# Fibonacci Ring Oscillators as True Random Number Generators - A Security Risk

Markus Dichtl*

Siemens Corporate Technology

**Abstract.** Fibonacci ring oscillators are shown to have a risk to oscillate periodically instead of chaotically. The security implications of this are discussed. The probability of the occurrence of the periodic oscillations is determined experimentally on an FPGA for Fibonacci ring oscillators of lengths 16 and 32. Means to overcome the problem of the periodic oscillations are also discussed.

**Keywords: ring oscillator, Fibonacci ring oscillator, true random number generator**

## 1 Introduction

In [Gol06], Jovan Dj. Golić introduced a very broad class of true random number generators. [Gol06] contains two concrete examples of this class, namely Fibonacci and Galois ring oscillators (FIROs and GAROs). [DG07] gave more detailed results about Fibonacci and Galois ring oscillators.

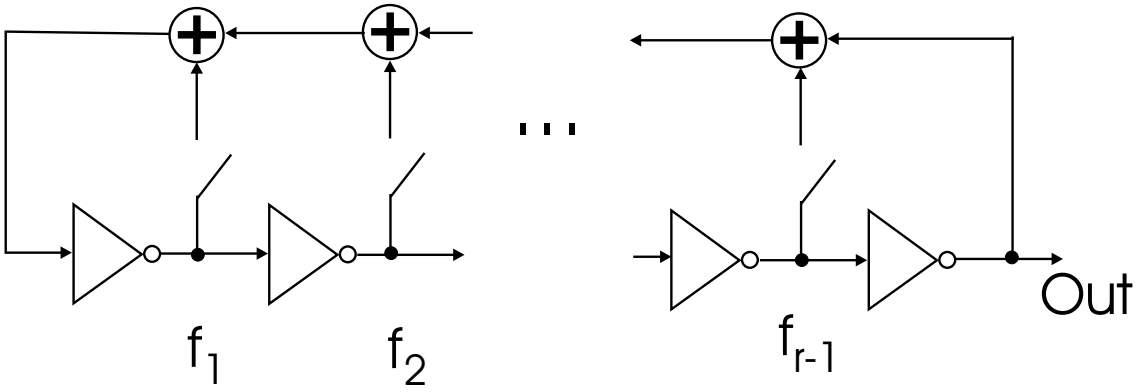


**Fig. 1.** Schematic of Fibonacci ring oscillators. The circles with"+"-symbols stand for XOR gates. The switches shown do not change their state during the operation of the FIRO. Their on- or off-state is determined by the feedback polynomial of the FIRO.

The schematic of Fibonacci ring oscillators is shown in Fig. 1.

[DG07] already gives a hint on the security risk of using Fibonacci ring oscillators to generate true random numbers. There, it is mentioned that very short Fibonacci and Galois ring oscillators occasionally show periodic oscillations. Another source for doubts on the proper operation of Fibonacci and Galois ring oscillators is [MJ11]. The authors of this paper deplore that there is no theory explaining how to choose feedback polynomials for Fibonacci and Galois ring oscillators which work properly.

In order to produce orders of magnitude more entropy per time than classical ring oscillators, Fibonacci and Galois ring oscillators have to oscillate chaotically, as described in [Gol06] and

* Markus.Dichtl@siemens.com

[DG07]. This allows Fibonacci and Galois ring oscillators to be sampled at rates several orders of magnitude faster than classical ring oscillators. When the chaotic oscillations do not occur, but the oscillations are periodic, the sampling rate designed for the chaotic oscillations is much too high. As a consequence, the random bits generated have very low entropy. This implies a considerable security risk.

[sSNA12] gives an example of a FIRO which keeps changing between chaotic and periodic oscillations.

The purpose of this paper is to provide a quantitative assessment of the risk of Fibonacci ring oscillators to fail by oscillating periodically instead of chaotically. Section 7 introduces a new method to visualize what is going on in a FIRO. In addition, means to overcome the problem of the periodic oscillations are discussed.

## 2 Experimental environment

All the experimental results described in this paper were achieved on a board based on an FPGA Spartan XCS31000 from Xilinx.

## 3 How to distinguish chaotic from periodic oscillations

In order to give quantitative results on the risk of periodic Fibonacci ring oscillator behaviour, one has to be able to distinguish chaotic from periodic oscillations. Theoretically, this is an extremely difficult task. The periods of the periodic oscillations could be very long, and it seems quite impossible to detect periods of, say, several minutes. We do not know whether such long periods exist for Fibonacci ring oscillators, but there is experimental evidence for very simple periodic behaviour of Fibonacci ring oscillators with short periods.

The easiest way to distinguish simple periodic oscillations from chaotic ones is to record the output voltage of an inverter of the Fibonacci ring oscillator with an oscilloscope, and to judge the periodic or chaotic behaviour by looking at the oscillograms.

Fig. 2 shows chaotic oscillations of a Fibonacci ring oscillator of length 16.

In contrast, Fig. 3 shows a very simple periodic behaviour of a Fibonacci ring oscillator of length 16.

However, as the experiments of this paper were executed on FPGAs, a method which allows the distinction of periodic from chaotic oscillations directly on the FPGA is preferable.

In order to gain as much digital information about the state of the Fibonacci ring oscillator as possible, the outputs of all the inverters in the ring are sampled simultaneously, and this simultaneous sampling is repeated periodically.

When the Fibonacci ring oscillator shows a simple periodic oscillation with a short period, the state of the ring oscillator as defined by the outputs all $n$ inverters in the ring repeats after a short time. So, when sampling these states periodically, only a small number of different samples is observed even in a large number of samples. For the feedback polynomial $x^{16}+x^7+x^2+1$ (Fig. 3), only 63 different sample values were obtained from sampling the 16 inverters 10000 times with 600 ns between individual samples.

