equation, $H_P = 0.99774$. From this example it is clear the combined entropy is greatly enhanced relative to the three original entropies. To further illustrate this, combine 32 independent sequences with entropy of only 0.01 in each sequence. The resultant sequence will have an entropy of 0.3015 – nearly a linear sum of entropy from the 32 sequences.

One may wish to calculate the number of independent sequences to be combined to achieve a target entropy in the resultant sequence. Starting with an entropy of 0.01 in

[†] The XOR function, or equivalently the parity function when more than two bits are combined, is used because it was proven to be the most efficient algorithm for improving the randomness properties when combining imperfect random sequences (Santha & Vazirani, 1984), and because its effect on combining entropy sources has been determined by the research underlying this paper.

each sequence and a target entropy of 0.99, first calculate the fractional predictability of the independent sequences and the resultant sequence. P_{FI} = 0.99827958499 for the independent sequences and P_{FR} = 0.11760477748 for the target resultant sequence. If the P_{FI} is similar but not the same for each independent sequence, it is possible to use the geometric mean of a sample of different sequences to estimate the appropriate P_{FI} for this calculation. The number of sequences required to provide the target entropy is *n*, where $P_{FR} = P_{FI}^{n}$. The solution to this equation is $n = LogP_{FR}/LogP_{FI}$ = 1243.063. The required *n* must be an integer and it should be rounded up to the next higher integer to ensure the minimum target entropy. With *n* = 1244, the combined predictability is, P_C = 0.5587075711, and the resultant entropy is, H_P = 0.990032. The same resultant entropy may be achieved by combining non-overlapping blocks of 1244 bits from a single sequence, provided the bits are independent. From a quantum non-deterministic perspective, these two approaches of combining independent sequences or blocks of bits in a single sequence are equivalent, but if other special properties of quantum measurements are desired, the single measurement of multiple independent sources may be required. Independent quantum sources are expected to be not entangled in a quantum mechanical sense. If the sources are entangled or partially entangled, the combined entropy may be substantially reduced. This is not a problem in most designs in standard integrated circuits, since entanglement does not normally arise except under special conditions that must be created intentionally.

In order to properly apply the design equations for combined entropy, it is necessary to know how to calculate or measure entropy of the various types produced by the source or sources being used in the generator. A basic principal in the combination of types of entropy is that they exist independently. That is, they do not mix, interact or change each other's per-bit entropy content. Calculations done for chaotic entropy are unrelated to pseudo-entropy, and calculations done for quantum entropy are unrelated to both chaotic entropy and pseudo-entropy. More specifically the fundamental unpredictability provided by quantum entropy is neither reduced nor increased by combining with either pseudo- or chaotic entropy, and the unpredictability of chaotic entropy is neither reduced nor increased by combining with pseudo-entropy. However, these combinations will provide an improvement in the statistical properties of the resulting sequences. Following are some examples to illustrate this principle:

1) A chaotic entropy source is measured to produce independent binary bits with a probability of a "1" occurring, $p(1)$ = 0.55. The chaotic entropy of the measured bits is 0.992774 as calculated using the Shannon entropy equation given above. The entropy may also be measured statistically using, for example, an update of Maurer's "Universal Test" (Coron, 1999). With a block length of 7 bits and a total of 1 million bits in the test sequence, the test gives a per-bit entropy of 0.99268(43) (± 1 standard deviation (SD)). While the entropy is fairly high, the predictability is 0.55, which is unacceptable for most applications. Therefore the chaotic bits are combined with a pseudo-entropy sequence by XORing one bit from each sequence to produce bits in a resultant sequence. By definition, there is no actual entropy, i.e., H = 0.0, in the pseudo-entropy sequence, hence its P = 1.0 and P_F = 1.0. P_F = 0.10 for the chaotic bits and P_{PF} = 0.10 for these two sequences combined. Finally, their total combined predictability is, P_T = 0.55 and H_P =

0.992774 per bit. These values are exactly the same as the original bits from the chaotic entropy source, but the statistical properties of the resultant sequence have been corrected so the statistically measured entropy will appear to be 1.0. However, given appropriate computational tools and knowledge of the pseudo-entropy algorithm and initial state, the bits in the resultant sequence are still 10 percent more predictable than bits with chaotic entropy of 1.0.

2) An entropic source produces bits with 0.8 bits per bit of total entropy, which includes 0.1 bits per bit of quantum entropy. The goal is to design a quantum random number generator with 0.99 bits of quantum entropy per output bit. In a mixed entropy source of this type the amount of quantum entropy must be estimated or calculated from theoretical knowledge of the source itself. While the total entropy can be approximately measured, there is no purely statistical measure that can distinguish between the two types. As with the first example, combining chaotic entropy with quantum entropy does not change the degree of quantum unpredictability and does not change the amount of quantum entropy. This is true whether the two types of entropy are mixed because of the nature of the source or by combining bits from separate sources. To clarify the difference between the entropy types, consider that just prior to the measurement of a bit, the quantum entropy source exists in a superposition of states that may represent either a one or a zero, describable only as a probability of the two possible states. The chaotic entropy component follows a deterministic and at least theoretically predictable signal evolution. Its non-determinism arises both from a lack of access to, and measurement of the variables that affect the signal, and to their computational intractability. In a manner similar to the first example above, the addition or presence of chaotic entropy may increase the statistically-measured randomness of the sampled or resultant numbers, but it has no effect on the amount of quantum entropy.

First calculate the predictability arising from the quantum entropy. The inverse entropy calculation gives P = 0.987013 and P_{FI} = 0.974026. The relative predictability of the target resultant bits is, P_{FR} = 0.1176048. Finally, the number of bits or independent sequences that need to be combined is, $n = LogP_{FR}/LogP_{FI}$ = 81.33, rounded up to 82. Now the statistical defect in the resultant sequence may also be calculated. From the total entropy of 0.8, P = 0.756996 and P_{FI} = 0.513992. Using n = 82, the resultant relative predictability is, $P_{FR} = P_{FI}^{n} = 1.988\times10^{-24}$, which is also the approximate size of statistical defects in the resultant sequence. This level of defect is immeasurably small under any testing conditions, so randomness correction or conditioning would be unnecessary. The corresponding total entropy is, H_T = (1 – ε) bits per bit, where ε = 2.85×10^{-48} while the quantum entropy is, H_Q = 0.990346. The total entropy per bit cannot exceed 1.0 so the output sequence has an entropy of 0.990346 quantum bits per bit and $0.009654 - 2.85\times10^{-48}$ chaotic bits per bit.

ENTROPY FROM VARIOUS SOURCES

There are two broad types of noise sources that relate to the production and measurement of entropy. They are extrinsic and intrinsic sources. Extrinsic sources are those which are not directly part of the generator source being measured and are coupled to the source by electromagnetic fields, power supply variations or even mechanical vibrations. Intrinsic sources are inherent in the generator source being measured and arise from fundamental physical principles. In transistors and integrated circuits the intrinsic sources include shot noise from diffusion and tunneling currents, thermal noise, flicker or 1/f noise and generation-recombination noise. Most extrinsic sources can be eliminated or reduced by proper design and shielding of the generator source while intrinsic sources are usually not reducible below their theoretical value.

The design equations presented in this paper require a measurement or estimate of the lower bound of chaotic and/or quantum entropy in the particular generator source being used. Most modern-day digital integrated circuits are constructed using MOS transistors in a complementary or CMOS configuration. The entropy produced in these transistors is measureable by amplifying their analog noise signals or by detecting variations in transition times or jitter in an oscillating signal passing through them. The latter is done by sampling a free-running ring oscillator with another oscillator, or by sampling an output transition that has been passed through many stages of a delay line, or by a combination of both. A ring oscillator is a multi-stage delay line containing an odd number of inverting elements with its output connected to its input. Each element in the delay line or ring oscillator adds a certain amount of jitter to the signal transition as it passes through. The statistical distribution of the jitter due to intrinsic sources is approximately normally distributed, and the total cumulative jitter from these sources is the jitter introduced by a single stage multiplied by the square root of the number of stages the transition has passed through before being measured. As a practical matter the effective jitter is that which accumulates in a continuously operating system in the time between measurements. Delay line sources or ring oscillators that are reset to a predetermined state prior to each measurements accumulate jitter from the time a signal is initiated or the ring enabled.

A typical integrated CMOS gate will have three predominant noise components affecting its output. One is extrinsic and is a type of power supply noise known as digital switching noise. Switching noise is a high frequency voltage variation caused by the combined current impulses that occur as many logic gates in the same integrated circuit switch from high to low or from low to high. Individual switching impulses may be deterministic, but the combination of hundreds or thousands of these impulses are not possible to observe and predict, especially if there are many free-running oscillators operating in the same IC. Switching noise is a type of chaotic noise. Extrinsic noise should not be solely relied on as an entropy source in a secure random number generator system because of the potential to observe and even inject patterns into the generator circuit.

