True Random Number Generation Using Barkhausen Noise

BY: BIEBEL, BOTOND; BILSZKY, MÁRK

CONSULTANT: DR. NAGY, KATALIN

2023

Table of Contents

Abstract 2
1 Introduction 3
2 Methodology 3
2.1 Barkhausen Measurement 3
2.1.1 Introduction to Barkhausen Noise 3
2.1.2 Barkhausen Noise Measurement Instrument (Barkhausenmeter) 4
2.2 Data Processing 4
2.2.1 Nature of the Data Set 4
2.2.2 Fast Fourier Transform 5
2.2.3 Creating a Binary Dataset 7
3 Randomness Testing 7
3.1 Introduction to Randomness Testing 7
3.2 Randomness Testing Methods 8
3.2.1 Frequency (Monobit) Test 8
3.2.2 Frequency Test within a Block 8
3.2.3 Runs Test 9
3.2.4 Test for the Longest Run of Ones in a Block 10
3.2.5 Binary Matrix Rank Test 11
3.2.6 Discrete Fourier Transform (Spectral) Test 11
3.2.7 Non-overlapping Template Matching Test 12
3.2.8 Overlapping Template Matching Test 14
3.2.9 Linear Complexity Test 15
3.2.10 Serial Test 16
3.2.11 Approximate Entropy Test 17
3.2.12 Cumulative Sums (Cusum) Test 18
3.2.13 Random Excursions Test 19
3.2.14 Random Excursions Variant Test 19
3.3 Results of Tests 22
4 Conclusion 23
5 References 23

Abstract

Barkhausen Noise emerges as a promising avenue for generating random numbers, offering inherent unpredictability and sensitivity to external influences. In this study, we explore the feasibility of utilizing Barkhausen Noise as a source for random number generation. Through meticulous measurements and in-depth analysis, our research demonstrates the viability of Barkhausen Noise in producing random numbers. However, we also investigated the limitations, indicating that while Barkhausen Noise provides a viable method for generating random numbers, it falls short of meeting the stringent requirements of high-security applications. This research sheds light on the potentials and limitations of Barkhausen Noise in the realm of random number generation, paving the way for further investigations into its practical applications.

Keywords: cryptography, true random numbers, Barkhausen Noise

1 Introduction

Random numbers play a crucial role in various fields, especially in cryptography, where they form the foundation of secure algorithms and communication protocols. In the realm of random number generation, two distinct categories are recognized: pseudo random numbers, generated by algorithms which are inherently deterministic, and true random numbers, characterized by unpredictable, non-repeating sequences. While pseudo random numbers have many use cases, certain applications however, particularly in cryptographic algorithms, demand a higher level of unpredictability that only true random numbers can provide. Hence true random numbers are vital for applications where the integrity and confidentiality of information are paramount, such as in secure communications, digital signatures, gambling, scientific research, and encryption.

Despite the advancements in random number generation techniques, efficiently and reliably generating true random numbers remains an ongoing challenge.

In this study, we explore Barkhausen Noise measurements as a potential avenue for generating true random numbers. Barkhausen Noise, arising from the movement of magnetic domain walls in ferromagnetic materials, exhibits inherent randomness due to its complex underlying physical processes. The question we seek to answer is whether these naturally occurring phenomena can be harnessed to create a reliable source of true random numbers.

Through stringent analysis of Barkhausen Noise measurements, this study delves into the intricacies of this phenomenon, assessing its potential as a robust source of true randomness. By scrutinizing the data obtained from these measurements and subjecting it to rigorous randomness tests, we look to assess the practicality and effectiveness of Barkhausen Noise as a true random number generator. Our findings could pave the way for novel applications in cryptography and other domains that demand unparalleled levels of randomness and security.

2 Methodology

2.1 Barkhausen Measurement

2.1.1 Introduction to Barkhausen Noise

Barkhausen Noise stands out as a pivotal non-destructive magnetic measurement technique with widespread industrial applications, rendering it indispensable for quality control and precise process validation. Its versatility is exemplified in various contexts, such as the in-depth analysis of grinding burn effects on diverse ground components and meticulous surface inspections conducted after a range of heat treatments.

The genesis of Barkhausen Noise unfolds within ferromagnetic metals during the magnetization process, where the movement of domain walls within the material mirrors an intricate avalanche-like phenomenon, culminating in the generation of Barkhausen noise. This unique acoustic signal is not only a testament to the material's magnetic state but also possesses exceptional sensitivity to external influences and inherent unpredictability.

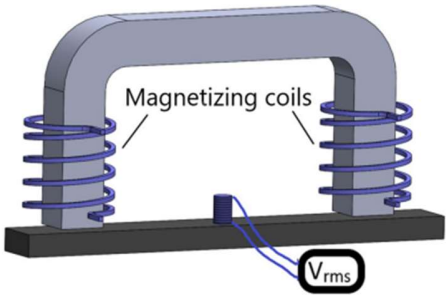
In the realm of our study, these characteristics of Barkhausen Noise become particularly intriguing. The sensitivity to external factors and the inherent unpredictability of this source raises the possibility of its suitability for random number generation, a fact that we explore in depth as part of our research.

2.1.2 Barkhausen Noise Measurement Instrument (Barkhausenmeter)

Traditional Barkhausen Noise measurement devices comprise two integral components: the magnetizing element and the probe, collectively unveiling the unique characteristics of materials under study. When placed in contact with the sample to be measured, they transmit voltage data to a computer.

At the heart of the measurement device lies the magnetizing element, typically fashioned as an open-loop solenoid. This solenoid plays a pivotal role in magnetizing the sample under investigation, setting the stage for the Barkhausen Noise to be effectively measured. It's noteworthy that the solenoid generates a sine magnetization signal, a fundamental aspect to bear in mind during subsequent data processing endeavors.

The magnetizing element consists of an open loop solenoid and provides magnetization for the tested sample allowing for the Barkhausen Noise to be measured. This solenoid generates a sine magenetization signal which is important to remember when processing the data.