In contrast, chaotic oscillations result in a very large number of different $n$ bit samples from sampling all the outputs of the $n$ inverters simultaneously. Typically, a considerable fraction of the possible $2^n$ samples is observed, if a high enough number of $n$-bit-samples is taken. Among the 10000 16-bit-samples from the feedback polynomial $x^{16}+x^6+x^3+1$ (see Fig. 2), 3442 different 16-bit-patterns occurred.

However, it should be noted that the probabilities of the different $n$-bit samples vary considerably. Again, the samples from the chaotic oscillations with the feedback polynomial $x^{16}+x^6+x^3+1$ were considered. To achieve more reliable statistics, 2 million 16-bit-samples were evaluated. There were 26245 different samples, 5407 of them occurred only once. The most frequent sample occurred 11449 times. Fig. 4 shows the sorted probabilities of the 26245 bit patterns.


**Fig. 2.** Oscillogram of a chaotic FIRO oscillation. The output voltage of one of the inverters in the FIRO is shown for a time of 500ns. The feedback polynomial was $x^{16} + x^6 + x^3 + 1$.

## 4 Systematically trying fixed point free Fibonacci ring oscillators of length 16

### 4.1 The experimental setup

As the main example of a Fibonacci ring oscillator of [DG07] is of length 16, a systematic survey of fixed point free Fibonacci ring oscillators of this length was performed.

How many fixed point free FIROs are there? There are 16 inverters in the FIRO, and the output of each of them may be fed back or not. But as we want only ring oscillators of length 16, we always have to feed back the output of the 16th inverter to the first one (the inverters are numbered from 1 to 16, the output of inverter $i$ being used as the input of inverter $i + 1$ for $1 \leq i < 16$).

In addition, fixed points have to be avoided. For FIROs, there are two kinds of potential fixed points. For one kind, the XOR sum of all the feedback tabs is logical 1, for the other kind, it is logical 0.

When the fed back XOR sum in the fixed point state of the FIRO is 0, it follows that the outputs of all the inverters at odd positions are 1. The inverters at the even positions have outputs of 0, which do not contribute to the feedback XOR sum, whether they are fed back or not. So, in order to really have a feedback XOR sum of 0 for a fixed point, there must be an even number of outputs from odd inverter positions contributing to the XOR sum. Conversely, to avoid this kind of fixed point, the number of fed back outputs from the inverters at odd positions must be odd.

For the other kind of fixed point, the fed back XOR sum is 1. Hence, the outputs of the inverters at odd positions are 0; they cannot contribute to the fed back XOR sum. The outputs

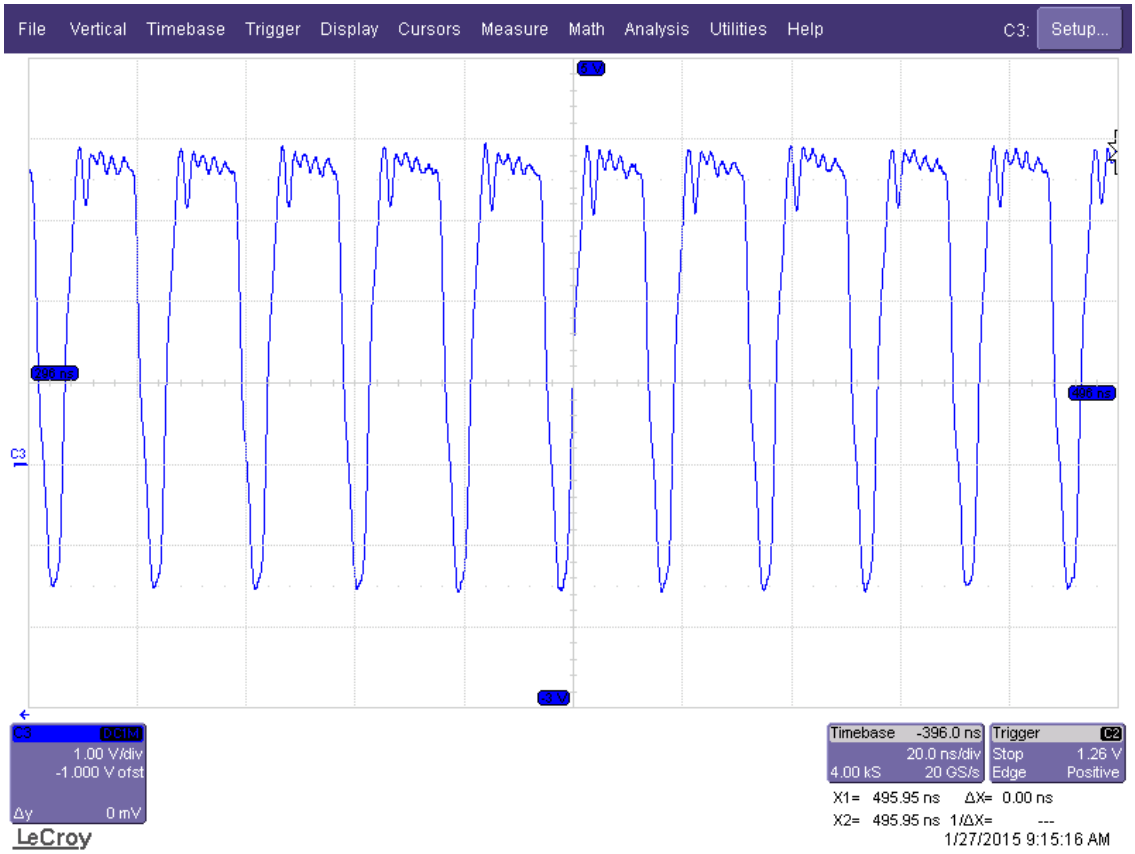


**Fig. 3.** Oscillogram of a periodic FIRO oscillation. The output voltage of one of the inverters in the FIRO is shown for a time of 200ns. The feedback polynomial was $x^{16} + x^7 + x^2 + 1$

of the inverters at even positions are 1; the numbers of them contributing to the XOR sum must be odd in order to result in the XOR sum of 1 required for this kind of fixed point. Conversely, there must be an even number of feedbacks from even inverter positions contributing to the fed back XOR sum in order to avoid this kind of fixed points.

