

Multiplierless Implementation of Digital Colored Noise Generator

DOI 10.7305/automatika.2014.01.582
UDK 621.37.049.77
IFAC 3.1; 4.5

Original scientific paper

Colored noise can be generated by filtering of the white noise. It is a simple task. However, it becomes challenging if high operating speed of the generator is required. The realization of digital filter generally requires one multiplication per coefficient. Therefore, a high operating speed is achieved only with the cost of several general-purpose multipliers. In this paper, a multiplierless realization of the colored noise generator is proposed. It is based on the filtering of 1-bit random signal by a finite impulse response filter. The design of the generator is described and its implementation is considered. Furthermore, an application is described in which the proposed generator is used in mitigation of undesired effects caused by nonlinearities in an analog to digital converter.

Key words: Noise generator, Colored noise, Dither generator, Analog to digital converter, ADC, Multiplierless implementation, Field programmable gate arrays, FPGA

Implementacija digitalnog generatora obojenog šuma bez množila. Obojeni šum može se generirati filtriranjem bijelog šuma. To je jednostavan postupak. Međutim, on postaje izazovan ako se od generatora traži velika brzina rada. Realizacija digitalnog filtra u pravilu zahtijeva jedno množenje po koeficijentu. Zato se velika brzina rada može postići samo uz cijenu većeg broja množila opće namjene. U ovom radu predložena je realizacija generatora obojenog šuma koja ne sadrži množila. Ona se temelji se na filtraciji 1-bitnog slučajnog signala pomoću filtra s konačnim impulsnim odzivom. Opisano je projektiranje generatora te je razmotrena njegova implementacija. Nadalje, opisana je primjena u kojoj je predloženi generator iskorišten za smanjivanje utjecaja nelinearnosti u analogno digitalnom pretvorniku.

Ključne riječi: generator šuma, obojani šum, generator podrhtavanja, analogno digitalni pretvornik, ADC, implementacija bez množila, programabilna logička polja, FPGA

1 INTRODUCTION

Noise generators are widely used in evaluation of transmission systems and in mitigation of undesired effects caused by nonlinearities. These applications set different requirements on the generator design. The former requires high performance Gaussian noise generators [1–4] or generators with arbitrary distributions [5]. On the other hand, simplicity of the design is preferred in the latter.

The mitigation of undesired effects caused by nonlinearities is based on the technique called dithering [6]. Dithering is often used to increase spurious free dynamic range (SFDR) of analog to digital converters (ADCs) [7], [8]. In such an application, a wideband noise is added to converter's input signal to randomize the error caused by nonuniform distribution of quantization levels. Clearly, a noise added at the converter's input reduces the signal to noise ratio (SNR). It is often acceptable because the gain

achieved in SFDR is larger than the loss in SNR [6]. However, significant loss in SNR is not tolerated in many applications such as in high dynamic-range radio receivers. In a receiver, the loss in SNR increases the noise floor and thus decreases the sensitivity [9], [10]. To avoid it, the noise is added only in the frequency band which is not occupied by the signal of interest. The band occupied by the noise is typically placed around zero or the Nyquist frequency [11], [12].

The noise with a specified power spectral density can be obtained by filtering of the white noise [13]. It is a simple task. However, it becomes challenging if the simplicity of the noise generator is of major importance.

The research in the design of colored noise generators has been focused on the filters with infinite impulse response, as in [14–16]. The approach based on the filter banks has also been proposed [17]. On the other hand, the finite impulse response (FIR) filters are often consid-

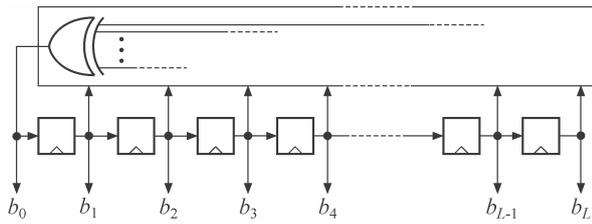


Fig. 1. Fibonacci linear feedback shift register.

ered impractical for the use in colored noise generators. Their realizations require high number of multiplications per processed sample. A high-speed implementation is therefore possible only with the cost of many general-purpose multipliers.