The other noise components are intrinsic. They are thermal or Johnson noise and shot noise. Thermal noise is caused by thermal agitation of charge carriers in any resistive

element. This noise is chaotic and non-deterministic. Its amplitude is proportional to the square root of absolute temperature, resistance and bandwidth. Shot noise occurs because charge carriers, which are tiny indivisible packets of the current flow, may bunch up or spread out in a statistical distribution. It is fundamentally non-deterministic and its amplitude is proportional to the square root of the total current flow and bandwidth. It may be formally considered either classical or quantum mechanical or a mixture of both depending on the circumstances of its generation. Shot noise occurs when charge carriers pass through a potential barrier such as a diode junction in MOS transistors. Shot noise in MOS transistors consists of three major components, which are sub-threshold leakage, gate direct tunneling leakage and junction tunneling leakage.

ESTIMATING SHOT NOISE IN MOS TRANSISTORS AND CMOS CIRCUITS

Charge carriers must cross an energy gap, such as a p-n junction, or tunnel across an insulating boundary to exhibit shot noise phenomena. Charge carriers that do not cross a barrier, such as electrons flowing in normal conductors, are highly correlated due to electrostatic forces and the Pauli principle, which reduce shot noise to low levels. Shot noise results from statistical variations of the flow of uncorrelated, quantized charge carriers and is proportional to the square root of the average number of carriers crossing a boundary in a given time interval, which is a definition of current. Shot noise in MOS transistors arises from sub-threshold leakage, tunneling across the insulating layer under the gate, known as gate leakage or gate oxide leakage and junction tunneling leakage due to high electric fields across reverse-biased p-n junctions. In a CMOS structure, both the pMOS and nMOS transistors contribute to the leakage into the load capacitance (also called node or output capacitance), *C*, where the variations in current appear as voltage variations in the output. In addition, both the gate leakage and junction leakage are partitioned into source leakage and drain leakage components. At their peak value in symmetrical transistors the tunneling currents are split equally between drain and source. When each transistor is turned off, its sub-threshold leakage and junction leakage are maximum and its gate leakage is minimum. As each transistor is turned on, its sub-threshold leakage and junction leakage decrease to essentially zero and its gate leakage increases to maximum (Liu & Kursun, 2007) and (Xue, Li, Deng, & Yu, 2010). The magnitude of the noise voltage across the load capacitance due to the sub-threshold leakage is calculated by integrating the leakage shot noise current delivered to the load capacitance through the equivalent output resistance over the frequency spectrum of the noise. Sarpeshkar, *et al*, (Sarpeshkar, Delbrück, & Mead, 1993) showed that shot noise in sub-threshold MOS transistor leakage accounts fully for what was previously thought of as thermal noise. The noise voltage is $V_S = \sqrt{kT/C}$, where V_S is the shot noise voltage in volts rms, *k* is Boltzmann's constant, 1.38065×10^{-23}, *T* is the temperature of the CMOS transistors in degrees Kelvin and *C* is the load capacitance in Farads. Gate leakage and junction leakage are only slightly affected by temperature but are strongly dependant on supply voltage. All three leakage currents are essentially independent in a simple inverter but they become state dependant in more complex gate structures and transistor stacks with multiple inputs (Mukhopadhyay, Raychowdhury, & Roy, 2005).

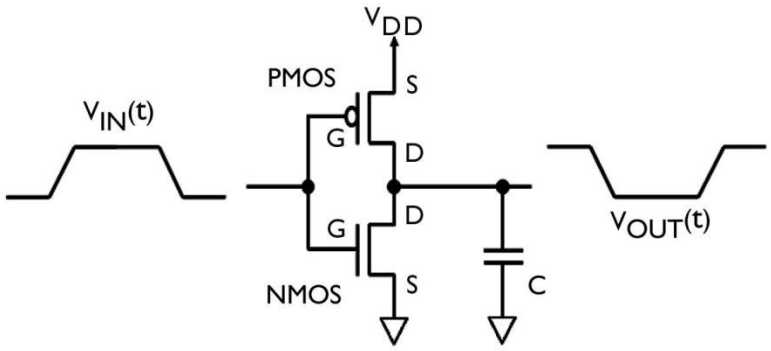
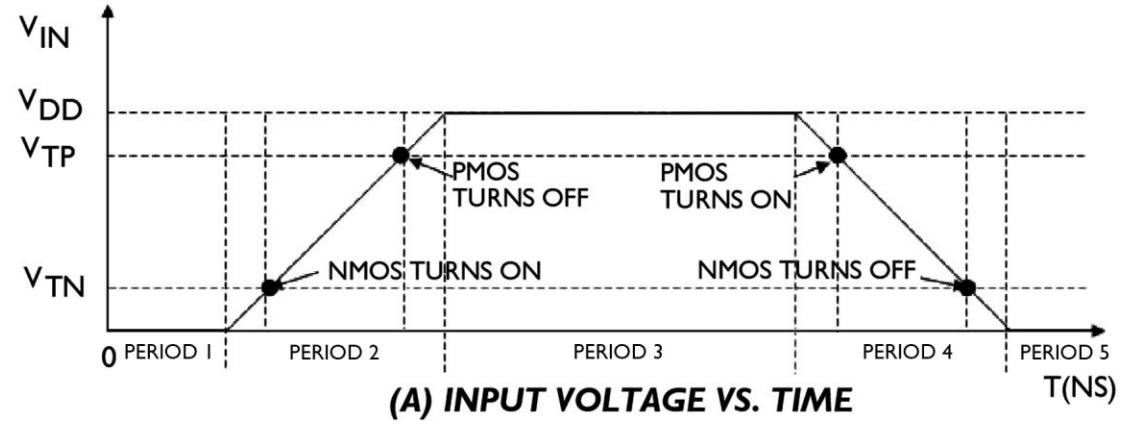
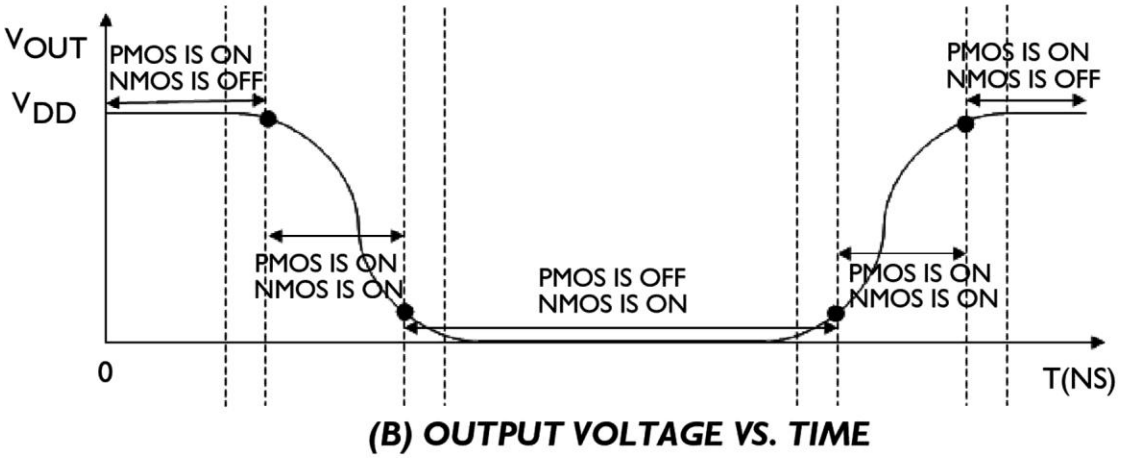


Fig. 1 shows a CMOS inverter representing the typical output structure of a CMOS gate in an integrated circuit.

FIG. 1



(A) INPUT VOLTAGE VS. TIME



(B) OUTPUT VOLTAGE VS. TIME

FIG. 2

Figs. 2(A) – 2(B) illustrate the details of the states and transitions of transistors in Fig. 1, and the relationship between the input and output voltages versus time.

In Period 1 the input is near zero volts (a low logic state) and the output is near V_{DD} (a high logic state). The pMOS transistor is turned on and the nMOS transistor is off. In Period 2 the input is in transition from a low to a high state. During this period the nMOS gate voltage, V_{GS} changes from sub-threshold, to the linear region and finally turns the

nMOS transistor fully on. During the same period the pMOS transistor's V_{GS}, initially near negative V_{DD}, increases to near zero volts, turning the pMOS transistor off. In Period 3 the nMOS transistor is turned on and the pMOS transistor is turned off. Period 4 is a high-to-low, or negative transition of the input voltage. In this period the transistors follow the steps of Period 2 in reverse, ending in Period 5 with the transistor states the same as in Period 1.

