Calculating Entropy Available to the PCQNG 2.0

The following is abstracted from one of the patents covering the PCQNG. This includes a detailed discussion of the method of calculating the entropy available from two oscillatory signals containing independent noise sources.

A circuit for generating true random numbers using high- and low-frequency oscillators is described in detail in *An LSI Random Number Generator (RNG)*, Proc. Advances in Cryptology, Conference on CRYPTO, 1984, by Fairfield, Mortenson and Coulthart. The circuit described therein uses a noise component in the low-frequency signal as the source of entropy.

Some authors have suggested methods of generating true random numbers using components normally available on personal computers. These include deriving entropy from keyboard stroke timing, computer mouse movements and air turbulence in the hard disk drive. Some discussion of such methods is given, for example, in *Randomness Recommendations for Security*, 1994, by Eastlake, Crocker and Schiller, a web site at http://www.ietf.org/rfc/rfc1750.txt. While these methods may be a source of entropy, the generation rates are very low and intermittent, and many require the interaction of a human operator.

All true random number generators require a physical source of entropy to produce random numbers, as distinct from algorithmically generated pseudorandom numbers, which are deterministic by nature of their source. The requirement for a physical generator has previously limited the availability of high quality, true random numbers due to cost, size, power requirements or difficulty in interfacing with the user's equipment.

A significant limitation of prior true random number generators is the uncertainty in the actual entropy and quality of the random numbers produced. Direct testing of a random sequence does not assure universally acceptable results in all applications as different types of defects may not show up in a certain set of tests. Also, direct testing may be impractical due to the large number of bits, hence time, required to test to the required level of significance.

Systems for scrambling bits or correcting defects in random number sequences are also known in the art. Examples of these are contained in U.S. Pat. Nos. 5,781,458, 5,963,104 and 6,061,702. Randomness defect correction can also be accomplished by many encryption methods, such as DES.

It is important to know that randomness defect correction or bit scrambling does not in any way add true entropy to the output sequence. Only the actual number of bits of true entropy input to any deterministic algorithm can be taken as true random output from such an algorithm.

SUMMARY OF THE INVENTION

The device and method of the present invention generates true random number sequences of calculable entropy content. The entropy is derived from a random noise component in one or

both of a low- and a high-frequency signal source that are coupled to a processing means for producing the random numbers. The high-frequency signal source includes a frequency multiplier that increases the size of the noise component in the high-frequency signal.

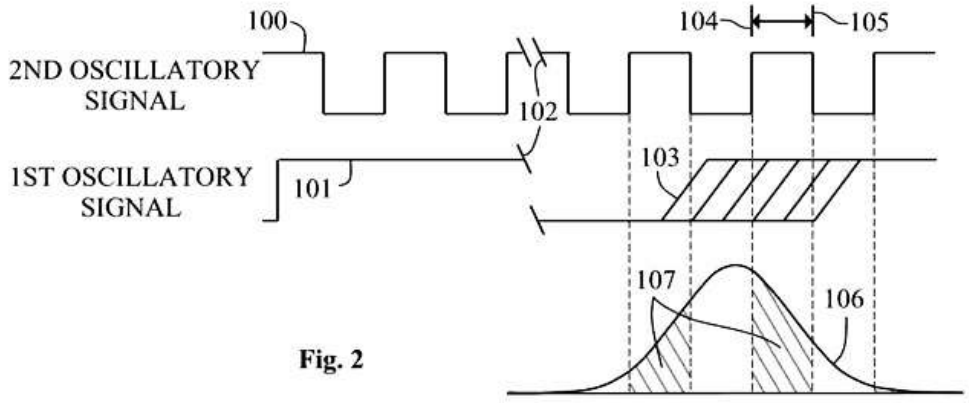

Fig. 2

FIG. 2 is a diagram showing oscillatory signals plus transition jitter to illustrate the method of calculating entropy of the present invention

A perfect sequence of true random numbers contains exactly one bit of entropy for each bit in the sequence. If the entropy content is less than one bit per bit, the sequence is not a true random sequence, but a mixture of true and pseudorandom components.

An understanding of the entropy source process, including an accurate and general mathematical model, is required in order to know the actual entropy content of a random number sequence embodying that entropy.

The entropy used in the present invention is produced by a combination of sources such as thermal noise in resistive elements, shot noise in transistors and diodes, power supply sensitive nonlinear amplifiers such as logic gates, noise that arises in frequency multipliers, and noise deriving from the bandwidth or finite Q-factor in crystal resonators. These physical noise sources produce a transition jitter in each transition of the first and second oscillatory signals, **101** and **100** respectively. This jitter manifests as a cycle-to-cycle variation in the periods of the oscillatory signals.