In this paper, we propose the design of a colored noise generator, which is based on filtering of 1-bit random signal by an FIR filter. The filtering is performed via multiplierless structure, resulting in a low complexity and high-speed implementation. The random signal is generated by a linear feedback shift register (LFSR).

The paper is organized as follows. In Section 2 some well-known facts of the LFSR are briefly summarized and the motivation for this work is given. The multiplierless realization of the proposed colored noise generator is described in Section 3. Section 4 discusses the choice of filter coefficients. Section 5 considers practical aspects of implementation. Finally, Section 6 describes the application in which the proposed generator is used to increase the SFDR of a high performance ADC.

2 FIBONACCI LFSR

Fig. 1 shows basic hardware implementation of the Fibonacci LFSR [18]. The circuit consists of a shift register whose input is fed by a combinatorial function of the predefined register taps. The combinatorial function is usually described by a mod-2 polynomial whose coefficients are 1 for the taps included in the feedback and 0 otherwise. Since the number of register states is finite, the sequence of the states is periodic. The period is determined by the register length, L , and by the used polynomial. For a given L , at least one polynomial exists which ensures maximum possible sequence length of $2^L - 1$ states. Such polynomials can be found in [19].

A common use of the Fibonacci LFSR is the generation of 1-bit (pseudo) random signal. Such a signal is taken from one of the taps, for example b_1 . For a maximum-length polynomial, each sequence period contains $2^{L-1} - 1$ zeros and 2^{L-1} ones. Apparently, for a large L , the distribution of $b_1(n)$ can be considered uniform. The power spectral density of $b_1(n)$ is flat.

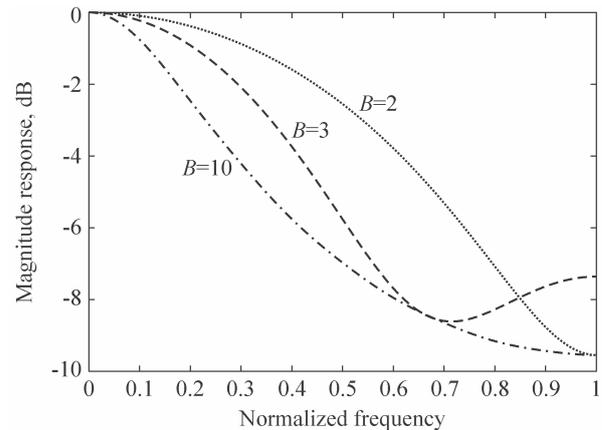


Fig. 2. Magnitude response of FIR filters whose impulse responses are decreasing sequences of powers of two, normalized to $|H(e^{j0})| = 1$.

LFSR is sometimes used to generate B -bit random signal

$$y(n) = \sum_{k=1}^B b_k(n) 2^{B-k} \quad (1)$$

Assuming $B < L$, $y(n)$ contains uniformly distributed integers from the interval $[0, 2^B - 1]$. However, its power spectral density is not flat. Namely, since $b_k(n) = b_1(n - k + 1)$, the above expression can be written as

$$y(n) = \sum_{k=1}^B b_1(n - k + 1) 2^{B-k} \quad (2)$$

Expression (2) is recognized as the convolution $b_1(n + 1) * h(n)$ where $h(n)$ is the impulse response of an FIR filter. Its transfer function is given by

$$H(z) = \sum_{k=0}^{B-1} 2^{B-k-1} z^{-k} \quad (3)$$

Fig. 2 shows the magnitude response of this filter for $B = 2, 3$ and 10 , normalized to $|H(e^{j0})| = 1$. Clearly, the response is not flat. Since it determines the power spectral density of $y(n)$, this signal is a colored noise.

This trivial example inspired us to investigate the systems in which 1-bit random signal is filtered by the FIR filter with an arbitrary impulse response. The structure suitable for the implementation of such systems is discussed in the next section. It is followed by the section considering the choice of the filter coefficients that result in a desired spectral density and approximately Gaussian distribution.