As, for length 16, the feedback from the 16th inverter to the first one is always needed, we have $2^{15}$ possibilities to form the fed back XOR sum. This number is reduced by a factor of 4 if both kinds of fixed points have to be avoided. So, there is a total of $2^{13} = 8192$ fixed point free FIROs of length 16.

Earlier experience showed that feedback loops including only a single inverter can sometimes be problematic, so feedback from the first inverter to the XOR sum was excluded from the feedback configuration. This results in a total of 4096 feedback configurations to test.

For each of these feedback configurations, the FIRO was restarted from a static condition, in which the inverters at odd positions had 0 outputs and those at even positions had an initial output value of 1.

For each of the 4096 feedback configurations, the oscillating FIRO was sampled 10000 times. The time between subsequent samplings was 600 ns. The time between the start of the oscillation and the first sample was also 600 ns.

### 4.2 Experimental results and their interpretation

Fig. 5 shows the sorted numbers of different samples for all the FIRO configurations considered.

One would expect to see a clear dichotomy of results, on one hand the simple periodic oscillations with a small number of different samples, and on the other hand, the chaotic oscillations

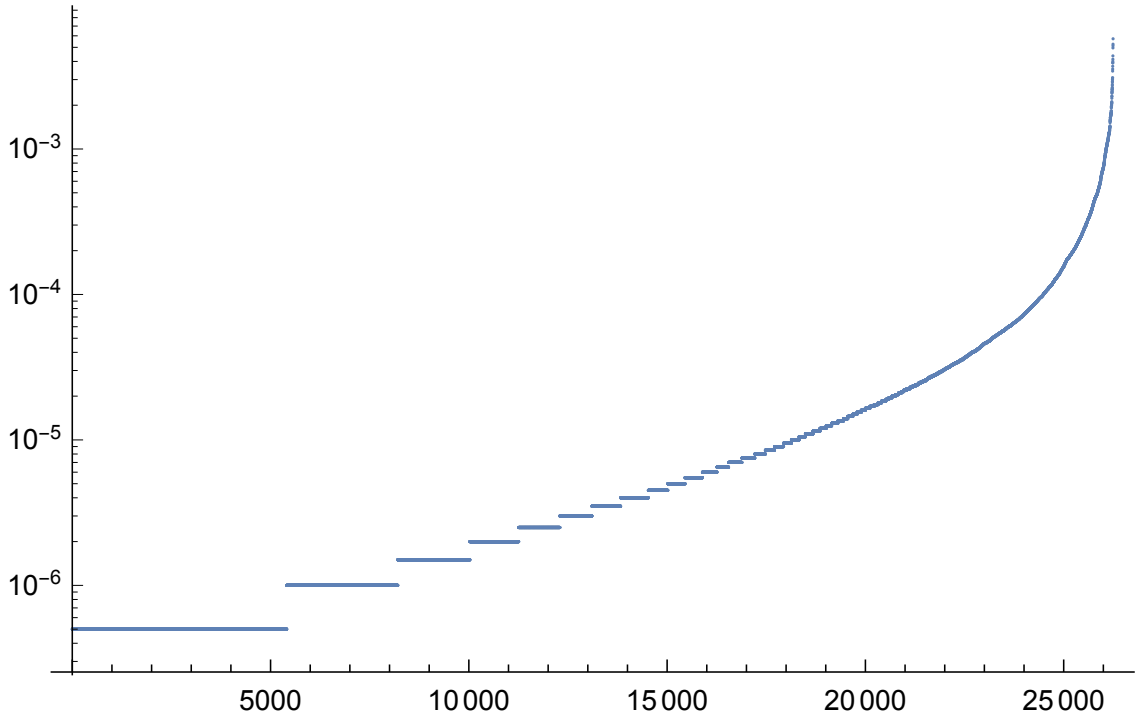


**Fig. 4.** Sorted probabilities (logarithmic ordinates) of 2 million samples from a chaotic FIRO oscillation. The feedback polynomial was again $x^{16} + x^6 + x^3 + 1$.

with several thousands of different samples among the 10000 samples. However, there is no clear divide. The number of intermediate cases is not very big, but by no means negligible. It seems quite impossible to determine the cause of the intermediate values from Fig. 5. So, we defer the interpretation of the intermediate results to Section 7.

Very clearly, a FIRO which assumes 100 or less different values among 10000 samples is not oscillating chaotically, and produces little entropy. Among the 4096 FIROs of length 16 tried experimentally, 100 or less different values were assumed in 542 cases, or 13.23 %.

## 5 Are the failing FIROs due to ignoring a required criterion?

In [Gol06], criteria for the feedback polynomial for FIROs were given. There, the polynomial was required to be of the form $(x+1)f(x)$ with $f(x)$ being a primitive polynomial for which $f(1) = 1$ holds. In the experiments described above, the primitivity of $f(x)$ was complelety ignored. In order to verify whether ignoring this criterion led to the failure of the 542 FIRO configurations, it was verified how many of the 4096 FIROS of length 16 tested with feedback polynomials of the form $(x+1)f(x)$ with primitive $f(x)$ led to 100 or less different values among the 10000 samples. 1091 of the 4096 FIROs met the primitivity requirement.

Fig. 6 shows the sorted numbers of different 16-bit-samples among the 10000 samples for each of the FIROs meeting the primitivity requirement. Fig. 6 looks very similar to Fig. 5, there seem to be no larger differences from the distribution observed in all the 4096 FIROs.