The oscillatory signals of FIG. 2 are shown in the form of square waves, but this is for illustrative purposes only. One of the oscillatory signals, in this example, the second oscillatory signal, is used for timing and is usually of a periodic nature. The first oscillatory signal may be derived from a number of possible signal sources including thermal noise, shot noise, quantum mechanical noise of various types and chaotic noise such as found in air turbulence. The noise source is then converted to a binary form for processing, but the sometimes huge cycle-to-cycle variations in this binary representation would not appear to be periodic at all.

The first step needed to calculate the entropy from the two oscillatory signals is to determine the value of transition jitter in each of the signals. This is done by a combination of direct

measurements and theoretical modeling depending on the particular sources of the oscillatory signal. Nuclear decay produces an easily measurable average rate and a theoretically known statistical distribution of decay timing. Crystal oscillators have cycle-to-cycle jitter characteristics that are measured by special instruments designed for that purpose. The distribution of a crystal oscillator jitter is nearly Gaussian, and the jitter values for particular crystals can be obtained from, or calculated from information in the manufacturers' specifications sheets. In particular, the jitter specifications for the output of oscillator/clock synthesizers used in personal computers are measured by the manufacturers and contained in the relevant specification sheets.

The second step is combining the independent transition values into one effective value. It must be noted that the total transition jitter between two signals is relative, that is, the jitter in both signals may be combined and represented as only existing in one of the signals, without the loss of generality or accuracy in the results. For reasons of simplicity and consistency, the total or effective jitter will always be represented as a component of the lower-frequency, in this example, the first oscillatory signal, **101**. The jitter values must first be properly adjusted before they can be combined. For Gaussian signals, this is done by multiplying the jitter value in the higher-frequency oscillatory signal, in this example, the second oscillatory signal, **100,** by the square root ratio of the average frequency of the second oscillatory divided by the average frequency of the first oscillatory signal. The adjusted jitter value can be combined with the first jitter value by adding in quadrature – that is, squaring each value and taking the square root of their sum.

If the adjusted jitter value of the second oscillatory is relatively small compared to that of the first oscillatory signal, the jitter of the second oscillatory signal may be omitted from the effective jitter value. A second jitter value that is 20% of a first jitter value will contribute only about 2% to the effective value. Assuming a zero value for the second oscillatory signal can greatly simplify the entropy calculations, especially when the statistical distributions of the two oscillatory signals are different. This simplification will cause the calculated entropy value to be slightly lower than the actual entropy.

The next step is simply to divide the effective jitter value by the period of the second oscillatory signal to produce a normalized jitter value. This step simplifies the final calculations. The normalized jitter is shown in the figure as diagonal lines, **103**, representing the indeterminacy of the transition time of the first oscillatory signal. The breaks in each signal, **102**, indicate that not all cycles of each signal are shown in the figure since there may be thousands of cycles in the second oscillatory signal for each period of the first oscillatory signal.

The next step is to calculate the average probability of correctly predicting that the second oscillatory signal will be in the high state, or 1, at a positive transition of the first oscillatory signal. A single calculation of the probability of correctly predicting a high value is made by integrating across the probability distribution function (PDF), **106**, of the effective jitter in the first oscillatory signal in all periods where the second oscillatory signal is in the high state. The hatched areas, **107**, in the PDF, represent these periods. The PDF illustrated in the figure is Gaussian; however, the actual distribution function representing the statistics of the jitter in the

first oscillatory signal is to be used. The center of the PDF is to be placed at the expected positive transition time of the second oscillatory signal.

The average probability of correctly predicting a high value, p(1) may be calculated numerically by averaging the results of a large number of single probability values calculated by uniformly distributing the placement of the center of the PDF across a single positive half-cycle of the second oscillatory signal starting at the rising edge, **104**, and ending at the falling edge, **105**. One skilled in the art of mathematics can readily determine the minimum number of single probability values necessary to achieve a particular accuracy.

The final step is to utilize the average probability, p(1), to calculate the available entropy. To state the equation in its simplest form, note that the average probability of correctly predicting a low state or 0 is simply $p(0) = 1.0 - p(1)$. These two probabilities are used in Shannon's equation for entropy, H:

$H = -(1.0/\text{Ln } 2.0)(p(1) \text{ Ln } p(1) + p(0) \text{ Ln } p(0))$, where Ln represents the natural log.