3 REALIZATION OF COLORED NOISE GENERATOR

The coefficients of the filter in (3) are decreasing powers of two. Therefore, filtering of a 1-bit signal is implemented without any hardware, simply by taking N -bit word from the LFSR. On the other hand, if the filter coefficients are arbitrary numbers, some components should be added.

The transfer function of the N th order filter is given by

$$H(z) = \sum_{k=0}^N a_k z^{-k} \quad (4)$$

A well-known implementation of the filter in (4) is based on the direct form structure [20]. This structure contains $N+1$ general-purpose multipliers and N adders. In our case, the input signal is 1 bit wide and the multipliers can be replaced by two-input multiplexers, switching between two possible products.

To avoid DC component in the input signal we formed the signal with the values -1 and 1 . Such a signal can be obtained from $b_1(n)$ as $2b_1(n)-1$. The output of the filter is then given by

$$y(n) = \sum_{k=0}^N [2b_1(n-k) - 1] a_k \quad (5)$$

The multiplierless structure for filtering random signal with values -1 and 1 is shown in Fig. 3. It requires one adder and one two-input multiplexer per filter order. Only one shift register is required since it is shared between the filter and the LFSR.

A similar structure implemented in analog technique, with adders built with operation amplifiers, was presented in the sixties [21]. In the paper referred to, the system with 16 bits long LFSR was considered, with empirically chosen filter coefficients.

4 FILTER COEFFICIENTS

4.1 Amplitude Distribution

The convolution in (5) can be considered as a sum of the signals

$$x_k(n) = [2b_1(n-k) - 1] a_k, \quad k = 0, 1, \dots, N \quad (6)$$

which are the sequences of independent identically distributed random variables. Therefore, as a consequence of the central limit theorem, $y(n)$ has the Gaussian distribution for $N \rightarrow \infty$. However, the question arises what is the distribution of $y(n)$ for a finite N . The sequences

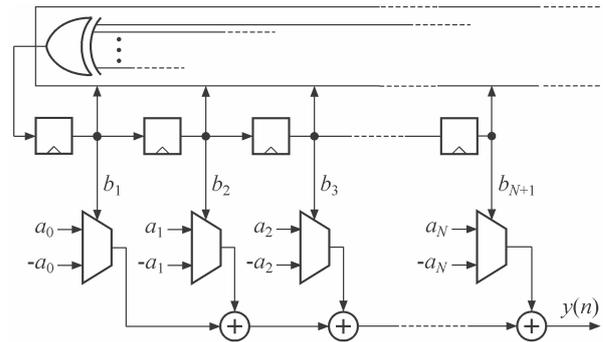


Fig. 3. Direct form structure for filtering random signal with values -1 and 1 .

in (6) have Bernoulli distributions with $p \approx 0.5$. Therefore, for an N that is not sufficiently large, the distribution of $y(n)$ might significantly differ from the Gaussian distribution. Moreover, for a small N the distribution might become bimodal or multimodal, which is not acceptable in noise generators.

The implementation of FIR filter shown in Fig. 3 is made by using a finite precision arithmetic, that is, with a_k being chosen from a finite set of values. It allows us to calculate the exact distribution of $y(n)$. For $b_1(n)$ taken from an L -bit LFSR, the probability of 1 is given by

$$p = \frac{2^{L-1}}{2^L - 1} \quad (7)$$

The distribution of $x_k(n)$ in (6) contains the values $p_1 = 1-p$ and $p_2 = p$ at the position $-a_k$ and a_k if $a_k > 0$, or $p_1 = p$ and $p_2 = 1-p$ if $a_k < 0$, as shown in Fig. 4. For $a_k = 0$, the distribution contains only the value of 1 placed at $x = 0$, and 0 otherwise. Since $y(n)$ is the sum of $x_k(n)$, $k = 0, 1, \dots, N$, the distribution of $y(n)$ is obtained as the convolution

$$P_y(x) = P_0(x) * P_1(x) * P_2(x) * \dots * P_N(x) \quad (8)$$

where $P_k(x)$ are the distributions of $x_k(n)$, $k = 0, 1, \dots, N$ [22].