1. Figure (Barkhausen Noise Measurement Device)

Complementing the magnetizing element, the probe features a pickup coil. As the magnetizing element excites the material, inducing magnetic responses within the sample, voltage is generated within the pickup coil. This induced voltage serves as a direct reflection of the material's magnetic behavior.

The voltage values acquired from the pickup coil are transmitted to a desktop computer for further analysis. Once received by the computer, these voltage measurements are processed and compiled into a comprehensive .txt file. This file encapsulates the essence of the material's magnetic response, forming the basis for in-depth examinations and subsequent data processing.

2.2 Data Processing

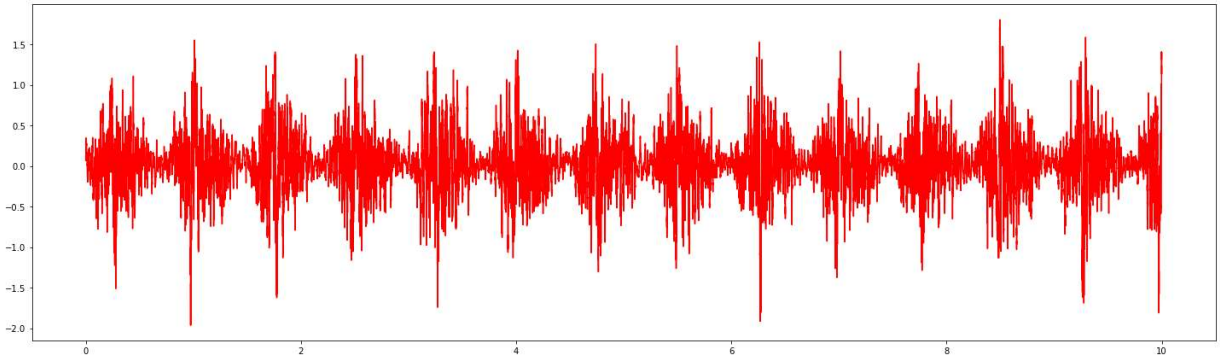
2.2.1 Nature of the Data Set

Upon completing the Barkhausen measurements, we obtained nine raw datasets (numbered from 1 to 9), each collected from different metals of varying hardness, structured in intervals marked by 'enters', and containing a generator signal inherent to the measurement method. Our data processing methodology was meticulously applied to refine these raw datasets into a usable format for analysis. To achieve this, we rigorously removed the generator signal and conducted thorough segmentation and filtering procedures tailored to the specific characteristics of each metal. This meticulous approach allowed us to separate genuine Barkhausen Noise signals from unwanted interference.

The resulting datasets were transformed into binary representations, where 'ones' and 'zeros' represented specific states of the magnetic domain walls' movements. Each of these binary datasets, derived from the nine distinct measurements conducted on metals of different hardness, served as the foundation for our subsequent analysis. This comprehensive approach ensured that the inherent randomness of Barkhausen Noise was accurately captured across all datasets, eliminating extraneous signals and noise, and establishing a robust basis for our research.

2.2.2 Fast Fourier Transform

In our dataset, a distinct sine generator signal (as depicted in Figure 2) was present alongside the Barkhausen noise. To extract the genuine Barkhausen noise, we employed the Fast Fourier Transform (FFT), a powerful technique widely used in signal processing.



2. Figure (plot of data set 1, showing the nature of the dataset)

Fourier analysis allows expressing a function as a sum of periodic components, enabling the recovery of the original signal from these components. When both the function and its Fourier transform are discretized, it becomes the Discrete Fourier Transform (DFT). The Fast Fourier Transform (FFT) is an efficient algorithm for computing the DFT.

Using the FFT method provided by the Python library scipy.py, we processed our dataset. Specifically, we utilized the Discrete Cosine Transform (DCT) and its inverse (IDCT) to achieve this goal.

In signal processing, there are various types of DCT, each with its own definition and application. We focused on Type I, Type II, Type III, and Type IV DCT, with Scipy implementing the first four types. These transforms allowed us to mathematically manipulate the data, isolating the generator signal for removal.

The definition of *unnormalized Type I DCT* is the following:

$$y[k] = x_0 + (-1)^k x_{N-1} + 2\sum_{n=1}^{N-2} x[n] \cos\left(\frac{\pi nk}{N-1}\right), \quad 0 \le k < N.$$

The definition of *unnormalized Type II DCT* is the following:

$$y[k] = 2\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi(2n+1)k}{2N}\right), \quad 0 \le k < N.$$

In case of the *normalized Type II DCT*, the DCT coefficients y[k] are multiplied by a scaling factor *f*:

$$f = \begin{cases} \sqrt{\frac{1}{4N}}, & if\ k = 0 \\ \sqrt{\frac{1}{2N}}, & otherwise \end{cases}$$

The definition of *unnormalized Type III DCT* is the following:

$$y[k] = x_0 + 2\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi n(2n+1)}{2N}\right), \quad 0 \le k < N.$$

The definition of *normalized Type III DCT* is the following:

$$y[k] = \frac{x_0}{\sqrt{N}} + \frac{2}{\sqrt{N}}\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi n(2n+1)}{2N}\right), \quad 0 \le k < N.$$