151 of the 1091 FIROs meeting the primitivity criterion had 100 or less different values among the 10000 samples. This is a percentage of 13.84.

Now, it seems unwise to conclude that meeting the primitivity criterion increases the risk of failing FIROs. We consider a binomial distribution with 1091 trials and a probability of success of 542/4096 (here, a success is a failure of the FIRO). For this distribution, the probability of 151 or more successes is 0.289. So, the observation of 151 failing FIROs among the 1091 ones meeting

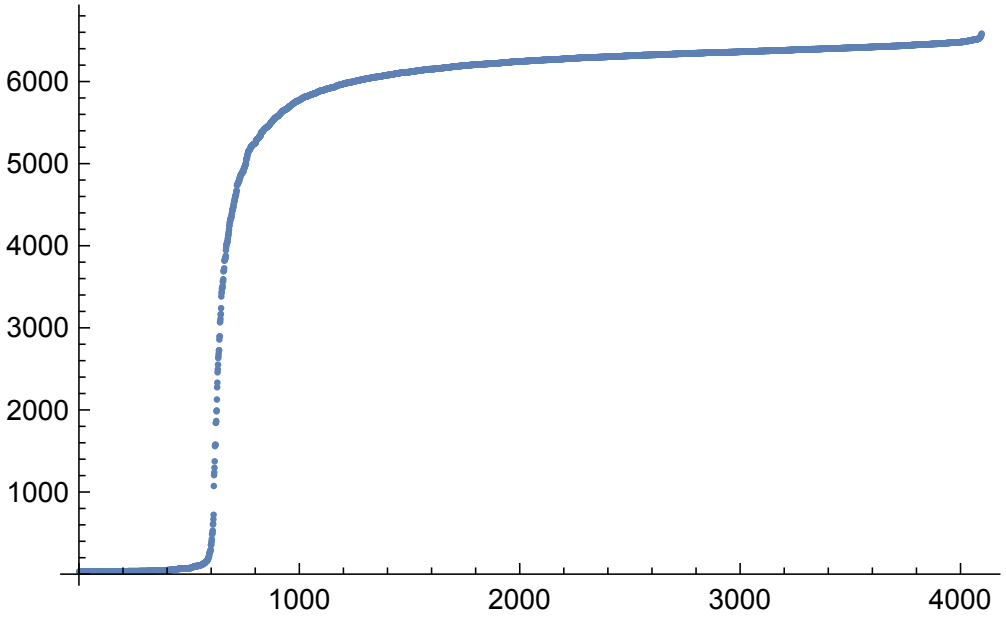


**Fig. 5.** The sorted numbers of different samples among the 10000 samples for each of the 4096 FIRO configurations of length 16.

the primitivity criterion is very well compatible with the assumption that the primitivity criterion is irrelevant for the success or failure of the FIRO.

To understand why the primitivity criterion is just irrelevant for the oscillatory properties of the FIRO, we have to consider how it was derived in [Gol06]. There, the behaviour of the FIRO is treated in a synchronous model, in which the computation of the XOR sum of the feedback and the transition of the state of the inverters to their next state take exactly the same amount of time. This model is extremely inadequate for FIROs implemented on ASICs, as there, the computation of the XOR of two signals is about 10 times as slow as the logical negation of a signal. But the synchronous model is also not adequate for FPGAs, as used in the experiments described above. As all logical functions are implemented as lookup tables (LUTs) on the FPGA, one might assume that all functions take the same time to be evaluated. However, there are considerable timing differences for the individual functions to be evaluated, due to varying routing. The time to compute the XOR sum increases to multiple LUT delay times as soon as there are more feedback tabs involved than the maximal LUT input size of the technology used. In any case, the synchronous model used to derive the primitivity criterion for the feedback polynomial is highly inadequate for the practical implementations of FIROs. So, it is not surprising that the failure rate of the FIROs turns out to be about the same, whether the primitivity criterion is met or not.

## 6 FIROs of length 32

How does the problem of the failing FIROs depend on their length? As the primitivity criterion was found to be irrelevant, for the FIROs of length 32 1000 random fixed point free feedback configurations meeting the Golić irreducibility criterion from [Gol06] were chosen. For each of the 1000 FIROs, 1000 32-bit-samples were acquired. The time between subsequent samples was again 600 ns.

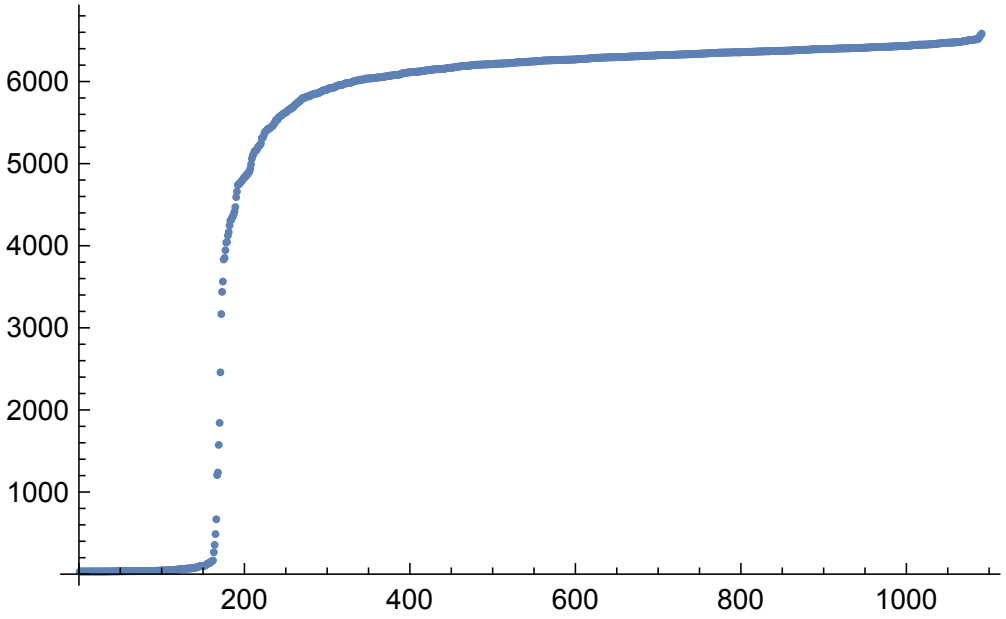


**Fig. 6.** The sorted number of different values among the 10000 samples for the 1091 FIRO configurations of length 16 which meet the primitivity requirement from [Gol06].