Let us assume that the adders in Fig. 3 operate with S bits long words. Using two's complement representation, their operating range covers the values for x from the interval $[-2^{S-1}, 2^{S-1}-1]$. To prevent wrapping of the result in the last adder, the sum of the absolute values of all coefficients a_k should be less than or equal to $2^{S-1}-1$. Starting from the real valued samples of the impulse response, h_k , $k = 0, 1, \dots, N$, filter coefficients are obtained as

$$a_k = -\text{sgn}(h_k) \left\lceil \frac{-|h_k|}{s_{max}} (2^{S-1} - 1) \right\rceil, \quad k = 0, 1, \dots, N \quad (9)$$

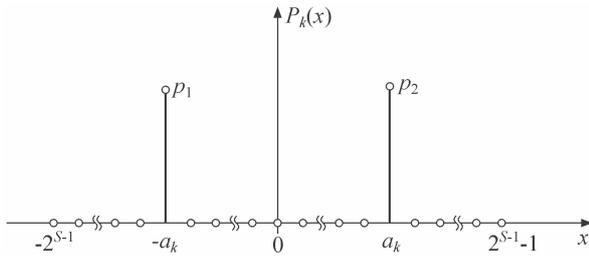


Fig. 4. Distribution of random signal with values $-a_k$ and a_k , $a_k \neq 0$.

where

$$s_{max} = \sum_{q=0}^N |h_q| \quad (10)$$

and $\lceil u \rceil$ gives the smallest integer equal to or greater than u . Using a_k in (9) the lowest negative value of -2^{S-1} is never reached. It is convenient since the interval of possible values of the final sum is symmetric around zero.

The simplest way for obtaining the convolution in (8) is calculating the partial convolutions as in

$$P_y(x) = (\dots((P_0(x) * P_1(x)) * P_2(x)) \dots * P_N(x)) \quad (11)$$

The length of $P_k(x)$, $k = 0, 1, \dots, N$, is 2^S . Therefore, the lengths of partial convolutions in (11) increase in each step by $2^S - 1$. However, using a_k in (9), the partial convolutions including the final result have zero values for $x \notin [-2^{S-1}, 2^{S-1} - 1]$. Therefore, after a partial convolution in (11) is calculated, it should be truncated by removing 2^{S-1} samples from the beginning and $2^{S-1} - 1$ samples from the end. The result of the truncation is then used to calculate the next partial convolution.

As shown in (5), $y(n)$ is obtained by adding or subtracting the coefficients a_k , which are odd and even integers. If the number of odd coefficients is odd, $y(n)$ is also odd. On the other hand, if the number of odd coefficients is even, $y(n)$ is even. This fact can be easily observed in $P_y(x)$, which always takes the value of 0 for all even or for all odd values of x . This problem can be solved by truncating the least significant bit from the result of the last addition, thus obtaining the output data width of $S-1$ bits. The distribution of the truncated signal can be easily obtained as

$$P_{out}(x_{out}) = P_y(2x_{out}) + P_y(2x_{out} + 1) \quad (12)$$

where $x_{out} \in [-2^{S-2}, 2^{S-2} - 1]$.

4.2 Transfer Function

The filter order N should be sufficiently large to allow the filter design to meet the specified magnitude response. In addition, N should be sufficiently large to ensure that

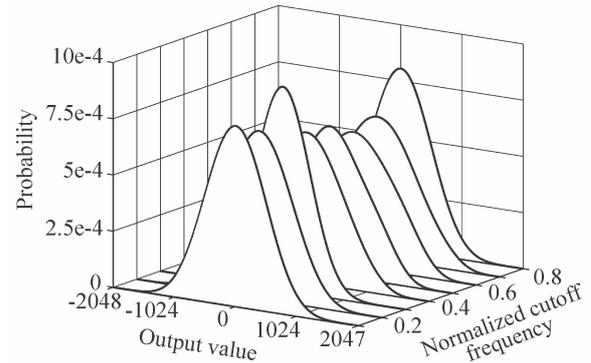


Fig. 5. Distributions of output noise of proposed generator obtained for lowpass filters of 32nd order with cutoff frequencies $\omega_c = 0.1\pi, 0.2\pi, \dots, 0.8\pi$.