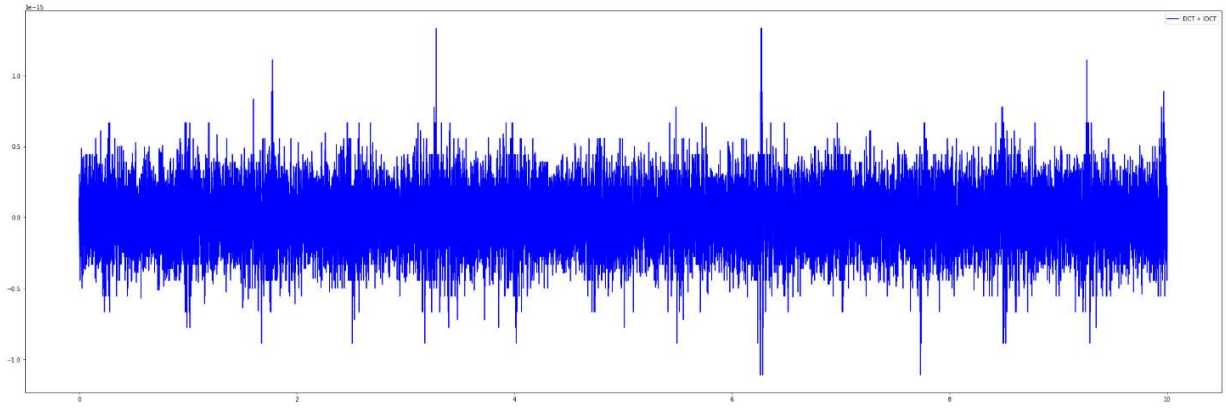
The definition of *unnormalized Type IV DCT* is the following:

$$[k] = 2\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi(2n+1)(2k+1)}{4N}\right), \quad 0 \le k < N.$$

The definition of *normalized Type IV DCT* is the following:

$$[k] = \sqrt{\frac{2}{N}}\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi(2n+1)(2k+1)}{4N}\right), \quad 0 \le k < N.$$

After applying FFT, we obtained the generator function. By subtracting this function from the original dataset, we successfully recovered our raw data, representing the pure Barkhausen noise (as illustrated in Figure 3). This refined dataset, free from unwanted signals, formed the basis of our subsequent analysis.



3. Figure (plot of noise on first dataset)

2.2.3 Creating a Binary Dataset

To convert the processed data into a binary format, a simple and effective method was employed. First, we calculated the median of the dataset, ensuring an equal distribution of ones and zeros. For each data point, if its value was greater than the median, we assigned it a value of zero; conversely, if the value was less than or equal to the median, it was assigned a value of one. This binary transformation created a balanced representation of the data, where 'ones' and 'zeros' accurately represented the specific states of the magnetic domain walls' movements. By employing this method, we achieved a well-defined binary dataset, ready for further analysis.

3 Randomness Testing

3.1 Introduction to Randomness Testing

Understanding true randomness is a nuanced challenge; there is no definitive source of randomness, and even seemingly random processes can reveal patterns under scrutiny. Consider a classic example like rolling a dice in a game of Yahtzee – seemingly random, yet influenced by various factors like table friction and rolling technique, potentially leading to recognizable patterns over multiple repetitions. While this might be inconsequential in the context of the game, in high-security applications where a substantial volume of numbers is required, these underlying patterns can pose significant problems.

In our experiments, our aim was to generate a random bit sequence comprising ones and zeros. For a bit sequence to be truly random, each flip must have a probability of ½ of being one or zero. Independence is crucial, ensuring that previous attempts will not influence future ones. Additionally, unpredictability is a necessity, in case the seed is unknown, no knowledge of previous outcomes should not allow for someone to determine the future outcomes.

To evaluate the randomness of our generated sequences, we adhered to the guidelines outlined by the National Institute of Standards and Technology (NIST). While these tests do not substitute for rigorous cryptoanalysis, they serve as essential tools in assessing the viability of Barkhausen Noise random number generation technology.

3.2 Randomness Testing Methods

3.2.1 Frequency (Monobit) Test

The test's primary focus lies in examining the ratio of zeros to ones within the entire sequence. The objective of this test is to ascertain whether the number of ones and zeros in a sequence closely approximates what one would expect in a truly random sequence. It evaluates how near the fraction of ones is to 1/2, meaning that the number of ones and zeros in the sequence should be roughly equal. All subsequent tests depend on the passing of this test. Therefore, if a sequence fails this test, there is no need to proceed with further testing, and the sequence should be regarded as non-random.

Test Description:

Conversion to ±1: The zeros and ones of the input sequence (ε) are converted to values of –1 and +1 and are added together to produce $S_n = X_1 + X_2 + \ldots + X_n$, where $X_i = 2\varepsilon_i - 1 = \pm 1$.

Compute the test statistic:

$$S_{obs} = \frac{|S_n|}{\sqrt{n}}$$

S_{obs} = the absolute value of the sum of the X_i in the sequence divided by the square root of the length of the sequence
n = the length of the bit string

$P_{value} = erfc\left(\frac{S_{obs}}{\sqrt{2}}\right)$, where erfc[1] is the complementary error function.

If the P_{value} were small (< 0.01), then this would be caused by $|S_n|$ or $|S_{obs}|$ being large. Large positive values of S_n suggest an excess of ones, while large negative values of Sn indicate an excess of zeros.

3.2.2 Frequency Test within a Block

The primary objective of this test is to assess the proportion of ones within M-bit blocks. Within each block the test runs the Frequency (Monobit) Test to determine whether the frequency of ones in an M-bit block is approximately equal to M/2, which aligns with the expectations of randomness.

Test Description:

Divide the sequence into $N = \left\lfloor \frac{n}{M} \right\rfloor$ equal, non-overlapping blocks, discarding any unused bits.

Determine the proportion q_i of ones in each M-bit blocks:

[1] The complementary error function, erfc(x), is defined, for **x ≥ 0**, as. (2/Sqrt[Pi]) erfc(t) =Integrate[E^(-t)^2, {t, z, Infinity}]

$$q_i = \frac{\sum_{j=1}^{M} \varepsilon_{(i-1)M+j}}{M}$$

Calculate χ^2_{obs}:

$$\chi^2_{obs} = 4M \sum_{i=1}^{M} (q_i - 0.5)^2$$

χ^2_{obs} = a measure of how closely the actual proportion of ones within a specific M-bit block aligns with the expected proportion (1/2)

Determine the P_{value}:

$P_{value} = igamc\left(\frac{N}{2}, \frac{\chi^2_{obs}}{2}\right)$, where igamc[2] is the incomplete gamma function

A small P_{value} means a large deviation from the equal proportion of zeros and ones in at least one of the blocks.