The optimum value of the duty cycle of the second oscillatory signal is 50%. Any increase or decrease in this value affects the average probability, p(1), producing a 1/0 bias in the output sequence and reducing the available entropy.

The method for calculating available entropy will be further illustrated by means of a specific example: In the following example a crystal oscillator operating at 14.318MHz supplies a second oscillatory signal. The cycle-to-cycle root mean square (rms) jitter of this oscillator is about .001 times its period, or 70ps rms. This value is taken from published articles describing direct measurements made with special equipment designed to measure the noise spectrum in such oscillators, whereby the jitter can be calculated. A resistor/capacitor, or RC, CMOS oscillator produces a first oscillatory signal with a frequency of 1.02KHz. The jitter in this oscillator may be measured with a readily available spectrum analyzer and is found to be $7x10^{-6}$ times the period of the RC oscillator, or about 7ns. The rms jitter of the RC oscillator can also be measured by observing the peak-peak jitter on an oscilloscope. The rms jitter is about one-sixth to one-seventh of the peak-peak jitter.

The jitter in the crystal oscillator is multiplied by the square root of 14318/1.02, which gives an adjusted jitter value of 8.3ns rms. The next step is to combine the rms jitter of both oscillators. This is done by squaring each component, adding and taking the square root of the sum. The resulting effective jitter is 10.86ns. Dividing by the period of the HF oscillator normalizes the effective jitter, which in this example is 69.8ns. The result of this normalization step is the dimensionless ratio, 0.156. The distributions of both oscillators is approximately Gaussian yielding a Gaussian distributed effective jitter on the first oscillatory signal with a standard deviation of 0.156 of the second oscillatory signal period.

The next step is to calculate the average probability, p(1). The following Mathematica program will illustrate this approach wherein: prob gives the probability of correctly predicting a single transition of the first oscillatory signal at a particular phase, mu, in the second oscillatory signal

cycle using a given normalized jitter, rho; and avgprob numerically calculates the average of the prob function across one positive half cycle of the second oscillatory signal.

```
prob[mu_, rho_]:=Sum[CDF[NormalDistribution[mu, rho], x+.5] -CDF[NormalDistribution[mu, rho], x], {x , -1. Round[6 rho], 1. Round[6 rho]}]
avgprob[rho_]:=If [rho > . 9, .5, N[2. Sum[prob[mu, rho], {mu, 0., .5, 1./1000.}]/501.- Sum[prob [mu, rho], {mu, 0., .5, 1./500.}]/251.]]
```

The result of this calculation is p(1) = 0.75, and therefore, p(0) = 0.25.

The final step is to use the calculated average probabilities, p(1) and p(0), to calculate the actual entropy, H, as illustrated in the Mathematica program:

```
H[p1_, p0_]:= (-1. /Log[2.]) (p1 Log[p1]+p0 Log[p0])
```

The result of the entropy calculation wherein p1 = p(1) and p0 = p(0), is H = 0.81 bits. This means that a random number generator composed of a counter being clocked by the second oscillator with the counter's least significant bit (LSB) being latched as output bits by the first oscillator's positive transitions will produce a sequence with an average entropy of 0.81 bits per output bit. This 0.81 bits is not the total available entropy; each bit in the counter of this example also contains entropy. The amount of entropy in each bit can be calculated by dividing the normalized jitter value by 2 to the power n, where n starts at 0 for the LSB and increases by one for each succeeding bit. The second bit in the counter produces a sequence of bits with entropy calculated from a normalized jitter of 0.165/2. = 0.0825, yielding an entropy value of 0.56 bits. Taking the entropy from each of the lower six bits of the counter gives a total available entropy of 2.06 bits. These bits must be subsequently processed to extract the entropy to produce a sequence of true random numbers with entropy of one bit per output bit.

The method of calculating the average probability, p(1), relies on the assumption that the ratio of the second oscillator frequency divided by the first oscillator frequency is an irrational number, or at least not an integer or a rational fraction containing small integers. This means that the rising edge of the first oscillatory signal will be nearly uniformly distributed across the cycles of the second oscillatory signal during the course of many cycles of the first oscillatory signal. If this condition is not met, the available entropy, and any entropy available in bits higher than the LSB may be significantly reduced. It is therefore desirable to select frequencies of the two oscillatory signals that give a ratio as far as possible from an integer or a rational fraction containing small integers.