the amplitude distribution of the output signal approximates the Gaussian distribution with an acceptable error. Clearly, the stronger of these two requirements should be applied. The design of the noise generator starts with the design of the filter with the required response. Then, the distribution is determined using the procedure described in Section 4.1. If the distribution does not meet the requirements, N is increased and the design procedure is repeated.

FIR filters are widely used because they allow the linear phase response, which is obtained for a symmetric impulse response. However, in our design only the magnitude response is important. Moreover, better approximation of the Gaussian distribution is expected if the filter coefficients are spread between their minimal and maximal values, instead of appearing in pairs of equal values. Therefore, we used a nonlinear phase filter design. In particular, we calculated the filter coefficients h_k , $k = 0, 1, \dots, N$, using the function `firlpnorm`, which is a part of MATLAB [23].

We made several experiments with various filters. As expected, better approximation of the Gaussian distribution was achieved for larger N . Furthermore, a large N was needed for large output resolution, that is, for a large S . Clearly, there is no straightforward estimation of the necessary filter order. Therefore, a few trials are usually needed before the satisfactory distribution is obtained.

If bandlimited noise is required, a selective filter is used. In practice, selective filters usually have the orders higher than 30. Such orders are also high enough to ensure decent distributions in most cases. Fig. 5 shows the distributions of the output noise of the proposed generator, obtained for the lowpass filters of the 32nd order, with cutoff frequencies from $\omega_c = 0.1\pi$ to $\omega_c = 0.8\pi$ in steps of 0.1π , and the transition bands of 0.1π . The filters were obtained by the function `firlpnorm`, resulting in about 1 dB of the passband and 50 dB of the stopband ripple. The

output resolution of 12 bits was chosen, what required 13 bits wide adders. A 32-bit LFSR was used as the generator of the input signal. It is clear from the figure that all distributions obtained approximate the Gaussian distribution. However, they differ in variance, which is not directly controlled in the proposed design.

5 PRACTICAL CONSIDERATIONS

The structure in Fig. 3 contains a long combinatorial function, which includes the entire chain of adders. To achieve proper functionality at a high-frequency system clock, the circuit should be pipelined. The highest speed is achieved if the pipeline registers are inserted between all adders, as shown in Fig. 6. The delays introduced between the adders should also be introduced in signals $b_k(n)$. It can be easily accomplished by shifting the connection of multiplexers from tap b_k to tap b_{2k-2} for $k > 2$.

For a given set of filter coefficients, the lengths of the adders in Fig. 3 can be optimized. The first adder operates with the coefficients a_0 and a_1 . Therefore, the absolute value of the largest sum is $|a_0| + |a_1|$, requiring the adder of $\lceil \log_2(|a_0| + |a_1|) \rceil + 1$ bits. The value of 1 is added since the coefficients in the filter are represented using two's complement. Generally, the length of the adder following the tap b_k is given by

$$W_k = \left\lceil \log_2 \left(\sum_{q=0}^{k-1} |a_q| \right) \right\rceil + 1, \quad k = 2, \dots, N+1 \quad (13)$$

The same approach can be applied to obtain the widths of the adders and registers in pipeline structure in Fig. 6.

The circuits in Fig. 3 and Fig. 6 can be further simplified by rearranging the filter coefficients. Namely, filter's magnitude response is not changed if the impulse response is reversed in time. Therefore, the order of $a_k, k = 0, 1, \dots, N$, that minimizes the total adders' length should be used.