3.2.3 Runs Test

The focus of this test is the total number of uninterrupted sequences of identical bits (further referred as runs) in the whole sequence. A run of length l is characterized by a sequence of l consecutive bits with the same value, and it is delineated by bits with the opposite value both before and after the run. The objective of the test is to determine whether the number of runs of ones and zeros is as expected for a random sequence. Specifically, this test aims to assess whether the alternation between such zeros and ones is too fast or too slow.

Test Description:

Determine the proportion q of ones in the input sequence:

$$q = \frac{\sum_j \varepsilon_j}{n}$$

This test fails initially if the sequence did not pass the first test (Frequency Test)

The test statistic:

$$V_{n(obs)} = \sum_{k=1}^{n-1} r(k) + 1$$

r(k) = 0, if εk = εk+1 and r(k) = 1 otherwise

[2] The upper incomplete gamma function, igamc(x) is defined as: ∫x^∞ t^(s - 1) e^(-t) dt

Calculate P_{value}:

$$P_{value} = erfc\left(\frac{V_{n(obs)} - 2nq(1-q)}{2\sqrt{2n}q(1-q)}\right)$$

A large $V_{n(obs)}$ value arises when the string's oscillation pattern suggests an excessive speed of oscillation, whereas a small value indicates an oscillation that is excessively slow. Both of these causes P_{value} being below the pass barrier (<0.01).

3.2.4 Test for the Longest Run of Ones in a Block

The focus of the test is the longest run of ones within M-bit blocks. This test compares the length of the longest run of ones within the tested sequence to the length of the longest run of ones that would be expected in a random sequence. Hence in case of an irregularity in the expected length of the longest run of ones would cause an irregularity to the longest run of zeros, there is no need to test for both.

Test Description:

Divide the sequence into M-bit blocks.

Produce a table that categorizes the frequencies ν_i for the longest runs of consecutive ones within each block into distinct categories. In each cell of the table, display the count of runs of ones of a specific length.

Minimum *n*	***M***
128	8
6272	128
750,000	10^4

This test supports only the given M sizes, thus the ν_i cells will be the following:

Calculate χ^2_{obs}:

$$\chi^2_{obs} = \sum_{i=0}^{K} \frac{(\nu_i - Nq_i)^2}{Nq_i}$$

χ^2_{obs} = a measure of how well the observed longest run length within M-bit blocks matches the expected longest length within M-bit blocks

q_i = as calculated in 4.2.2

the value of K and N are determined by the following table:

M	K	N
8	3	16
128	5	49
10^4	6	75

Determine the P_{value}:

$$P_{value} = igamc\left(\frac{K}{2}, \frac{\chi^2_{obs}}{2}\right)$$

3.2.5 Binary Matrix Rank Test

This test focuses on the rank of disjoint sub-matrices of the entire sequence. The purpose of this test is to examine the presence of linear dependence within fixed-length substrings of the original sequence.

Test Description:

Divide the sequence into MxQ-bit disjoint blocks sequentially. There will exist $N = \left\lfloor \frac{n}{MQ} \right\rfloor$ such blocks. Make M by Q matrices of the MxQ bit segments. Each row of the matrix is filled with successive Q-bit blocks of the original sequence.

M = the number of rows in each matrix (in this test: M=32)

Q = the number of columns in each matrix (in this test: Q=32)

Calculate the binary rank (R_i) of each matrix, where i = 1,…,N.

Determine χ^2_{obs}:

$$\chi^2_{obs} = \frac{(F_M - 0.2888N)^2}{0.2888N} + \frac{(F_{M-1} - 0.5776N)^2}{0.5776N} + \frac{(N - F_M - F_{M-1} - 0.1336N)^2}{0.1336N}$$

χ^2_{obs} = a measure of how well the observed number of ranks of various orders match the expected number of ranks expected for randomness

F_M = the number of matrices with full rank à $R_i = M$

F_{M-1} = the number of matrices with full rank – 1 à $R_i = M-1$

Determine the P_{value}:

$$P_{value} = e^{-\chi^2_{obs}/2}$$

A large value of χ^2_{obs} indicates a deviation of the rank distribution from what would be expected in a random sequence.

3.2.6 Discrete Fourier Transform (Spectral) Test

This test focuses on the peak heights in the Discrete Fourier Transform of the sequence. It aims to detect periodic features (repetitive patterns that are near each other for instance) in the tested sequence. These would indicate a deviation from the assumption of randomness. The intention is to determine if the quantity of peaks surpassing the 95% threshold significantly differs from the expected 5%.

v_i	M = 8	M = 128	$M = 10^4$
v_0	≤ 1	≤ 4	≤ 10
v_1	2	5	11
v_2	3	6	12
v_3	≥ 4	7	13
v_4		8	14
v_5		≥ 9	15
v_6			≥ 16

Test Description:

Conversion to ±1: The zeros and ones of the input sequence (ε) are converted to values of –1 and +1 and are added together to produce $X = x_1 + x_2 + \ldots + x_n$, where $x_i = 2\varepsilon_i - 1 = \pm 1$.

Apply a Discrete Fourier transform (DFT) on X to produce: S = DFT(X). It produces a sequence of complex variables which represents periodic components of bits of the sequence at different frequencies.

Determine M:

$$M = modulus(S') = |S^*|$$

S'= the substring including the first n/2 elements of S à peak heights are being produced by the modulus function

Calculate T:

$$T = \sqrt{\left(log\frac{1}{0.05}\right)n}$$

T = the 95% peak height threshold value

Calculate N_0:

$$N_0 = .95/2.$$

N_0 = the expected theoretical (95%) number of peaks that are less than T

Calculate N_1; the count of observed peaks in M that are less than T

Determine d:

$$d = \frac{(N_1 - N_0)}{\sqrt{n(.95)(.05)/4}}$$

d = the normalized difference between the observed and the expected number of frequency components exceeding the 95% threshold

Calculate P_{value}:

$$P_{value} = erfc\left(\frac{|d|}{2}\right)$$

A small d value indicates that there were not enough peaks(<95%) below T, and too many peaks above T (more than 5%), thus a sequence should fail.