Fig. 7 shows the distribution of the numbers of different bit patterns among the 1000 samples. The large majority of the FIROs of length 32 works properly: among the 1000 samples, only very few patterns occur repeatedly. But in a low percentage of cases, there are many repetitions of bit patterns cuased by periodic oscillations of the FIROs. It is difficult to compare theses results to those of the FIROs of length 16, as both the number of samples were different, and the probabilities for repeatedly appearing 32-bit-patterns and 16-bit-patterns are substantially different. For the 1000 32-bit-samples, 67 of the 1000 feedback configurations resulted in 100 or less different patterns. This percentage of 6.7 cannot be compared directly to the 13.2 % of failing ROs of length 16 pointed out above. But it is clear that FIROs of length 32 do not solve the problem of failing FIROs. A failure rate of 6.7 % is very far away from an acceptable failure rate for serious security applications.

## 7 The FIROs of length 16 with intermediate numbers of different samples

One might speculate that the FIROs of length 16 which show an intermediate number of different samples among the 10000 samples have periodic oscillations with a longer period and complicated behaviour within the period. But it seems a good idea to have a closer look at the experimental data instead of speculating.

A way to get more insights into the complex behaviour of FIROs is to use the 16-bit-samples from all of the inverters, and to consider, how many samples ago the same 16-bit-pattern has been sampled.

In Fig. 8, the behaviour of a FIRO with the feedback polynomial $x^{16} + x^{14} + x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^4 + x^2 + 1$ is shown, which was sampled 10000 times, with a sampling period of

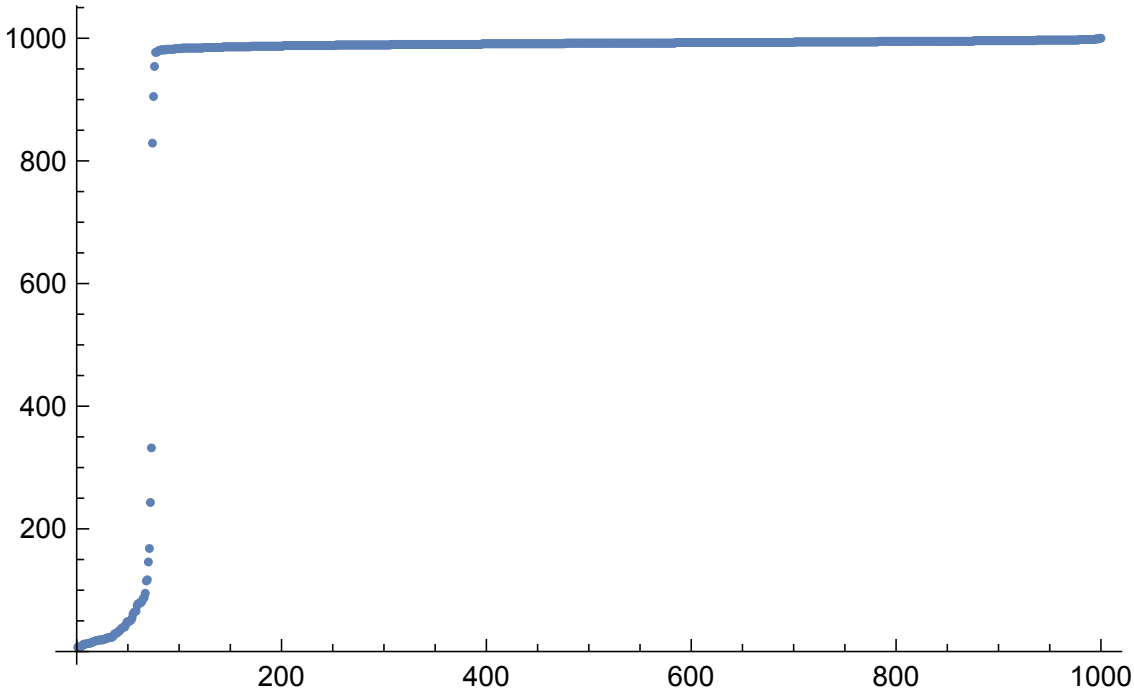


**Fig. 7.** The sorted number of different samples among the 1000 32-bit-samples for the 1000 configurations for FIROs of length 32 (all meeting the primitivity requirement from [Gol06]).

600 ns. In the first 1919 samples, the FIRO oscillated chaotically. Among the first 1919 samples, 1706 different bit patterns occurred. It should be noted that in this sample range, there are very few points in Fig. 8, as the vast majority of bit patterns appear for the first time. At the 1920th sample, the behaviour of the FIRO changes drastically: its oscillations become periodic. In the 8081 samples from the 1920th sample to the 10000th, only 32 different bit patterns occurred. The FIRO had a period of approximately 600 ns divided by a positive integer, as in 2437 cases of the 8081, the 16-bit-pattern sampled was identical to the previous one, sampled 600 ns before. As the FIRO period was not exactly 600 ns divided by a positive integer, there was a beating pattern between the FIRO frequency and the sampling frequency, so that different patterns are sampled as well. It should also be noted that the periodic FIRO oscillation is of course not perfectly periodic, but shows some jitter, as classical ring oscillators do.