As shown in Section 4.1, the coefficients $a_k, k = 0, 1, \dots, N$, are obtained by truncation of the real valued coefficients h_k to the length of S bits. Such a truncation deteriorates the frequency response of the original filter. Therefore, S should be chosen high enough to ensure an acceptable frequency response. However, a high S results in a high output word length of $S-1$ bits. Therefore, the output samples are often additionally truncated to the length of $D < S-1$ bits, to meet the requirements of the application at hand. Note that such a truncation adds quantization noise. Since this noise is uniformly spread along the Nyquist band, it increases but it does not change the original shape of the output power spectral density. However, D should be chosen high enough to ensure an acceptable increase.

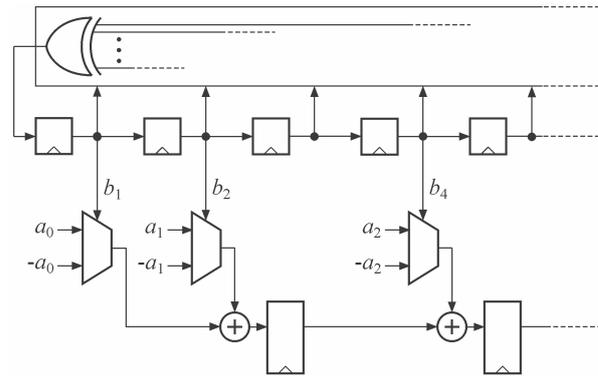


Fig. 6. Pipelined direct form structure for filtering random signal with values -1 and 1 .

The structure in Fig. 6 is suitable for efficient implementation on various platforms. However, further modifications are possible to tailor the circuit for the platform at hand. For example, the multiplexers used for selecting the positive or negative coefficients may be omitted if adder/subtractor circuits are used instead of simple adders. Note that both solutions are suitable for the implementation on field programmable gate arrays (FPGAs) because they can be mapped in the look-up-table (LUT) resources of the full adders. Moreover, the flip-flops used for pipelining are usually included in the slices which realize the adders, what results in a compact circuit.

6 APPLICATION OF NOISE GENERATOR

Modern ADCs exhibit excellent dynamic performances. For example, the SFDR of up to 90 dBc [24] is achieved for signals approaching converter's full scale. Unfortunately, with the decrease of the input signal, the SFDR also decreases (see, for example, Fig. 21 in [24]). It is a consequence of the spurs, which in ADCs do not significantly decrease with the decrease of the input signal. Operational amplifiers and other components surrounding the ADC behave differently. Their SFDRs remain the same or even increase with the decrease of the input signal. Therefore, any improvement obtained in ADC's SFDR pays off.

The spurs in an ADC depend on the distribution of bad codes in ADC's transfer function [11]. Adding a random signal into the ADC's input randomizes the addressing of these codes. In frequency domain it results in spreading of discrete spurs along the entire Nyquist band and, consequently, in improving the dynamic performances. Such an approach is effective in dealing with small, as well as with large scale nonlinearities, which are both present in ADC's transfer functions [11].

Many systems for digital signal processing deal with signals whose bands do not occupy the entire Nyquist