MOS Transistor Leakage Currents

Table 1 shows estimated maximum leakage currents for 65nm nMOS and pMOS transistors normalized to the sub-threshold leakage of the pMOS transistor. Temperature is 45°C and V_{DD} is 1.2V. The width of the pMOS transistor is scaled to 2× the nMOS transistor to approximately balance the lower mobility in the pMOS device. This also scales the pMOS leakage values by the same factor since they are directly proportional to width. Values adapted from (Gupta & Khare, 2013), (Liu & Kursun, 2007), (Zhang, Parikh, Sankaranarayanan, Skadron, & Stan, 2003) and (Xue, Li, Deng, & Yu, 2010). The junction leakage indicated in Xue, *et al*, is higher than the gate leakage in 65 nm transistors, but a conservative estimate equal to gate leakage will be used for calculations in this paper.

	nMOS	pMOS
I_{sub}	0.77	1.0
I_{gate}	0.39	0.035
I_{junc}	0.39	0.035

Table 1

When the source-drain voltage approaches zero, the gate leakage is partitioned equally between the source and the drain. This condition exists when one transistor as shown in Fig. 1 is turned on and the other is off. The transistor that is turned on has a source-drain voltage near zero and the gate leakage is maximum, but only 50 percent of the leakage current flows into the drain (source-drain partition) where it contributes to the shot noise voltage at the output of the gate. The gate leakage in the transistor that is turned off is orders of magnitude less than the one turned on, so it is effectively zero. Junction leakage is maximum for the transistor that is turned off with its source-drain voltage at maximum. In this state the total junction leakage is partitioned about equally between source and drain.

Noise Voltage from Leakage Currents

The noise voltages due to gate leakages into the load capacitor, *C*, can be derived by integrating the shot noise from the leakage currents. However, a simple estimate of the normalized gate leakage shot noise voltages is made by setting them equal to the square root of the ratio of gate leakage current to sub-threshold leakage current. The normalized nMOS and pMOS gate leakage currents from Table 1 are first multiplied by the geometric mean of the pMOS and nMOS sub-threshold leakage currents, which approximately accounts for the differences in effective resistance and resulting bandwidth

of the two transistor types. Then the currents are divided by two (partitioned), which gives 0.17 for the nMOS and and 0.015 pMOS, and finally the square root of these values is 0.41 and 0.12 V rms respectively. The normalized shot noise voltages from junction leakages are calculated in the same way as for the gate leakages. In contrast to the sub-threshold leakage that has a period during a transition where neither transistor is producing leakage, the gate leakages and junction leakages have a significant value for both transistors during a transition.

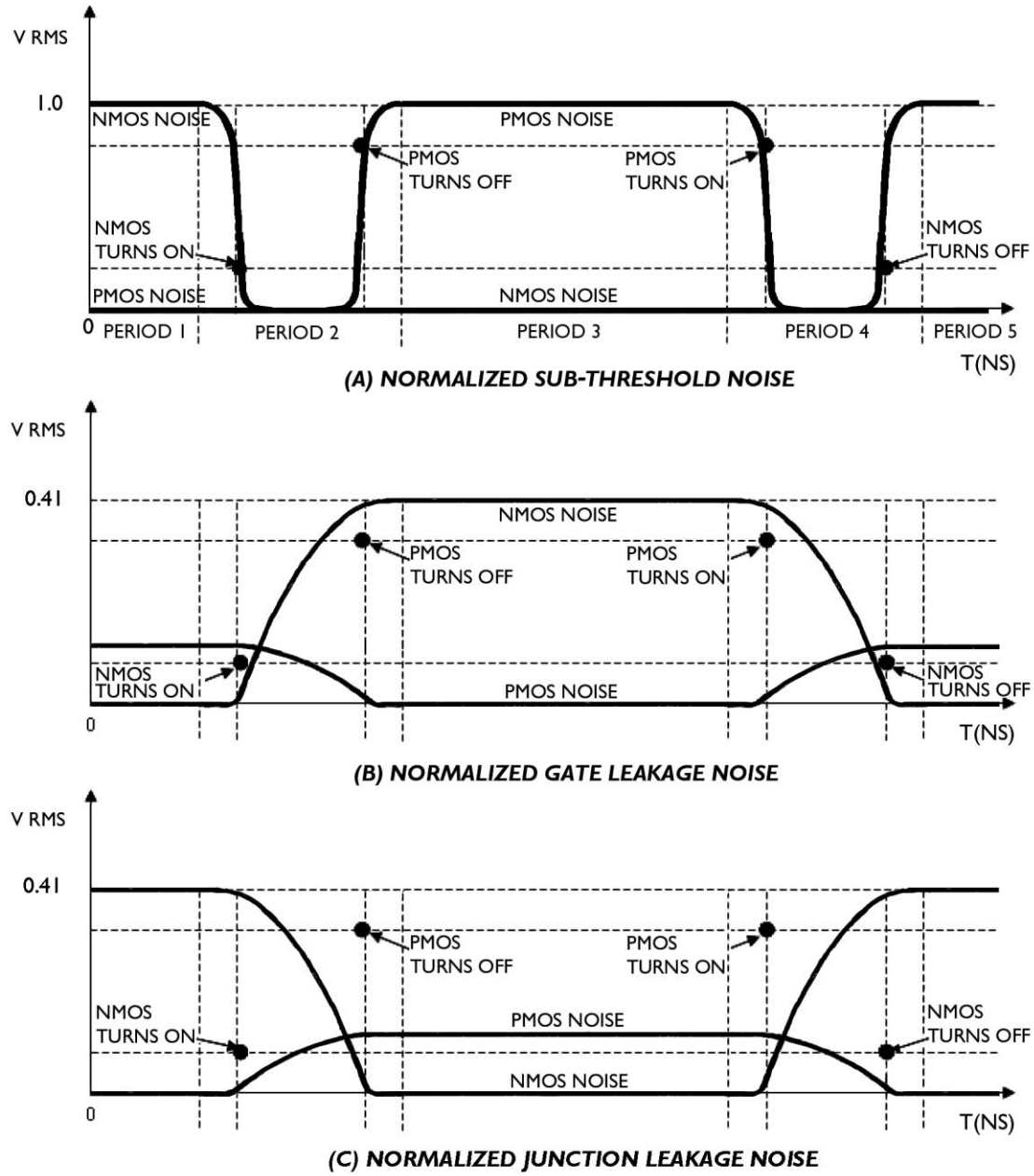


FIG. 3

FIGS. 3(A) – 3(C) show the shot noise voltages during the periods described for FIG. 2.

The noise voltage levels in Figs. 3(A) through 3(C) are normalized by dividing by the maximum of the sub-threshold voltage levels. Table 2 summarizes the normalized shot noise voltages for the various leakage types for nMOS and pMOS transistors. Note that even though the sub-threshold currents are not quite equal for nMOS and pMOS transistors, this leakage is a diffusion process in thermal equilibrium and the equipartition theorem indicates the total noise voltage for both transistors together is, $V_S = \sqrt{kT/C}$.

	nMOS	pMOS
I_{sub}	1.0	1.0
I_{gate}	0.41	0.12
I_{junc}	0.41	0.12

Table 2

Fig. 3(A) shows only the sub-threshold leakage shot noise voltages. Sub-threshold leakage current flows through each transistor when its gate-source voltage, V_{GS}, is near or below the threshold voltage, and it is maximum when V_{GS} is zero volts and the transistor is turned off. Note, the V_{GS} is negative for the pMOS transistor and positive for the nMOS transistor. During Period 1 the nMOS transistor is off and its leakage noise is maximum so its normalized value is 1.0. At the same time the pMOS transistor is turned on so its sub-threshold leakage noise is zero. During Period 2 the gate voltage increases through the threshold voltage of the nMOS transistor and the sub-threshold noise rapidly decreases to zero as the transistor turns on. When the gate voltage increases further and nears the threshold voltage of the pMOS transistor, it turns off and its sub-threshold leakage shot noise voltage rapidly increases from zero to its maximum at the end of the period. During Period 3 the pMOS noise is at its maximum and the nMOS noise is zero. In Period 4 the pMOS transistor turns on and the nMOS transistor turns off, changing the noise levels in reverse order to Period 2. Finally in Period 5 the noise levels are the same as in Period 1.

Fig. 3(B) shows the normalized shot noise voltage due to gate tunneling leakage current. Gate leakage is a challenge for integrated circuit (IC) designers as the feature dimensions are constantly reduced in order to reduce power consumption, increase speed and pack more transistors into each IC. As dimensions are scaled, the thickness of the insulating oxide layer under the gate is also decreased. This results in exponentially increasing gate leakage current, which has become a significant component of the total power dissipation in modern CMOS IC's through the 65nm technology node. Subsequent nodes will begin to rely on high-K gate dielectrics and other methods to reduce this leakage component, or at least keep it from increasing (Cao, 2011).

Fig. 3(C) shows the normalized shot noise voltage due to junction tunneling leakage current. Junction leakage is a significant new issue for deep sub-micron transistors starting at about 65nm and smaller. Simple dimensional scaling is not sufficient to maintain desired performance at these dimensions. High substrate doping and "halo" profiles near the source and drain junctions of the channel reduce the depletion region widths but also dramatically increase tunneling current through these junctions when they are reverse biased.