Fig. 9 shows a FIRO which starts to oscillate periodically but changes to short chaotic oscillations repeatedly.

So, the cases of intermediate numbers of different samples are explained by FIROs which change between oscillating chaotically and periodically.

For comparison, figures of the same kind are also shown for FIROs with non-intermediate numbers of samples. The FIRO from Fig. 10 had 36 different sampled patterns; the one from Fig. 11 had 6505 different sampled patterns.

## 8 Can the problem be fixed?

One idea to find suitable FIRO feedback configurations, which result in the desired chaotic oscillations, could be an experimental approach, which finds suitable FIRO designs by trial and error. However, it is not sufficient to determine a suitable feedback configuration once and for all and to use this configuration on a number of devices of the same build. It turned out that the occurrence of chaotic or periodic oscillations in many cases depended on the individual FPGA chip (of course all of them being of the same type). This makes FIROs good PUFs, a topic which will not be

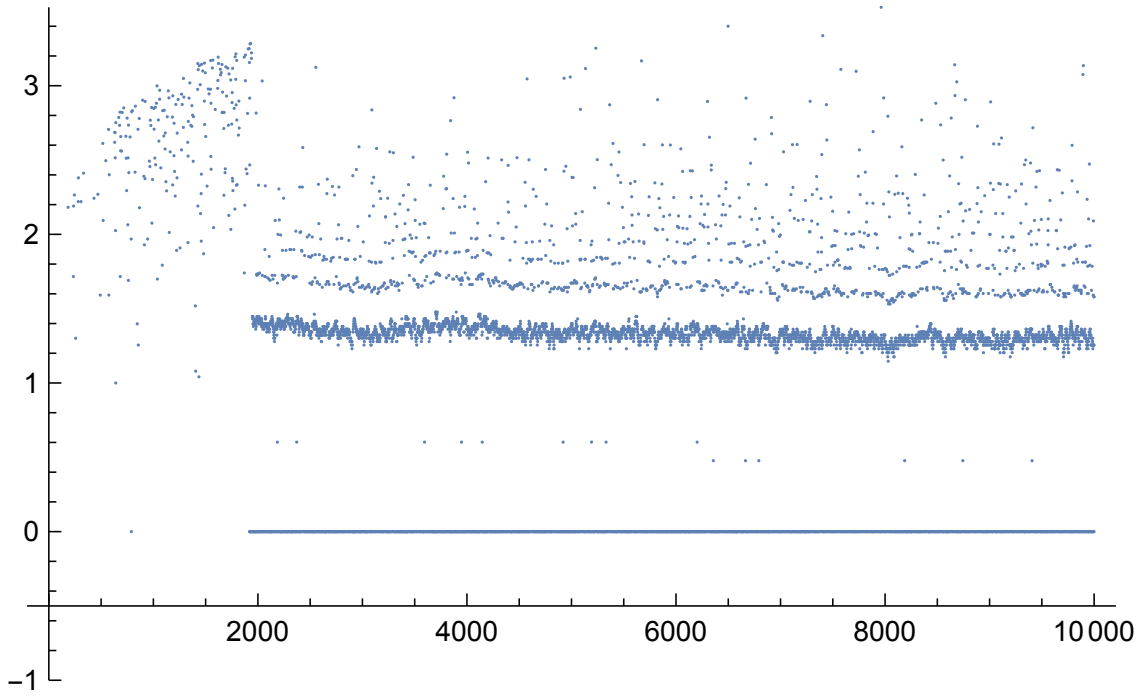


**Fig. 8.** The ordinate gives the decimal logarithm of the number of samples the same 16-bit-pattern has been sampled before for the last time. The abscissa gives the 10000 samples in their original temporal sequence. Note that what is shown is not a mathematical function; for the abscissae corresponding to a 16-bit-pattern observed for the first time, there is no dot. The FIRO sampled 10000 times had a feedback polynomial of $x^{16} + x^{14} + x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^4 + x^2 + 1$. The line (which is really a sequence of a large number of dots) in the lower right of the figure should be noted. It marks the 2437 cases when identical patterns occurred in subsequent samples. This example shows that even when a FIRO starts to oscillate chaotically, it may at any time - here at sample 1920 - switch its behaviour and start to oscillate periodically.

elaborated further in this paper. Even if a FIRO has shown to oscillate chaotically once, Section 7 makes clear that it cannot be relied on to keep up this chaotic oscillations, as it may change to periodic behaviour at any time.

So, a FIRO must be tightly monitored permanently whether it is indeed oscillating periodically. This monitoring can be achieved with the method described in this paper, namely by sampling all inverters of the FIRO simultaneously many times and by determining how often the different bit patterns occur. If the number of different bit patterns is low, the FIRO is oscillating periodically and does not produce sufficient amounts of entropy.

After finding out that the FIRO is oscillating periodically, the question is what to do. One option is to inform the application that needs random numbers that none can be produced. Then, the application can refuse to perform the security critical operation and inform the user about the unavailability of the service she wanted. This approach would be acceptable if the risk of FIRO malfunction was very low, but with a failure rate of more than 13 %, it is not acceptable.

So, just to give up and to declare that no random numbers can be produced, is no viable solution. Instead, the FIRO must be changed to work properly. This can, with a certain probability, be achieved by changing its feedback polynomial. For the new polynomial, it must also be determined whether the FIRO is oscillating chaotically; it not, the feedback polynomial must be changed again, ... . So, if the hardware circuitry for the feedback provides a sufficiently large number of different feedback polynomials, there is a very good chance of finding a suitable configuration for which the FIRO works properly.