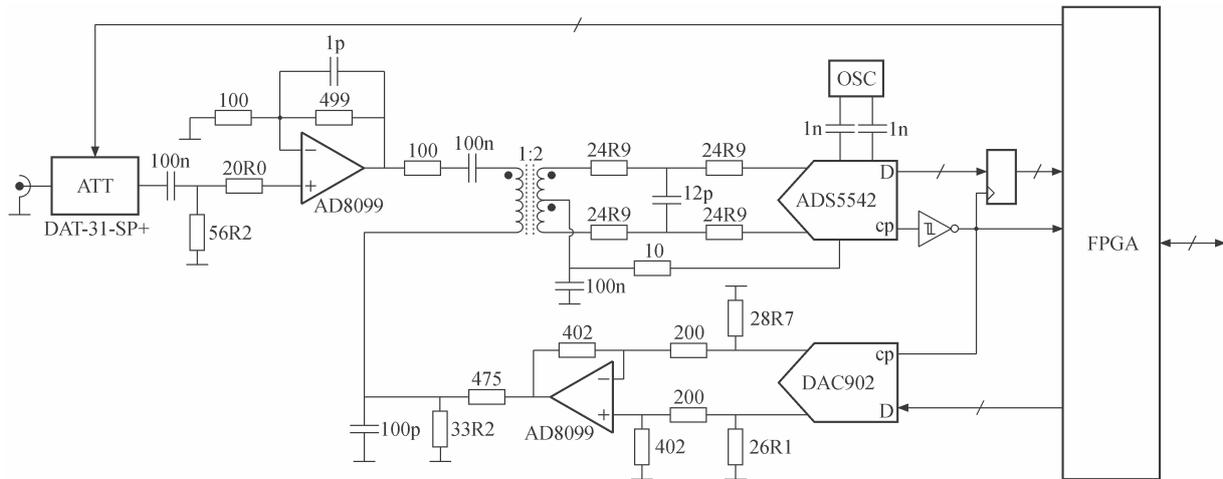


Fig. 7. Front end of high dynamic system for digital processing of continuous-time signals.

band. Such is the case in spectral analyzers which rarely use the band around zero. On the other hand, the radio receivers simultaneously process only a limited number of channels. Moreover, the band around the Nyquist frequency is beyond of interest in both systems because it is usually covered by the transition band of the antialiasing filter. Clearly, these systems are suitable for the use of colored noise playing the role of dither signal. The following text describes the application in which a lowpass dither is used.

6.1 Analog Front End

Fig. 7 shows the front end of a high-dynamic system for digital processing of continuous-time signals. Analog to digital path consists of digital step attenuator (ATT) [25], input amplifier, transformer and high performance analog to digital converter ADS5542 [24]. Dither signal is injected into the transformer. We used a similar system as the front end of a high dynamic radio receiver described in [10]. In the paper referred to, an analog filter was used for coloring the noise. Here we use the noise generator described in previous sections. The digital part of the noise generator is implemented on a Spartan-3 FPGA device XC3S1500-4FG676 [26]. It is followed by digital to analog converter DAC902 [27] and the corresponding amplifier.

The circuit in Fig. 7 is used for processing signals with a dynamic range of more than 120 dB. To achieve such performances the circuit's layout should be performed carefully, with shielding to prevent crosstalk from digital to analog part of the system.

Fig. 8 shows the prototype of the analog front end. The entire circuit from Fig. 7 except the FPGA device is placed on a single two-layer printed circuit board together with the

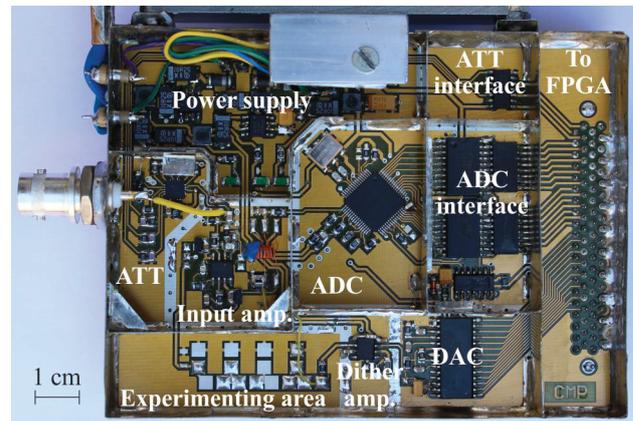


Fig. 8. Prototype of front end of high dynamic system for digital processing of continuous-time signals (top shield removed).

corresponding power supplies. The crosstalk from digital subsystem in the prototype is obtained below the level of -127 dBFS.

6.2 Implementation of Dither Generator

In the design of dither generator three issues have to be addressed: the bandwidth within which the noise is injected into the ADC, the level of the injected signal, and the overall noise level within the band of interest. In this example, the noise is injected within the band $|\omega| \leq 0.1\pi$, assuming the signal of interest occupies the rest of the Nyquist band. Therefore, the generator implements a low-pass filter. The shape of the power spectral density of the injected noise is not critical. Therefore, any lowpass approximation with reasonable wide transition band can be used.