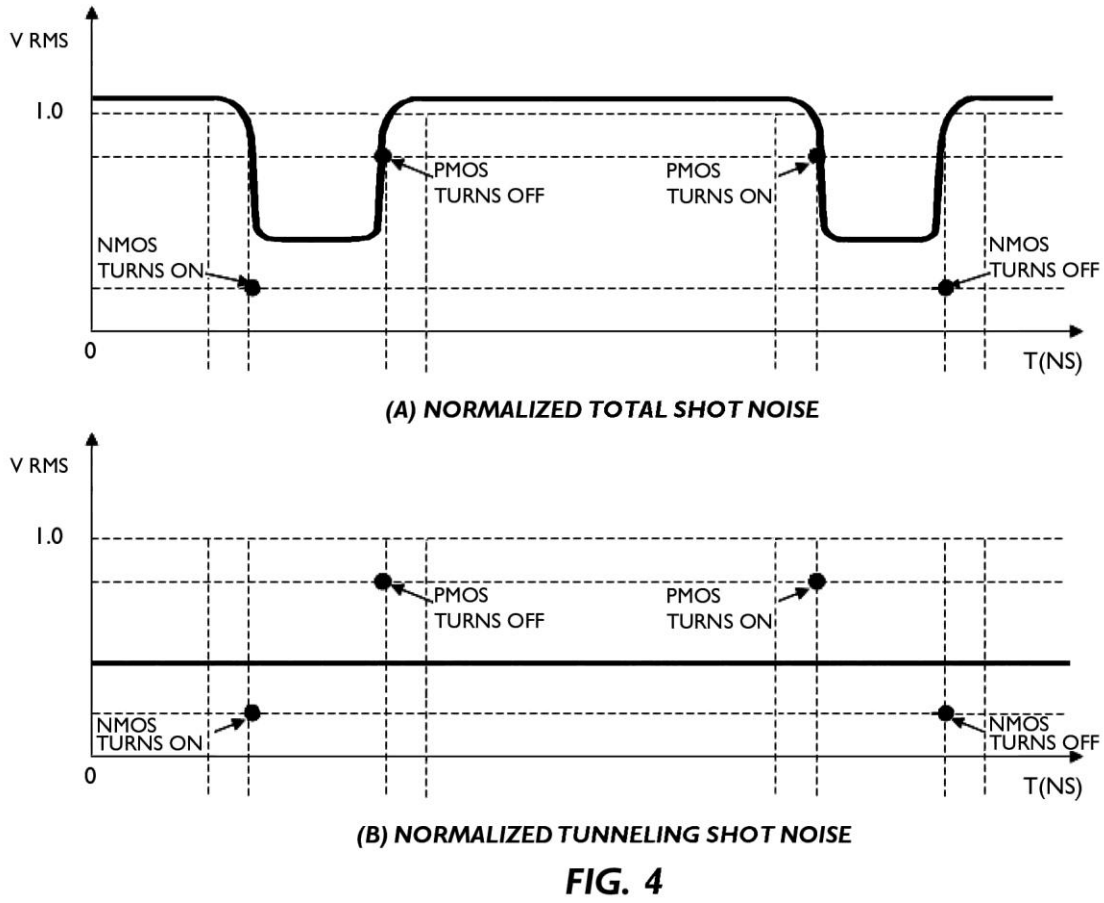


FIG. 4

Figure 4(A) – (B) are the combined normalized shot noise voltages.

The components of shot noise voltage are independent and approximately normally distributed so the sum of these noise sources is the square root of the sum of the squares of the individual sources (added in quadrature). FIG. 4(A) shows the total normalized shot noise voltage at the output of the CMOS gate. This is the sum of the sub-threshold noise voltage of FIG. 3(A), the gate leakage noise voltage FIG 3(B) and the junction leakage noise voltage FIG 3(C) for both the pMOS and nMOS transistors added in quadrature. The normalized maximum value in Period 1 is 1.087V rms, in Period 2 the value is 0.427V rms and in Period 3 the value is 1.087V rms. The weighted average for both stable transistor states and transition states is $V_S = \sqrt{1.087^2 t_{fs} + 0.427^2(1 - t_{fs})}$, where t_{fs} is the fractional time the signal is stable. The example calculations in this paper use two ring sizes of 19 and 24 LUT delays per cycle, each including 2 LUT delays during which a transition occurs. The corresponding V_S values are 1.037 and 1.048 times the normalized sub-threshold values respectively. When tunneling leakage is not included these two values become 0.946 and 0.957 respectively, showing gate leakage and junction leakage together only contribute about 9% to the total shot noise voltage. A value of V_S = 1.0 times the nominal value will be used throughout this paper for the total from all sources. When only gate tunneling leakage and junction leakage are taken, Fig. 4(B), V_S is about 0.427 times the nominal value.

SHOT-NOISE-BASED QRNG DESIGN IN A CMOS IC

An example of a shot-noise-based QRNG is designed in a 65-nm Field-Programmable Gate array (FPGA). Such an FPGA is one of the devices in the Cyclone III family (Altera Corporation, 2012) made by Altera Corporation. A specific device in this family is the EP3C10E144C8N, which contains 10,320 programmable logic elements, each comprising one 4-input look-up table (LUT) and one latch. Each LUT is programmable to create a wide range of logic functions such as AND, OR and XOR. To estimate the theoretical quantum entropy available from each LUT requires a reasonable model of its physical design and operation.

Approximate model of a LUT in Altera Cyclone III FPGA's

A first-order approximation of a LUT is to treat it as a normal logic gate, such as a simple inverter shown in Figure 1. It is necessary to know the slew rate of the inverter and the load capacitance, C, to make the first estimate of quantum entropy. The slew rate is calculated from rise and fall times, which are estimated from the propagation delay through the LUT. Although there is no simple relationship between these two parameters, rise and fall times are approximately equal to or a little longer than delay times in a simple CMOS inverter circuit. The propagation delay, τ_p, of the LUT is found by measuring the average frequency of several ring oscillators and calculating $\tau_p = 1/(2\, n_{lut}\, f_{ring})$, where n_{lut} is the number of LUT's in the ring and f_{ring} is the frequency of oscillation. A ring oscillator was designed with 11 non-inverting gates and one inverting gate which were arranged vertically in a single logic block (LAB) to minimize interconnect delays and variations between rings. An average ring of this design oscillated at 155 MHz giving a propagation delay of 268.8 ps[‡]. The rise and fall times, which are assumed to be equal by design, are approximately equal to the propagation delay. The slew rate is $0.8(V_{OH} - V_{OL})/T_r$, where V_{OH} and V_{OL} are the output high and low voltage levels respectively, and T_r is the rise (or fall) time. $V_{OH} - V_{OL}$ is effectively equal to V_{DD} or nominally 1.2V, giving a slew rate of 3.57 V/ns.

The load capacitance was first estimated by using the Altera Early Power Estimator (Altera Corporation, 2013) to calculate the dynamic power consumed by one LUT. Dynamic power is composed of two components: load power, P_L, which is caused by charging and discharging the load capacitance and short circuit power, P_{SC}, which is due to current that flows when both transistors are turned on during a transition. The total dynamic power is

$$P_{DYN} = C_L V_{DD}^2 f + V_{DD}\, I_{max} \left(\frac{T_r + T_f}{2}\right) f$$

where C_L is the load capacitance, V_{DD} is the supply voltage, f is the switching frequency and I_{max} is the maximum short circuit current during a transition. The short-circuit power

[‡] The Altera compiler does not always select the minimum delay path through input "D" of the 4-LUT, but sometimes routes the signal through input "C". This results in a decrease in ring oscillator frequency and a proportional increase in average delay time.

is typically between 10 and 20% of the total dynamic power (Korkmaz, 2005). For this estimate P_{SC} is conservatively taken as 10% of the total dynamic power. From the power indicated by the Power Estimator, C_L = 120 fF.

The power calculator does not take into consideration the specifics of input address configuration and fan out of the LUT's used in a ring oscillator, so a measurement was made to refine this result. The equivalent of 135 – 12-LUT rings were placed in an FPGA. The inverting gate in each ring was configured to be turned on or off by using an external jumper. The current difference with the rings turned on versus off was 33.57 mA, V_{DD} was 1.222V and a ring oscillator frequency of 155 MHz was measured. Taking the fraction of short circuit power at 10% of dynamic power yields, C_L= 98.5 fF. This value will be used in the following calculations. Using 20% short-circuit power would have resulted in a C_L of 87.6 fF and a 6% increase in shot noise voltage.

The shot noise voltage at the output of the LUT is the noise voltage developed across the load capacitance due to shot noise in leakage currents in the CMOS transistors. While an in-depth calculation of the shot noise is very complex, an approximate solution is quite simple. The average shot noise voltage is about $V_S = \sqrt{kT/C}$, where V_S is the noise voltage, in volts rms, k is Boltzmann's constant, 1.38065×10^{-23}, T is the temperature of the CMOS transistors in degrees Kelvin (about 318 degrees or 45 degrees Centigrade during operation) and C is the load capacitance in Farads. Solving for V_S gives 2.11×10^{-4} volts rms in the output of the LUT due to shot noise. Now the voltage noise must be converted to a transition time jitter. This is simply the shot noise voltage divided by the slew rate (McNeill & Ricketts, 2009, p. 166) which gives $J_{LUT} = 5.91 \times 10^{-14}$ s rms. This 59.1 fs rms is the transition jitter in a single LUT due solely to shot noise.