3.2.7 Non-overlapping Template Matching Test

This test searches through the entire sequence, looking for the number of occurrences of predefined target strings. It aims to detect if there are any aperiodic patterns occurring too many times. An m-bit window is used to search for a specific m-bit pattern both in this, and the next test (Overlapping Template Matching test). The window slides one bit position if the pattern is not found. Upon discovering the pattern, the search process continues by resetting the window to the bit immediately following the identified pattern.

Test Description:

Separate the sequence into N independent blocks, each of length M.

N is set to 8 in our test

The number of times that B (the template) occurs within the block j:= W_j, where j=1,…,N

B = the m-bit template to be matched, which is a predefined sequence of ones and zeros

To look for matches, an m-bit window is positioned on the sequence, and the bits within this window are compared to the template. If no match is detected, the window moves forward by one bit. If there is a match, the window slides over m bits.

Bit Positions	Block 1		Block 2	
	Bits	W_1	Bits	W_2
1-3	101	0	111	0
2-4	010	0	110	0
3-5	100	0	100	0
4-6	001 (hit)	Increment to 1	001 (hit)	Increment to 1
5-7	Not examined		Not examined	
6-8	Not examined		Not examined	
7-9	001	Increment to 2	011	1
8-10	010 (hit)	2	110	1

Calculate the theoretical mean μ and variance σ^2 under the assumption of randomness:

$$\mu = \frac{(M-m+1)}{2^m} \qquad \sigma^2 = M\left(\frac{1}{2^m} - \frac{2m-1}{2^{2m}}\right)$$

m = the length in bits of each template

M = the length in bits of the substring of ε to be tested

Determine χ^2_{obs}

$$\chi^2_{obs} = \sum_{j=1}^{N} \frac{(W_j - \mu)^2}{\sigma^2}$$

χ^2_{obs} *= An indicator of the degree to which the actual count of template "hits" corresponds to the anticipated count of template "hits" (under the assumption of randomness)*

Calculate the P_{value}:

$$P_{value} = igamc\left(\frac{N}{2}, \frac{\chi^2_{obs}}{2}\right)$$

If the P-value is extremely low(<0.01), it indicates that the sequence exhibits irregular instances of potential template patterns.

3.2.8 Overlapping Template Matching Test

This test searches through the entire sequence, looking for the number of occurrences of predefined target strings. It aims to detect if there are any aperiodic patterns occurring too many times. An m-bit window is used to search for a specific m-bit pattern both in this, and the previous test (Non-overlapping Template Matching test). The window slides one bit position if the pattern is not found. The difference between this test and the previous test (Non-overlapping Template Matching test) is that when the pattern is found, the window slides only one bit before resuming the search.

Test Description:

Separate the sequence into N independent blocks, each of length M.

N is set to 968 in our test

Determine the number of occurrences of B within each of the N blocks. The search for matches involves the establishment of an m-bit window on the sequence, where the bits contained within this window are compared to B, and a counter is incremented upon finding a match. After each examination the window slides over one bit. Maintain a record of the occurrences of B in each block by updating an array q_i (where i ranges from 0 to 5). Specifically, increment q_0 when there are no B occurrences in a substring, q_1 when there is one occurrence of B, and q_5 when there are five or more occurrences of B.

B = the m-bit template to be matched

Calculate values for λ and η that will be used to determine the theoretical probabilities π_i corresponding to the classes of v_0:

$$\lambda = \frac{(M-m+1)}{2^m} \qquad \eta = \lambda/2$$

m = the length in bits of the template, here it is the length of the run of ones

M = the length in bits of a substring ε to be tested, in this test it has been set to 1032

Determine χ^2_{obs}:

$$\chi^2_{obs} = \sum_{i=0}^{5} \frac{(q_i - Nq_i)^2}{Nq_i}$$

χ^2_{obs} *= An indicator of the degree to which the actual count of template "hits" corresponds to the anticipated count of template "hits" (under the assumption of randomness).*

Calculate the P_{value}:

$$P_{value} = igamc\left(\frac{5}{2}, \frac{\chi^2_{obs}}{2}\right)$$

For the 2-bit template (B = 11), if the entire sequence had too many 2-bit runs of ones, then: v5 would have been too large, the test statistic would be too large, the P-value would have been small (< 0.01) and a conclusion of non-randomness would have resulted.

3.2.9 Linear Complexity Test

The test primarily assesses the length of a linear feedback shift register (LFSR) to determine whether or not the sequence exhibits the requisite complexity to be classified random. Longer LFSRs characterize random sequences, while a too short LFSR implies non-randomness.

Test Description:

Separate the n-bit sequence into N distinct blocks, each consisting of M bits, such that n = MN.

With the help of the Berlekamp-Massey algorithm[3], calculate the linear complexity L_i of each of the N blocks (i = 1, 2,…,N). L_i represents the minimum length of a linear feedback shift register sequence required to generate all the bits within block i. In any sequence of L_i bits, there exists a particular combination of those bits such that, when summed modulo 2, it generates the next bit in the sequence, denoted as bit $L_i + 1$.