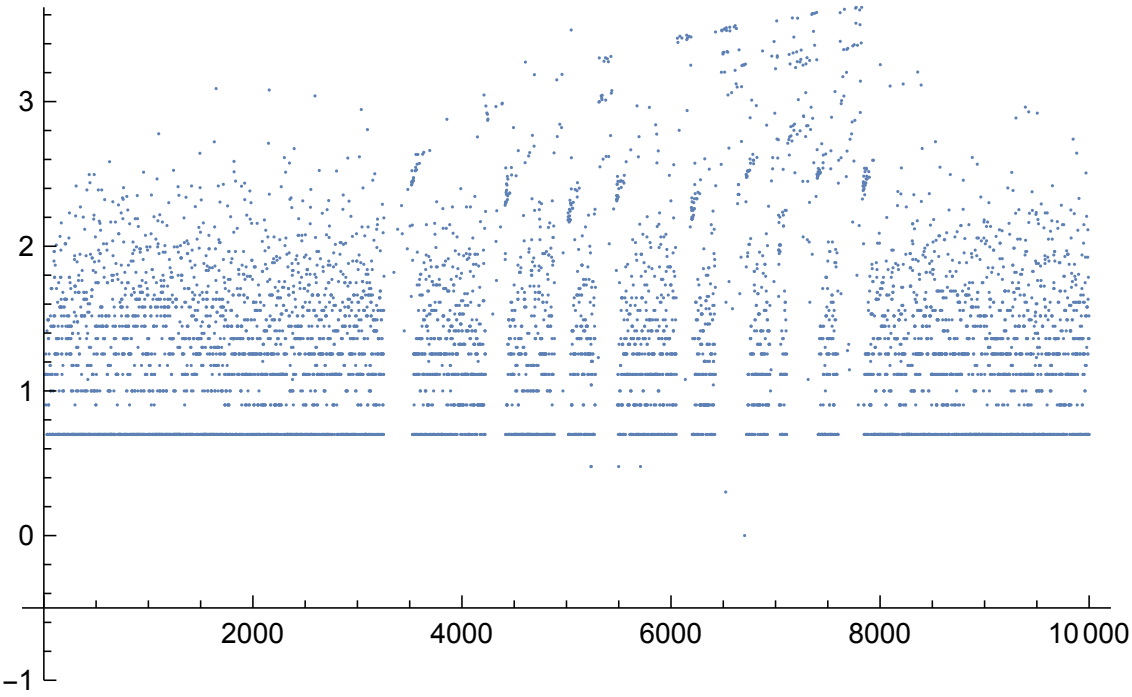


**Fig. 9.** Decimal logarithm of the number of samples the same 16-bit-pattern has been sampled before for the last time. Note that this is not a function; for the abscissae corresponding to a 16-bit-pattern observed for the first time, there is no point. The FIRO sampled 10000 times had a feedback polynomial of $x^{16}+x^{15}+x^{14}+x^{13}+x^{11}+x^8+x^7+x^5+x^2+1$. This example shows that even when a FIRO starts to oscillate periodically, it may switch its behaviour several times and start to oscillate chaotically for some milliseconds.

While this approach can deliver good random numbers with a very high probability, it is nevertheless problematic. One does not know in advance how many trials will be needed until a suitable feedback configuration is found. This implies that no time bound for the production of a certain number of random bits can be guaranteed. Typically, application designers want a fixed time bound and are not too happy about statements of the form "In 99 % of the cases, you will get your 128 random bits within 5 ms."

Besides the unachievable fixed time bound, the considerable hardware requirements for the suggested approach must also be taken into account. The flexible feedback circuitry can be implemented with a modest amount of additional hardware, but the storage needed to determine the number of different bit patterns is substantial.

## 9 Conclusions

### 9.1 What to conclude from the results of this paper

The experimental results described above make obvious that there is a considerable risk of Fibonacci ring oscillators to oscillate periodically. If this happens, the entropy produced is only a small percentage of the one from a Fibonacci ring oscillator working correctly. So, the failure of the Fibonacci ring oscillators means a considerable risk for security applications needing reliable random numbers. Security problems can be avoided if the correct operation of the Fibonacci ring oscillator is tightly monitored, but security applications which refuse to work in a substantial percentage of cases are inacceptable, because availability is also required.

The only way to operate Fibonacci ring oscillators in a manner which makes the application both secure and highly available is to combine the tight monitoring of the correct operation of the