Approximate Quantum Entropy in the Simplified LUT Model

In a ring oscillator a single edge continuously passes through one LUT after another. As this happens the time jitter of that edge accumulates according to the equation $J_T = J_{LUT}\sqrt{n_L}$, where J_T is the total jitter and n_L is the number of LUT's the edge has passed through. For this example a ring oscillator is designed with 12 gates including one inverting gate and 11 non-inverting gates. Each cycle of the ring oscillator is composed of 12 delays for the negative half-cycle and 12 delays for the positive half-cycle, so the total period is 24 times 268.8 ps = 6.451 ns resulting in a frequency of 155 MHz. The total jitter for each cycle is $\sqrt{24} \times 59.1 \times 10^{-15}$ = 290 fs rms. The fractional jitter, J_F, is the total jitter per cycle divided by the cycle period. $J_F = 2.9 \times 10^{-13} / 6.45 \times 10^{-9} = 4.5 \times 10^{-5}$ rms. US Pat. no. 6,862,605, (Wilber, 2005) discusses how to calculate the entropy of a sampled oscillatory signal given rms jitter as a fraction of the oscillatory signal period. (See also: (Sunar, Martin, & Stinson, 2007)). The entropy is calculated numerically by first calculating the average probability of correctly predicting the next sampled value of the oscillator signal and then using Shannon's entropy equation as described above. The fractional jitter must be adjusted to an effective jitter, $J_E = J_F\sqrt{f_{osc}/f_{samp}}$, where f_{osc} is the ring oscillator frequency and f_{samp} is the sampling frequency. This adjustment accounts for the fact the effective cumulative jitter at each sample time is that jitter which

accumulates since the previous sample. The following Mathematica program performs the required numerical calculations:

```
prob[mu_, rho_]:=Sum[CDF[NormalDistribution[mu, rho], x+1/2]-
CDF[NormalDistribution[mu, rho], x], {x, -Round[6 rho], Round[6 rho]}]
avgprob[rho_, hf_, lf_]:=(ro=rho Sqrt[hf/lf]; divisions=Max[1000, Ceiling[5/ro]];
If[ro>.9, .5, N[2Sum[prob[mu, ro], {mu, 0, 1/2, 1/(4divisions)}]/(2divisions+1)-
Sum[prob[mu, ro], {mu, 0, 1/2, 1/(2divisions)}]/(divisions+1)]])
H[rho_, hf_, lf_]:=(apr=avgprob[rho, hf, lf];
(-1/Log[2])(apr Log[apr]+(1-apr)(Log[1-apr])))
```

The function that calculates entropy is H[rho_, hf_, lf_], where the arguments, rho, hf and lf are the fractional jitter, J_F, and the ring oscillator and sampling frequencies respectively, and the output, apr, is the average predictability, P. When the fractional jitter gets smaller the number of divisions used in the function avgprob must be increased. $5/J_E$ divisions rounded up to the next higher integer will yield about three significant digits of accuracy for J_E down to 0.00001. Using the values of J_F, hf and lf for this example design, 4.5×10^{-5} rms, 155 MHz and 128 MHz respectively, the above program gives H = 0.0011904, P = 0.999921012 and P_F = 0.99984202.

QRNG Design using Simplified LUT Model

To achieve a target quantum entropy of 0.99 bits per bit in the final output, a number of bits of the type described must be combined by XORing in non-overlapping blocks to produce each output bit. That number of bits is, $n = Log(0.1176048)/Log(0.99984202)$ = 13,548.

A Better LUT Model

A LUT does not seem to be well approximated by a simple gate model. Figure 1(A) shows an inverting or non-inverting gate equivalent in a typical LUT. A 4-LUT or four-input LUT is actually a type of static RAM (SRAM) with the four inputs multiplexing a data path through pass transistors from one of 16 possible SRAM bits to the data output. When a single input is needed the minimal-delay circuit using only the final multiplexor and pass transistors, P0 and P1, is required to be active. The rest of the multiplexors and pass transistors of the 4-LUT are typically inactive and are not shown in the figure. The active input, *IN*, selects one of two data paths from the output of the SRAM (or previous multiplexor stage) by turning on one pass transistor while turning the other one off using the complement of *IN*. The output of the active pass transistor is connected to the input of an inverter, which provides a buffered output for the LUT. The output buffer includes a pMOS transistor on its input that actively bootstraps slowly rising input voltages when nMOS pass transistors are used. The gate is either inverting or non-inverting depending on the values set for X_0 and X_1.

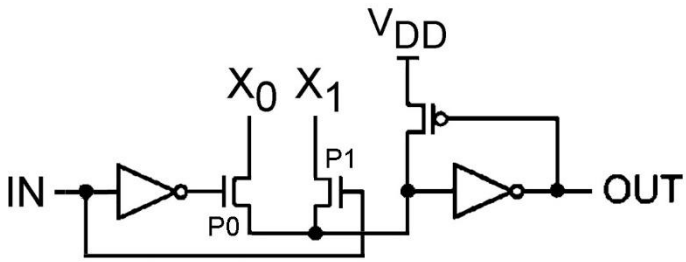


FIG 5

FIG. 5 is a better model of a LUT implementing an inverting or non-inverting gate. Variations in typical LUT design may include CMOS pass transistors versus the nMOS shown here, and an additional inverter prior to the final signal *OUT*.

For purposes of shot noise calculations, these stages of the LUT are more closely modeled by two consecutive CMOS inverters, each with its own load capacitance. The rise and fall times and the load capacitances are taken to be equal for the inverters, and are set to half their respective values for the simple LUT model. The noise contributed by the pass transistors and bootstrap transistor is not explicitly included in this model. The estimated slew rate becomes 7.14×10^{9} volts/second, and the load capacitances, which were lumped together in the simple model become 49.25 fF. The shot noise becomes 2.9854×10^{-4} volts rms in each of the two inverter stages, and the jitter is 4.18×10^{-14} s rms. The total jitter for these two stages in the LUT is $\sqrt{2} \times 4.18\times10^{-14} = 5.91\times10^{-14}$ s rms, the same amount calculated for the simpler model.

While the two-inverter model is still a somewhat crude representation of the exact implementation of the LUT circuitry, this exercise indicates the results obtained by using an improved model do not diverge from those obtained by using lumped values in the simple model.

PURE QUANTUM™ DESIGN: MODEL PQ32MU

This section describes a specific design of a QRNG in an Altera Cyclone III FPGA, which follows the general form used in the preceding example. The sampling of entropy is made more efficient, that is, requiring fewer resources in the FPGA, by placing three connections or taps at three equally spaced positions on the 12-LUT ring oscillator. These three tap signals are combined in a 3-input XOR gate to produce an enhanced ring oscillator output signal at three times the ring oscillation frequency. The three signals provide the equivalent of three independent entropy sources because the time spacing between the taps is very large compared to the jitter distribution at each tap (over 10,000 standard deviations), and therefore the amount of mutual entropy due to sampling of overlapping jitter distributions is insignificant. The tripled, enhanced output frequency triples the probability of sampling a ring oscillator output signal exactly during a transition when the shot noise-induced jitter makes the measurement quantum mechanically indeterminate. The fractional predictability from the enhanced output is the fractional predictability of the single tap output cubed.

The enhanced outputs of multiple rings of the design described in the previous section, but of different oscillatory frequencies, may be combined by XORing them together.

XORing multiple enhanced ring outputs produces a resultant signal containing the sum of the individual signal frequencies. There are two limitations with this approach: the combined frequency should not exceed the switching speed of the LUT (Dichtl, Meyer, & Seuschek, 2008), and the fractional jitter must still be small enough so each transition is effectively independent of all others to maintain insignificant mutual entropy during sampling. The maximum switching frequency of a LUT in the example FPGA is about 1.8 GHz. An enhanced oscillator signal in the example design has an average frequency of 456.7 MHz and a maximum combined frequency of 1.15 GHz. Combining more than two enhanced oscillator outputs caused significant loss of sampled transitions because the LUT circuitry was not fast enough to track them. The geometric mean of the number of LUT's per full cycle in the rings of this design is 19.057. This yields a mean jitter of $\sqrt{19.056} \times 59.1 \times 10^{-15}$ = 258 fs rms and a fractional jitter, $J_F = 2.58\times10^{-13}/\ 5.122\times10^{-9} = 5.037\times10^{-5}$ rms. The entropy per single sampled tap is 0.001463, yielding a relative predictability of 0.99980156 and finally, n = 10,786 taps for a target quantum entropy of 0.99. The weighted average number of taps in an enhanced output from a ring in the PQ32MU design is 2.3477, and the number of taps from two enhanced outputs combined is 4.6954. Then 315 of these combined outputs are combined further to produce a raw data stream, with a total of 1479 taps. Three of these raw data streams from three duplicate generators are combined by XORing them to produce a single quantum random bit stream produced from sampling 4,437 taps. Finally, four sequential bits from the combined streams are XORed together to produce output bits at 32 Mbps, each of which was produced from sampling a total of 17,748 original taps. The relative predictability of the output bits, based solely on shot noise is 0.02953 and the predictability is 0.515, giving a quantum entropy of 0.9994 bits per bit; substantially above the design goal of 0.99 bits per bit. The steps for this design are summarized below:

- Oscillator period with 19.056 LUT's per period is 5.122 ns, yielding a mean ring oscillator frequency of 195.2 MHz.
- $J_{LUT} = 5.91\times10^{-14}$ s rms.
- Ring oscillator fractional jitter, $J_F = 5.037\times10^{-5}$ rms.
- Entropy per sampled tap = 0.001463.
- P = 0.99990078 and P_{FI} = 0.99980156.
- $n = Log(0.1176048)/Log(0.99980156)$ = 10,786 samples for $H > 0.99$.
- Weighted average of 2.3477 sample taps per ring enhanced output, times 2 rings (XORed) per sample, times 15 samples per channel, times 21 channels per output stream, times 3 streams, times 4 samples per 128 million samples per second in the combined output stream = 32 MHz output rate composed of 17,748 tap samples per bit.
- $P_{FR} = P_{FI}^{n} = 0.99980156^{17748} = 0.02953$. $P = (0.02953 + 1)/2 = 0.5147$.
- Final quantum entropy in the output stream is, H_Q = 0.9994 bits per bit. For a minimum of two of three redundant streams combined as required by the design,

the quantum entropy is 0.993. This is an emergency backup mode in case of partial generator failure.