Under the assumption of randomness, calculate the theoretical mean μ:

$$\mu = \frac{M}{2} + \frac{(9 + (-1)^{M+1})}{36} - \frac{\left(\frac{M}{3} + \frac{2}{9}\right)}{2^M}$$

M = the length in bits of a block

Determine the value of T_i for each substring:

$$T_i = (-1)^M \bullet (L_i - \mu) + \frac{2}{9}$$

Record the T_i values in $q_0,\ldots,q_6$ as follows:

If:

$T_i \leq -2.5$	Increment v_o by one
$-2.5 < T_i \leq -1.5$	Increment v_1 by one
$-1.5 < T_i \leq -0.5$	Increment v_2 by one
$-0.5 < T_i \leq 0.5$	Increment v_3 by one
$0.5 < T_i \leq 1.5$	Increment v_4 by one
$1.5 < T_i \leq 2.5$	Increment v_5 by one
$T_i > 2.5$	Increment v_6 by one

[3] Defined in The Handbook of Applied Cryptography; A. Menezes, P. Van Oorschot and S. Vanstone; CRC Press, 1997.

Determine χ^2_{obs}:

$$\chi^2_{obs} = \sum_{i=0}^{K} \frac{(q_i - N\pi_i)^2}{N\pi_i}$$

$\pi_0 = 0.010417$, $\pi_1 = 0.03125$, $\pi_2 = 0.125$, $\pi_3 = 0.5$, $\pi_4 = 0.25$, $\pi_5 = 0.0625$, $\pi_6 = 0.020833$ are predefined probabilities

K = the number of degrees of freedom, which is set to 6 in this test

χ^2_{obs} = a measure of the degree to which the observed number of fixed-length LFSR occurrences aligns with the expected number of occurrences based on the assumption of randomness.

Calculate the P_{value}:

$$P_{value} = igamc\left(\frac{K}{2}, \frac{\chi^2_{obs}}{2}\right)$$

If the P-value were less than 0.01, it would imply that the observed frequency counts of T_i stored in the q_i bins deviate from the expected values. The expectation is that the distribution of T_i frequencies within the q_i bins should be in proportion to the computed π_i.

3.2.10 Serial Test

The central objective of this test is to analyze the frequency of all possible overlapping m-bit patterns throughout the entire sequence. The aim is to assess whether the observed number of occurrences of these 2^m m-bit overlapping patterns closely matches what would be expected in a random sequence. In random sequences, there is a uniform distribution, meaning that every m-bit pattern has an equal likelihood of appearing compared to every other m-bit pattern. It's worth noting that when m equals 1, the Serial test is equivalent to the Frequency test.

Test Description:

Create an augmented sequence ε′, by extending the original sequence. This extension involves adding the first m-1 bits to the end of the sequence, and this process is repeated for different values of n.

m = the length in bits of each block

n = the length in bits of the bit string

Calculate the occurrence frequency of all potential overlapping m-bit, (m-1)-bit, and (m-2)-bit blocks. Let $q_{i_l \ldots i_m}$ denote the frequency of the m-bit pattern $i_l \ldots i_m$; let $q_{i_l \ldots i_{m-1}}$ denote the frequency of the (m-1)-bit pattern $i_l \ldots i_{m-1}$; let $q_{i_l \ldots i_{m-2}}$ denote the frequency of the (m-2)-bit pattern $i_l \ldots i_{m-2}$.

Calculate:

$$\psi_m^2 = \frac{2^m}{n} \sum_{i_l \ldots i_m} \left(q_{i_l \ldots i_m} - \frac{n}{2^m} \right)^2$$

$$\psi_{m-1}^2 = \frac{2^{m-1}}{n} \sum_{i_l \ldots i_{m-1}} \left(q_{i_l \ldots i_{m-1}} - \frac{n}{2^{m-1}} \right)^2$$

$$\psi_{m-2}^2 = \frac{2^{m-2}}{n} \sum_{i_l \ldots i_{m-2}} \left(q_{i_l \ldots i_{m-2}} - \frac{n}{2^{m-2}} \right)^2$$

Calculate the test statistic:

$$\nabla\psi_m^2 = \psi_m^2 - \psi_{m-1}^2$$

$$\nabla^2\psi_m^2 = \psi_m^2 - 2\psi_{m-1}^2 + \psi_{m-2}^2$$

$\nabla\psi_m^2$ *and* $\nabla^2\psi_m^2$ *= a measure of how well the observed frequencies of m-bit patterns match the expected frequencies of the m-bit patterns*

Calculate the P_{value}-s:

$$P_{value} = igamc(2^{m-2}, \nabla\psi_m^2)$$
$$P_{value2} = igamc(2^{m-3}, \nabla^2\psi_m^2)$$

Large $\nabla\psi_m^2$ *and* $\nabla^2\psi_m^2$ *values indicates non-uniformity of the m-bit blocks.*

3.2.11 Approximate Entropy Test

Much like the Serial test discussed previously, this test primarily examines the frequency of all conceivable overlapping m-bit patterns throughout the sequence. The objective of the test is to evaluate how the frequency of overlapping blocks with two consecutive or adjacent lengths (m and m+1) compares to the expected frequencies for a random sequence.

Test Description:

Create an augmented sequence ε′, by extending the original sequence. This extension involves adding the first m-1 bits to the end of the sequence.

m = the length of each block

n = the length of the entire bitsequence

Create a frequency count of the n overlapping blocks. Represent the count of the possible m-bit ((m-1)-bit) values as C_i^m.

i = the m-bit value

Calculate for each value of i:

$$C_i^m = \frac{\#i}{n}$$

Determine $\phi^{(m)}$:

$$j = log_2 i$$

$$q_i = C_j^3$$

$$\phi^{(m)} = \sum_{i=0}^{2^m-1} q_i log(q_i)$$

Repeat that process sated above replacing m by m+1

Calculate the test statistics:

$$\chi^2_{obs} = 2n[log2 - ApEn(m)]$$

ApEn(m)= $\phi^{(m)}$- $\phi^{(m-1)}$

χ^2_{obs} = *a measure of how well the observed ApEn(m) matches the expected value*

Calculate the P_{value}:

$$P_{value} = igamc\left(2^{m-1}, \frac{\chi^2_{obs}}{2}\right)$$

If the value of ApEn(m) is small, it implies strong regularity. Large values would imply substantial fluctuation or irregularity.