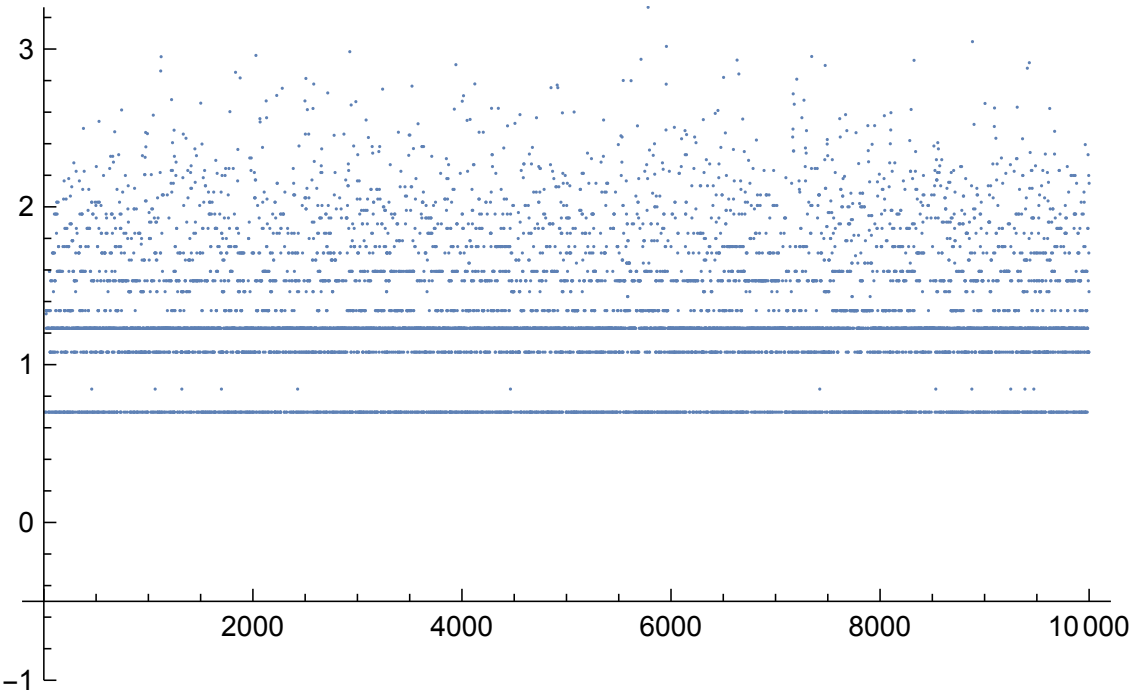


**Fig. 10.** Decimal logarithm of the number of samples the same 16-bit-pattern has been sampled before for the last time. The FIRO, which oscillated periodically all the time, had a feedback polynomial of $x^{16} + x^9 + x^5 + x^4 + x^3 + 1$.

Fibonacci ring oscillator with a highly flexible feedback circuitry. This feedback circuitry must be able to implement a large number of different feedback polynomials which keep changing, until the monitoring unit confirms the correct operation of the Fibonacci ring oscillator. However, the hardware effort for the monitoring and the flexible feedback is much larger than for the straight forward implementation of a Fibonacci ring oscillator.

### 9.2 What not to conclude from the results of this paper

Now, one might conclude from this paper that, as Fibonacci ring oscillators have turned out to be a risky way to generate random numbers, Galois ring oscillators might be the good choice for true random number generation. However, there is experimental evidence - without a broad statistical basis so far - that Galois ring oscillators may also show periodic instead of chaotic oscillations.

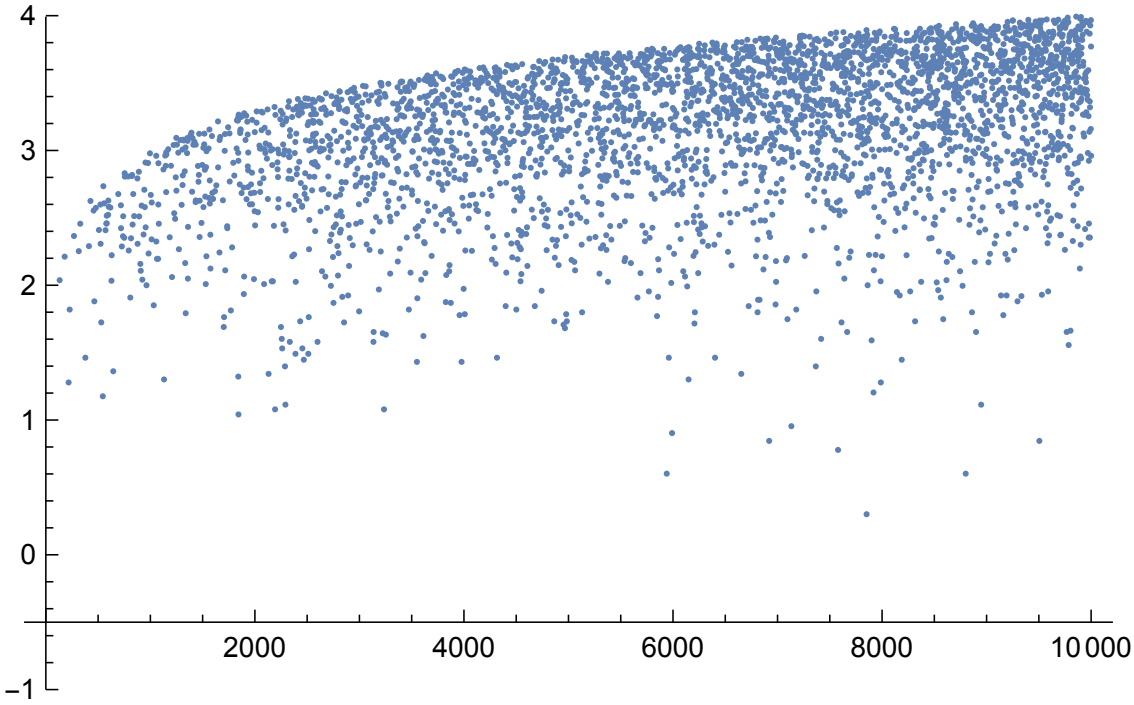


**Fig. 11.** Decimal logarithm of the number of samples the same 16-bit-pattern has been sampled before for the last time. The FIRO, which kept oscillating chaotically all the time, had a feedback polynomial of $x^{16} + x^3 + x^2 + 1$.

## References


[DG07] Markus Dichtl and Jovan Dj. Golić, *High-speed true random number generation with logic gates only*, CHES (Pascal Paillier and Ingrid Verbauwhede, eds.), Lecture Notes in Computer Science, vol. 4727, Springer, 2007, pp. 45–62.

[Gol06] Jovan Dj. Golić, *New methods for digital generation and postprocessing of random data*, IEEE Trans. Computers **55** (2006), no. 10, 1217–1229.

[MJ11] Łukasz Matuszewski and Mieczysław Jessa, *An auxiliary source of randomness for combined TRNG based on ring oscillators*, Poznańskie Warsztaty Telekomunikacyjne 2011, `http://www.pwt.et.put.poznan.pl/PWT_2011/PWT%202011_2237.pdf`, 2011.

[sSNA12] Jochen Ertel (secunet Security Networks AG), *Periodisches Schwingen eines FIRO mit einstellbarem Rücksetz-Pattern*, Project internal communication within the research project DICECUP, 2012.