- Higher quantum entropy can be achieved at the expense of the final output bit rate. XORing two consecutive non-overlapping bits in the output sequence (a jumper-selectable operating mode) produces a quantum entropy of 0.99999945 bits per bit at a rate of 16 Million bits per second (Mbps).

DETERMINING CHAOTIC ENTROPY IN THE SAME GENERATOR DESIGN

Along with the quantum entropy derived from shot noise, a substantially larger amount of chaotic entropy is also present in each sample. This entropy is due to power supply noise, digital switching noise, other types of transistor noise and thermal noise. Güler and Dündar (Güler & Dündar, 2011) model intrinsic noise sources in ring oscillators constructed from standard 0.35 μm CMOS logic and assume a fractional jitter of 2 percent of the oscillator period. Rather than trying to quantify these various sources from basic principles, it is much easier to directly measure the combined result of all chaotic noise sources. The quantum noise component is much smaller than the total noise so its contribution will not alter the empirical measurement of chaotic entropy sources.

The jitter caused by non-quantum chaotic sources is determined by measuring the entropy at a number of different sampling periods for individual taps in a ring and for the enhanced output of that ring, and finding the jitter that produces the best curve fit to the sampled data that is consistent with the entropy-combining model. The measured entropy is first converted to a predictability, P, by using the inverse entropy calculation. The predictabilities are then converted to relative predictabilities, P_R. The relative predictabilities are then plotted versus the square root of the multiple of the base sample period that produced each data point. This plot is shown in FIG. 6.

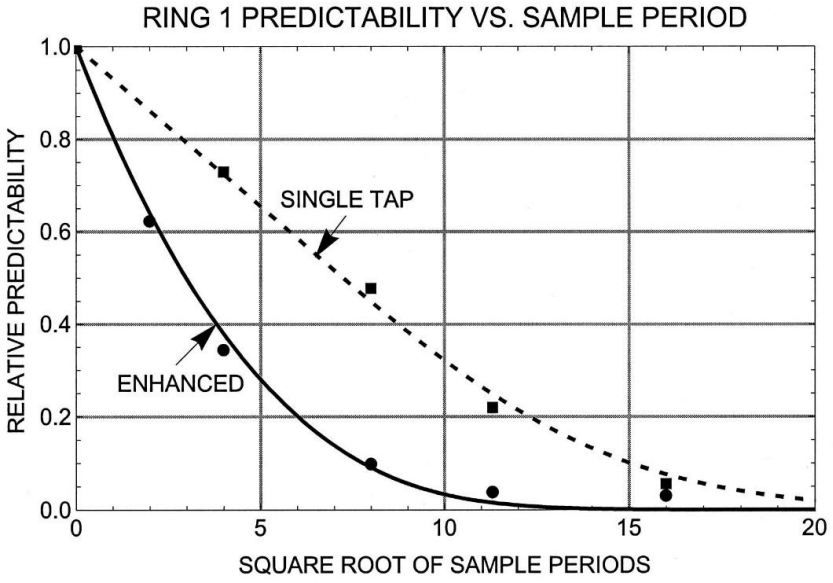


FIG. 6

For this measurement a ring oscillator composed of 12 LUT's with 3 equally spaced taps is used. The base sampling frequency was 128 MHz with a sample period of 7.8125 ns and the ring frequency was 155 MHz. The solid squares represent the measured data for a single tap of the ring and the solid circles represent the enhanced ring output resulting from XORing the three equally spaced taps. According to the model the relative predictability of the enhanced output should be the relative predictability of the single tap cubed. The lines are the curve fits of relative predictability versus the square root of the number of sample periods used to produce the measured data points. By construction the enhanced curve is the cube of the single tap curve. That leaves a single independent variable, the fractional jitter, J_F, which was found to be 0.0197 rms. The curve fit matches the data very well, both with respect to relative predictability versus sample period and the relationship between the single tap and enhanced output, although this type of measurement can typically be very noisy.

The jitter for this 12-LUT ring was also measured directly on an oscilloscope. The ring output was connected to an external test point on the FPGA. The period was 6.55 ns as observed on the oscilloscope for a frequency of 153 MHz. The jitter after 12 cycles from the oscilloscope trigger point was estimated to be 3.5 ns peak-to-peak and the rms value, which is about one-sixth the peak-to-peak value, was 583.3 ps rms. Finally this value was converted to a single cycle jitter by dividing by the square-root of the number of cycles over which it accumulated, yielding 168.4 ps rms per cycle. The per-LUT jitter was 34.4 ps rms and the fractional jitter was 0.026 rms. This "eyeball" estimation is sufficiently close to confirm the effective jitter obtained by curve fitting the more accurately measured data set of Fig. 6.

Now it is possible to calculate the jitter per LUT due to chaotic sources. First multiply 0.0197 rms by the ring period to find the total jitter of 127 ps rms, and then divide this by the square root of 24 to find the jitter for a single LUT, J_{LUT} = 25.9 ps rms. This is over 400 times the size of the jitter due to shot noise alone.

The following steps use the same design and frequency parameters used for the quantum entropy example calculations, except the LUT jitter, J_{LUT}, is the measured chaotic jitter:

- Oscillator period at 19.056 LUT's per period is 5.122×10^{-9} s, yielding a mean ring oscillator frequency of 195.2 MHz.
- $J_{LUT} = 2.59 \times 10^{-11}$ s rms.
- Ring oscillator fractional jitter, $J_F = 0.022$ rms.
- Entropy per sampled tap = 0.257.
- $P = 0.956647$ and $P_{FI} = 0.913293$.
- Weighted average of 2.3477 sample taps per ring enhanced output, times 2 rings (XORed) per sample (Level one sample output)[§], times 15 samples per data

[§] Raw data samples of Levels one, two and three outputs are available for external testing using off-line testing modes in the software provided with *Pure Quantum* generators.

stream = 128 MHz internal (Level two sample output) rate composed of 70.431 mean samples per bit.

- $P_{FR} = P_{FI}^{n}$ = $0.913293^{70.431}$ = 0.00168164. P = (1.68164×10^{-3} +1)/2 = 0.50084082. The chaotic entropy at this internal level is already 0.99999796. This is the last level at which direct statistical testing can be applied to confirm the calculations since the number of bits required becomes too large to achieve at subsequent levels.
- The next internal level (Level three) is the output of one of three redundant generators resulting from XORing 21 Level two outputs. $P_{FR} = P_{FI}^{n}$ = 0.913293^{1479} = 5.5256×10^{-59}. P = (5.5256×10^{-59} +1)/2 = 0.5 + 2.7628×10^{-59}. The entropy at this level is, H = 1– ε where ε = 2.2×10^{-117}.
- The final output of the generator is the result of XORing the 3 Level-three generator outputs and finally XORing 4 non-overlapping consecutive bits to produce each final output bit at 32 Mbps. $P_{FR} = P_{FI}^{n}$ = 0.913293^{17748} = 8.1016×10^{-700}. P = (8.1016×10^{-700} +1)/2 = 0.5 + 4.0508×10^{-700}. The theoretical entropy at the final output is, H_C = 1– ε where ε = 4.7×10^{-1399}.