3.2.12 Cumulative Sums (Cusum) Test

This test focuses on the maximum deviation (from zero) in the random walk formed by the cumulative sum of adjusted (-1, +1) digits in the sequence. Its purpose is to assess whether the cumulative sum of partial sequences within the tested sequence deviates significantly from the expected behavior of such a sum in random sequences. This cumulative sum can be likened to a random walk. In truly random sequences, the deviations of this random walk from zero should be minimal. However, in specific types of non-random sequences, these deviations can be substantial.

Test description:

Create the normalized sequence (consisting of -1, +1) named X, by converting the input 1 and 0 values of the input sequence (ε) to values of -1 and +1. this conversion is performed according to the following formula:

$$X=X_1, X_2, \ldots, X_n, \text{ where } X_i = 2\varepsilon_i - 1$$

Calculate partial sums S_i for increasingly larger subsequences, initiating each subsequence with X_1 (if mode = 0) or X_n (if mode = 1):

Mode = 0 (forward)	Mode = 1 (backward)
$S_1 = X_1$ $S_2 = X_1 + X_2$ $S_3 = X_1 + X_2 + X_3$ . . . $S_n = X_1 + X_2 + X_3 + \ldots + X_n$	$S_1 = X_n$ $S_2 = X_n + X_{n-1}$ $S_3 = X_n + X_{n-1} + X_{n-2}$ . . . $S_n = X_n + X_{n-1} + X_{n-2} + \ldots + X_1$

Calculate the test statistic $z = \max_{1\le n}|S_k|$, where $\max_{1\le n}|S_k|$ represents the largest absolute value among the partial sums S_k.

Calculate P_{value}:

$$P_{value} = 1 - \sum_{k=\frac{\left(\frac{-n}{z}+1\right)}{4}}^{\frac{\left(\frac{n}{z}-1\right)}{4}} \left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}}\right)\right] + \sum_{k=\frac{\left(\frac{-n}{z}+1\right)}{4}}^{\frac{\left(\frac{n}{z}-1\right)}{4}} \left[\Phi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right)\right]$$ [4]

If the P-value < 0.01, then the sequence is concluded to be non-random, in every other case it is considered random.

3.2.13 Random Excursions Test

This test specifically examines the occurrences of cycles with precisely K visits in a cumulative sum random walk. The cumulative sum random walk is constructed by summing partial values after converting the (0,1) sequence to the corresponding (-1, +1) sequence. In this context, a cycle represents a sequence of random steps with a unit length, starting and ending at the origin. The purpose of this test is to determine whether the number of visits to a particular state within a cycle

[4] Φ is the Standard Normal Cumulative Probability Distribution Function

deviates from what one would expect for a random sequence. The test comprises eight individual assessments, each corresponding to one of the states: -4, -3, -2, -1, and +1, +2, +3, +4.

Test description:

Create the normalized sequence (consisting of -1, +1) named X, by converting the input 1 and 0 values of the input sequence (ε) to values of -1 and +1. this conversion is performed according to the following formula:

$$X=X_1, X_2, \ldots, X_n, \text{ where } X_i = 2\varepsilon_i - 1$$

Next, we must calculate the partial sums of successively larger subsequences, each starting with X_1. The set is the following: $S = \{S_i\}$.

$$S_1 = X_1$$

$$S_2 = X_1 + X_2$$

$$S_3 = X_1 + X_2 + X_3$$

$$\cdot$$

$$\cdot$$

$$\cdot$$

$$S_n = X_1 + X_2 + X_3 + \ldots + X_n$$

Let J represent the total count of zero crossings in S', where a zero crossing means a transition from a non-zero value to zero in S'. J also corresponds to the number of cycles in S', where a cycle in S' is defined as a subsequence containing an initial zero, followed by non-zero values, and concluding with another zero. The ending zero in one cycle can serve as the beginning zero in another cycle. The quantity of cycles in S' is directly equated to the count of zero crossings. If the value of J is less than 500, the test should be discontinued.

Calculate the occurrence frequency of each non-zero state value x, where $-4 \leq x \leq -1$ and $1 \leq x \leq 4$, within every cycle.

Calculate νk(x) for each of the eight states of x, representing the total number of cycles in which state x occurs exactly k times across all cycles, where k ranges from 0 to 5 (for k = 5, store all frequencies ≥ 5 in ν5(x)).

Now we compute for each of the eight state of the test statistic:

$$\chi 2(\text{ops}) = \sum_{k=0}^{5} \frac{(v_k(x) - J\pi_k(x))^2}{J\pi_k(x)}$$

where $\pi_k(x)$ is the probability that the state x occurs k times in a random distribution

For each of the eight state of x, we calculate the P_{value}-s according to the following equation:

$$P_{value} = igamc\left(\frac{5}{2}, \frac{\chi 2(\text{obs})}{2}\right)$$

If the P-value < 0.01, then the sequence is concluded to be non-random, in every other case it is considered random.

3.2.14 Random Excursions Variant Test

The test aims to assess the frequency of encountering a particular state during various walks. Its purpose is to detect any deviations from the expected visits that would occur in a random walk scenario. The test comprises eighteen subtests, each corresponding to a specific state: -9, -8, … -1, and +1, +2,…, +9.

Test description:

Create the normalized sequence (consisting of -1, +1) named X, by converting the input 1 and 0 values of the input sequence (ε) to values of -1 and +1. this conversion is performed according to the following formula:

$$X=X_1, X_2, \ldots, X_n, \text{ where } X_i = 2\varepsilon_i - 1$$

Next we must calculate the partial sums of successively larger subsequences, each starting with X_1. The set is the following: $S = \{S_i\}$.

$$S_1 = X_1$$

$$S_2 = X_1 + X_2$$

$$S_3 = X_1 + X_2 + X_3$$

$$\cdot$$

$$\cdot$$

$$\cdot$$

$$S_n = X_1 + X_2 + X_3 + \ldots + X_n$$

Now we generate the modified sequence S', by adding zeros before and after the original set S. In other words, S' is constructed as: $S' = 0, S_1, S_2, \ldots, S_n, 0$.