PURE QUANTUM™ DESIGN: MODEL PQ4000KU

The Model PQ4000KU uses the same generator circuits as the PQ32MU except fewer of them are needed to produce the target entropy of 0.999 at the lower generation rate of 4 Mbps. The design steps are summarized below:

- Oscillator period at 19.056 LUT's per period is 5.122 ×10^{-9} s, yielding a mean ring oscillator frequency of 195.2 MHz.
- J_{LUT} = 5.91×10^{-14} s rms.
- Ring oscillator fractional jitter, J_F = 5.037×10^{-5} rms.
- Entropy per sampled tap = 0.001463.
- P = 0.99990078 and P_{FI} = 0.99980156.
- $n = Log(0.0372286)/Log(0.99980156)$ = 16,582 samples for H > 0.999.
- Weighted average of 2.3477 sample taps per ring enhanced output, times 2 rings (XORed) per sample, times 15 samples per channel, times 5 channels per output stream, times 3 streams, times 32 samples per 128 million samples per second in the combined output stream = 4 MHz output rate composed of 33,806 tap samples per output bit.
- $P_{FR} = P_{FI}^{n}$ = 0.99980156^{33806} = 0.012197. P = (0.012197 +1)/2 = 0.50061.
- Quantum entropy in the final output stream is, H_Q = 0.9999989 bits per bit. For a minimum of two of three redundant streams combined as required by the design,

the quantum entropy is 0.999906. This is an emergency backup mode in case of partial internal failure.

- Higher quantum entropy can be achieved at the expense of the final output bit rate. XORing two consecutive non-overlapping bits in the output sequence (a jumper-selectable operating mode) produces a quantum entropy of, $H_Q = 1 - \varepsilon$ where $\varepsilon = 1.6\times10^{-10}$ bits per bit at rate of 2 Million bits per second (Mbps).

The steps for calculating the chaotic entropy are:

- Oscillator period at 19.056 LUT's per period is 5.122 ns, yielding a mean ring oscillator frequency of 195.2 MHz.
- $J_{LUT} = 2.59\times10^{-11}$ s rms.
- Ring oscillator fractional jitter, $J_F = 0.022$ rms.
- Entropy per sampled tap = 0.257.
- $P = 0.956647$ and $P_{FI} = 0.913293$.
- Weighted average of 2.3477 samples per ring, times 2 rings XORed together per sample (Level one sample output), times 15 samples per data stream = 128 MHz internal (Level two sample output) rate composed of 70.431 mean samples per bit.
- $P_{FR} = P_{FI}^n = 0.913293^{70.431} = 0.00168164$. $P = (1.68164\times10^{-3} + 1)/2 = 0.50084082$. The chaotic entropy at this internal level is already 0.99999796. This is the last level at which direct statistical testing can be applied to confirm the calculations since the number of bits required becomes too large to achieve at subsequent levels.
- The next internal level (Level three) is the output of one of three redundant generators resulting from XORing 5 Level two outputs. $P_{FR} = P_{FI}^n = 0.913293^{352.16} = 1.3442\times10^{-14}$. $P = (1.3442\times10^{-14}+1)/2 = 0.5 + 6.721\times10^{-15}$. The entropy at this level is, $H = 1 - \varepsilon$ where $\varepsilon = 1.3\times10^{-26}$.
- The final output of the generator is the result of XORing the 3 Level-three generator outputs and then XORing 32 non-overlapping consecutive bits to produce each output bit at 4 Mbps. $P_{FR} = P_{FI}^n = 0.913293^{33806} = 2.4323\times10^{-1332}$. $P = (2.4323\times10^{-1332} + 1)/2 = 0.5 + 1.2162\times10^{-1332}$. The theoretical entropy at the final output is, $H_C = 1 - \varepsilon$ where $\varepsilon = 4.3\times10^{-2664}$.

EFFECT OF ERRORS IN QUANTUM NOISE ESTIMATES

The leakage and shot noise values used in the calculations in this paper are estimates based on the information and assumptions described, but clearly more exact numbers could be calculated given complete knowledge of the manufacturer's CMOS IC design. In addition, the simplest models are taken for leakage and shot noise voltage at the CMOS outputs. Errors in any of the estimated parameters will result in an increase or decrease in the actual quantum entropy available, but will not change the methods of

calculating combined entropy of various types or the general design approach for the QRNGs.

Table 3 summarizes the effect on quantum entropy in the output bits when using a wide range of shot noise voltage from a low of $\sqrt{0.5}$ to a high of $\sqrt{2}$ times the nominal value used in the paper.

Shot noise voltage	Low ($\sqrt{0.5}\times$)	High ($\sqrt{2}\times$)
Model PQ32MU	0.9956	0.999974
Model PQ4000KU	0.999957	$1.0 - 2.6\times10^{-9}$

Table 3

The shot noise voltage is inversely proportional to the square root of the load capacitance. Table 4 shows the effect on quantum entropy of varying C_L over a range of 0.5 to 2 times the value used in the paper.

Load Capacitance	Low (½×)	High (2×)
Model PQ32MU	0.999974	0.9956
Model PQ4000KU	$1.0 - 2.6\times10^{-9}$	0.999957

Table 4

Another estimated variable is slew rate of the LUT output, which is calculated from the assumed rise or fall times. Table 5 shows the effect on quantum entropy of varying slew rate over a range of 0.5 to 2 times the value used in the paper, although it is very unlikely the slew rate could ever be as low as one-half the estimated value.

Slew Rate	Low (½×)	High (2×)
Model PQ32MU	0.99999945	0.979
Model PQ4000KU	$1.0 - 1.6\times10^{-14}$	0.99912

Table 5

These three tables show the estimated or indirectly measured variables used to calculate quantum entropy in the output of the two *Pure Quantum* models described in this paper. When these variables are changed over a wide range the design goal for quantum entropy is satisfied in all cases except for the high slew rate of the Model PQ32MU, where it is low by about two percent[**]. The ranges are believed to include worst-case values, but the lower frequency, jumper-selectable modes in either generator can be used to greatly extend the range over which the minimum design quantum entropy level would be produced. As an illustration, the Model PQ32MU set to output 16 Mbps and given the "High" slew rate variable would produce output bits with $H_Q = 0.99937$.

[**] The chaotic entropy maintains the total entropy in the output and ensures effectively perfect statistical properties.

Recently Güler and Dündar (Güler & Dündar, 2012) described theoretically the jitter due to white noise in a ring oscillator implemented with standard 0.25 μm CMOS logic operating at 0.5V. At V_{DD} = 0.5V all the transistors are substantially operating in sub-threshold region at all times. In contrast, the transistors in the LUT's in the *Pure Quantum* generators operate at normal supply voltage, meaning they are in weak or strong inversion at various times, and are in transition at other times. Using their equation (23) for an inverter-based ring, the fractional jitter – *randomness* in their terminology – was calculated with C = 98.5 fF, V_{DD} = 1.2V and a number of stages, M = 12 which gave a fractional jitter of about 4×10^{-4} rms. This compares with the 4.5×10^{-5} rms calculated in this paper for pure shot noise sources using the same variables. Using variables in their paper, the frequency of the inverter-based ring oscillator with V_{DD} = 0.5V is about 1/51.9 = 0.0193 times the frequency with V_{DD} = 2.5V. The entropy per sample at a sample rate of 128 MHz is 0.0018 bits per bit due to the effective jitter adjustment in the jitter to entropy calculation, while the entropy in this paper's example design is 0.00119 bits per bit. The entropy is expected to be somewhat higher for transistors operating in sub-threshold all the time so these numbers calculated using very different approaches are quite close. Güler and Dündar only presented theoretical calculations with no actual measurements, so the delay time and operating frequency of their inverters were estimated for this comparison.

QUANTUM VERSUS CLASSICAL NOISE

Shot noise in a broad sense is inherently quantum mechanical because the inability to make exact predictions of instantaneous current is due to the quantization of the moving charge carriers that embody the current. The uncertainty principle precludes exact determination of each charge carrier's position or motion, resulting in a degree of irreducible noise in predicting future positions and motions. Finally the noise in charge carrier positions translates into non-determinism in voltage on a capacitor, which is sampled to produce random bits.

Formally shot noise can be either classical or quantum mechanical, or a mixture of both. See (Reznikov, De Picciotto, Heiblum, Glattli, A., & Saminadayar, 1998) and (Whitney, 2007) for a discussion of the factors affecting this classification. Qualitatively the noise begins to be quantum mechanical when wave properties of the charge carriers begin to alter the outcome of their measurement, since wave properties of particles are strictly non-classical. Shot noise due to gate direct tunneling leakage and junction tunneling leakage (composed of band-to-band tunneling (BTBT) leakage and trap-assisted tunneling (TAT) currents (Xue, Li, Deng, & Yu, 2010, p. 356)) are taken to be entirely quantum mechanical for purposes of calculating quantum entropy in this paper.

The magnitude of sub-threshold leakage, which is a diffusion process, and the Poissonian statistics of the resulting shot noise to a smaller degree are both affected by quantum mechanical adjustments in MOS transistors of 65 nm and less (Mukhopadhyay, Raychowdhury, & Roy, 2005) and (Jannaty, 2012). Rather than trying to quantify the degree of quantum mechanical versus classical properties of this component of noise, it is

simpler to show the entropy due to shot noise calculated both with and without inclusion of sub-threshold leakage.

Table 6 summarizes the results of calculating quantum entropy using only shot noise resulting from tunneling leakage currents as well as results including both tunneling leakage and sub-threshold leakage.

Pure Quantum Model	PQ32MU		PQ4000KU	
Generator Rate (MHz)	32	16	4	2
Quantum Entropy, H_Q Tunneling Leakage Only	0.964	0.998	0.998	0.999992
Entropy Including Sub-Threshold Leakage	0.9994	0.9999995	0.9999989	$1\text{-}1.6\times10^{-10}$

Table 6

DISCUSSION

Previous quantum random number generators (QRNGs) required quantum measurements in hardware that were complex and expensive, or were not implementable in common integrated circuitry. Furthermore there was not an adequate understanding of how to generate random numbers with a precisely specified or known amount of quantum entropy.

Design equations and specific practical designs[††] for simple, inexpensive yet high quality quantum random generators are presented. The designs target CMOS integrated circuits as their functional platform, but the principles may be applied to random number generators of virtually any design or entropy source. NIST defines "full entropy" as $H = (1 - \varepsilon)$ bits per bit, where $0 \leq \varepsilon \leq 2^{-64}$ (Barker & Kelsey, 2012, p. 4), that is, 5.421×10^{-20}. The generator designs described here not only meet, but vastly surpass that requirement without any type of post processing, conditioning or randomness correction.

Common methods of gathering or concentrating entropy are not usable to significantly increase quantum entropy in most circumstances. A number of approaches for increasing statistical randomness use some type of compression or extraction (Peres, 1992) algorithm to reduce the predictability of a sequence by removing patterns and redundancies in the data. A sequence can be compressed arbitrarily close to an average per-bit entropy of 1.0, but no further (Shannon, 1948), so the data compression ratio, i.e., the fraction of output bits in a compressed sequence divided by the number of input bits, is an approximate measure of statistical entropy of the input bits. Because no algorithm can distinguish or separately compress the quantum entropy, these algorithms do not change the ratio of quantum entropy to other types of entropy. Assuming perfect compression or a Shannon entropy of 1.0 in the output sequence, both the quantum entropy and the chaotic and/or pseudo-entropy are increased by a factor equal to the

[††] Patent pending.

reciprocal of the compression ratio. To illustrate: from one of the design examples, a typical single enhanced output sampled at 128 MHz has 0.003147 quantum entropy bits per bit and 0.45646 bits per bit of total entropy. After compression the total entropy would theoretically be 1.0 bits per bit composed of 0.99531 bits per bit of chaotic entropy and only 0.00689 bits per bit of quantum entropy. Compression- or extraction-type algorithms cannot concentrate the quantum entropy any further.

It should be noted that manufacturers of MOS and CMOS devices and integrated circuits make every effort to reduce leakage and noise any way they can devise. This is required to reduce power consumption and increase reliability of their products, especially as the dimensions of the circuitry are reduced to pack an ever-increasing number of transistors in a given area (Fallah & Pedram, 2005). For the purpose of true random number generation, and especially quantum random number generation, the understanding of the factors affecting leakage can be used to increase rather than decrease the leakage and hence the shot noise in specialized QRNG circuits. The jitter at the output of a CMOS gate is inversely proportional to the slew rate. Therefore decreasing the slew rate without increasing load capacitance will increase the jitter and hence the entropy. Gate leakage is proportional to the area of the gate and inversely proportional to the thickness of the oxide insulating layer under the gate. Increasing the gate area or especially decreasing the insulation thickness will increase the gate leakage current and its related shot noise. Decreasing the channel length or otherwise reducing the size of the threshold voltage of the transistors increases sub-threshold leakage and its shot noise contribution. Other factors such as doping levels, halo profiles and surface area of the junctions strongly affect junction leakage. Several of these factors are easily modified in normal CMOS design to greatly increase total shot noise and hence the quantum entropy available for sampling, although some of these parameters are dependent and cannot be separately optimized for maximum noise production.

Bibliography

Altera Corporation. (2012). Cyclone III Device Family Overview. In *Cyclone III Device Handbook* (Vol. 1, pp. 1-14). Altera Corporation.

Altera Corporation. (2013). *PowerPlay Early Power Estimator*. San Jose: Altera Corporation.

Barker, E., & Kelsey, J. (2012, Aug.). Recommendation for Random Bit Generator (RBG) Constructions. *DRAFT NIST Special Publication 800-90C*, 1-45.

Cao, Y. (2011). Predictive Technology Model for Robust Nanoelectronic Design. In Y. Cao, *The Predictive Technology Model in the Late Silicon Era and Beyond* (pp. 7-23). Springer Science+Business Media.

Coron, J.-S. (1999). On the Security of Random Sources. In H. Imai, & Y. Zeng (Eds.), *Lecture Notes in Computer Science* (Vol. 1560, pp. 29-42). Springer-Verlag.

Davies, R. (2002, Feb. 28). *Exclusive OR (XOR) and hardware random number generators*. Retrieved May 31, 2013, from www.robertnz.net: http://www.robertnz.net/pdf/xor2.pdf

Dichtl, M., Meyer, B., & Seuschek, H. (2008). *SPICE Simulation of a "Provably Secure" True Random Number Generator*. Siemens Corporate Technology. Munich: Siemens AG.

Fallah, F., & Pedram, M. (2005). Standby and Active Leakage Current Control and Minimization in CMOS VLSI Circuits. *IEICE Transactions*, 509-519.

Güler, Ü., & Dündar, G. (2011). Maximizing Randomness in Ring Oscillators for Security Applications. *European Conference on Circuit Theory and Design* (pp. 118-121). Linkoping: ECCTD.

Güler, Ü., & Dündar, G. (2012). Modeling Phase Noise and Jitter in Subthreshold Region and Assessing the Randomness Performance of CMOS Ring Oscillators. *International conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design* (pp. 257-260). Seville: SMACD.

Gupta, T., & Khare, K. (2013, Jan.). A New Dual-Threshold Technique for Leakage Reduction in 65nm footerless Domino Circuits. *International Journal of Computer Applications, 61*(5), pp. 14-20.

Jannaty, P. (2012, May). Shot noise in nanoscale MOSFETs. *Low-voltage End-of-Roadmap Transistors and their Reliability in the Presence of Noise, Section 4.2*, pp 84-95. RI: Brown University.

Korkmaz, P. (2005). *Modeling the Short-Circuit Energy Dissipation of a CMOS Inverter*. Technical Report, Rice University, Computer Science.

Liu, Z., & Kursun, V. (2007, Dec.). PMOS-Only Sleep Switch Dual-Threshold Voltage Domino Logic in Sub-65-nm CMOS Technologies. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 15*(12), pp. 1311-1319.

McNeill, J., & Ricketts, D. (2009). Sources of Jitter in Ring Oscillators. In *The Designer's Guide to Jitter in Ring Oscillators* (pp. 161-229). Springer US.

Mukhopadhyay, S., Raychowdhury, A., & Roy, K. (2005, March). Accurate Estimation of Total Leakage in Nanometer-Scale Bulk CMOS Circuits Based on Device Geometry and Doping Profile. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 24*(3), pp. 363-381.

Peres, Y. (1992). Iterating von Neumann's Procedure for Extracting Random Bits. *The Annals of Statistics, 20*(1), 590-597.

Reznikov, M., De Picciotto, R., Heiblum, M., Glattli, D., A., K., & Saminadayar, L. (1998). Quantum shot noise. *Superlattices and Microstructures, 23*(3/4), pp. 901-915.

Ruggeri, N. (2006, Aug. 26). *Principles of Pseudo-Random Number Generation in Cryptography*. Retrieved June 22, 2013, from http://www.math.uchicago.edu: http://www.math.uchicago.edu/~may/VIGRE/VIGRE2006/PAPERS/Ruggeri.pdf

Santha, M., & Vazirani, U. (1984). Generating Quasi-Random Sequences from Slightly-Random Sources. *Proceedings of the 25th Annual Symposium on Foundations of Computer Science* (pp. 434-440). IEEE computer Society.

Sarpeshkar, R., Delbrück, T., & Mead, C. (1993). White Noise in MOS Transistors and Resistors. *IEEE Circuits and Devices Magazine*, *9*(6), 23-29.

Shannon, C. (1948, Jul.; Oct.). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*, 379-423; 623-656.

Stipčević, M. (2011). Quantum random number generators and their use in cryptography. *MIPRO Proceedings of the 34th International Convention*, (pp. 1471-1479). Opatija, Croatia.

Sunar, B., Martin, W., & Stinson, D. (2007, Jan.). A Provably Secure True Randon Number Generator with Built-in Tolerance to Active Attacks. *IEEE Transactions on Computers*, pp. 109-119.

Whitney, R. (2007). Shot-noise of quantum chaotic systems in the classical limit. *Proceedings of SPIE 6600, 66000R*, (pp. 1-9).

Wilber, S. (2005, Mar.). *Patent No. 6,862,605 B2*. U.S.

Xue, J., Li, T., Deng, Y., & Yu, Z. (2010). Full-chip leakage analysis for 65 nm CMOS technology and beyond. *INTEGRATION, the VLSI journal, 43*, pp. 353-364.

Zhang, Y., Parikh, D., Sankaranarayanan, K., Skadron, K., & Stan, M. (2003). HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects. *Univ. of Virginia Dept. of Computer Science Tech. Report CS-2003-05*, 1-15.