For each of the eighteen non-zero states of X, $\xi(x)$[5] must be computed.

For each $\xi(x)$ we must compute the P_{value} (for all eighteen states) as:

$$P_{value} = erfc\left(\frac{|\xi(x) - J|}{\sqrt{2J(4|x| - 2)}}\right)$$

If the P-value < 0.01, then the sequence is concluded to be non-random, in every other case it is considered random.

[5] The number of times state X occurred across the J cycle

3.3 Results of Tests

In the table below, we present a summary of our testing results for each data set[6]:

	File 1	File 2	File 3	File 4	File 5	File 6	File 7	File 8	File 9
Test 1	✓	✓		✓	✓			✓	✓
Test 2									
Test 3									
Test 4									
Test 5	✓	✓	✓	✓	✓	✓	✓	✓	✓
Test 6		✓	✓					✓	✓
Test 7									
Test 8									
Test 9	✓	✓	✓	✓	✓		✓	✓	✓
Test 10									
Test 11									
Test 12a	✓	✓		✓	✓			✓	✓
Test 12b	✓	✓		✓	✓			✓	✓
Test 13 / 8	7✓	8✓	6✓	7✓	6✓	8✓	8✓	8✓	6✓
Test 14 / 18	18✓	18✓	18✓	18✓	18✓	18✓	18✓	18✓	18✓
Failed tests:	9	7	12	9	10	12	11	7	9
Is it random?	No	Yes	No	No	No	No	No	Yes	No

Note: A test is considered passed if a tick is present in the appropriate column. The numbers next to the ticks indicate the subtests that have been passed. According to the standards proposed by NIST a data set is considered random if it fails no more than seven tests.

These results underscore the innate randomness within Barkhausen Noise. However, our data processing methods proved inadequate for generating true random numbers. Challenges emerged particularly in tests assessing subsequent equal bits and overlapping patterns within the dataset. These issues might be attributed to our data processing approach; although we successfully identified the generator signal through FFT, subtle generator signals could have been overlooked.

Barkhausen Noise is susceptible to external influences, such as Cosmic Background Radiation (CBR) and radio signals, which might have interfered with the results. Further refinements in our data processing techniques are necessary to harness the full potential of Barkhausen Noise for random number generation.

[6] The code used for testing and the tested data sets can be found at: https://github.com/BBotond03/Barkhausen_TDK

4 Conclusion

In this study, we examined the potential of utilizing Barkhausen Noise measurements for random number generation, while also exploring the limitations inherent in our approach. Although our experiments did not immediately yield results applicable to high-security applications, they illuminated the substantial degree of randomness inherent in Barkhausen Noise. The findings of this research carry significant implications for the field of random number generation, shedding light on the challenges of harnessing natural phenomena for cryptographic purposes.

To enhance the effectiveness of Barkhausen Noise as a random number generator, future research should concentrate on refining methodologies in data processing. Exploring innovative techniques, such as simultaneous measurements and cross-referencing signals from multiple sources, holds promise in mitigating the limitations observed in this study, particularly the presence of repeating signals from external sources.

While challenges persist, the pursuit of refining these methods is essential in realizing the full capabilities of Barkhausen Noise as a reliable source of true randomness.

Acknowledgments

The authors would like to express their sincere gratitude to Professor Dr. Mészáros, István for his invaluable guidance and support at the inception of this research. His expertise and insightful suggestions played a pivotal role in shaping the initial idea and conducting the preliminary measurements. We are deeply thankful for his mentorship and contributions to this study.

5 References

1. Spasojević, D., Bukvić, S., Milošević, S. and Stanley, H.E., 1996. Barkhausen noise: Elementary signals, power laws, and scaling relations. Physical Review E, 54(3), p.2531.
2. Santa-aho, S., Laitinen, A., Sorsa, A. and Vippola, M., 2019. Barkhausen noise probes and modelling: A review. *Journal of Nondestructive Evaluation*, *38*(4), p.94.
3. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A. and Dray, J., 2001. *A statistical test suite for random and pseudorandom number generators for cryptographic applications* (Vol. 22). US Department of Commerce, Technology Administration, National Institute of Standards and Technology.
4. Barr, A.S., 2010. Fast Fourier Transform.
5. Cooley, James W., and John W. Tukey, 1965, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.* 19: 297-301.
6. Press, W., Teukolsky, S., Vetterline, W.T., and Flannery, B.P., 2007, *Numerical Recipes: The Art of Scientific Computing*, ch. 12-13. Cambridge Univ. Press, Cambridge, UK.
7. https://github.com/stevenang/randomness_testsuite/blob/master/BinaryMatrix.py?fbclid=IwAR2LbTUbNBCaMFGmJZL9gVxiD4t8mBLN-tA1jQcDHFPrRYgOx9SC4yUrHWA
8. J. Makhoul, 1980, 'A Fast Cosine Transform in One and Two Dimensions', *IEEE Transactions on acoustics, speech and signal processing* vol. 28(1), pp. 27-34, DOI:10.1109/TASSP.1980.1163351
9. A. J. S. Hamilton, 2000, "Uncorrelated modes of the non-linear power spectrum", *MNRAS*, 312, 257. DOI:10.1046/j.1365-8711.2000.03071.x

10. https://docs.scipy.org/doc/scipy/tutorial/fft.html
11. Chen, Y., Gou, B., Yuan, B., Ding, X., Sun, J. and Salje, E.K., 2022. Multiple Avalanche Processes in Acoustic Emission Spectroscopy: Multibranching of the Energy− Amplitude Scaling. *physica status solidi (b)*, *259*(3), p.2100465.
12. https://www.comm.utoronto.ca/frank/notes/erfc.pdf
13. https://mathworld.wolfram.com/IncompleteGammaFunction.html