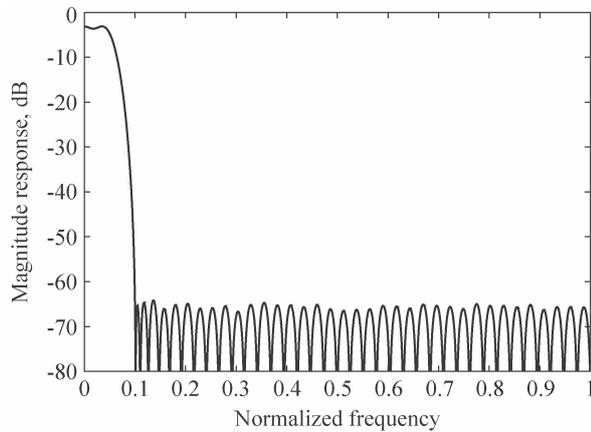


Fig. 9. Magnitude response of 75th-order filter with 17-bit coefficients, which is implemented in noise generator.

Table 1. Coefficients of filter used in implemented noise generator

$a_k, k = 0, \dots, 75$
-30, -33, -49, -69, -93, -121, -151, -183, -215, -247,
-277, -302, -320, -330, -329, -316, -289, -248, -191,
-119, -33, 64, 172, 286, 404, 520, 630, 728, 810, 870, 904, 909,
881, 818, 719, 584, 415, 215, -13, -264, -531, -810, -1091,
-1369, -1636, -1885, -2110, -2306, -2468, -2593, -2678,
-2723, -2728, -2694, -2625, -2524, -2394, -2242, -2071,
-1888, -1698, -1506, -1316, -1133, -959, -799, -653,
-523, -410, -314, -233, -167, -116, -76, -46, -34

Dealing with high-scale nonlinearities requires high-level dither signals. Unfortunately, a high level dither decreases the maximum range of the input signal. Therefore, the level of the dither is a tradeoff between the decrease of the input range and the improvement achieved in SFDR. In this application we assume that the decrease of input range of 1 dB is acceptable.

Digital noise generator described in previous section provides the samples from the range $[-2^{D-1}, 2^{D-1}-1]$ thus covering all codes of a D -bit digital to analog converter. The amplifier following DAC, shown in Fig. 7, is designed to provide the noise voltage of $2 V_{pp}$. The differential input voltage of the ADC is $2.3 V_{pp}$, which corresponds to $1.15 V_{pp}$ at the primary coil of the transformer. The decrease of input dynamic range of 1 dB is therefore achieved for dither signal of $0.13 V_{pp}$. Such dither signal is obtained by the divider with the attenuation of $A_a = 0.065$, which follows the amplifier. Note that the differential input impedance of the ADC is $6.6 \text{ k}\Omega$. Therefore, we assume here that the impedance observed at the primary coil is high enough to cause negligible voltage drop of the injected signal.

The SNR of the used ADC is 74 dBFS [24], what results in equivalent noise voltage of

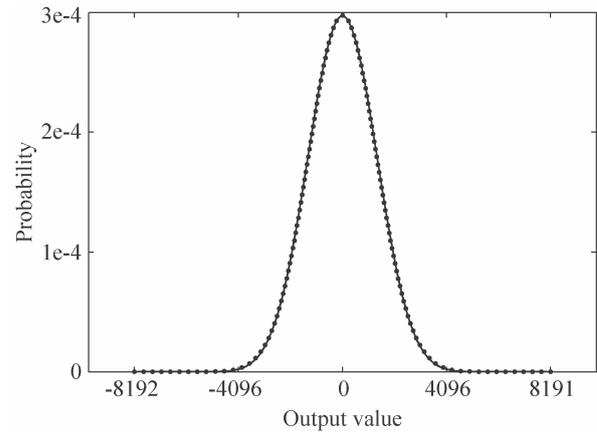


Fig. 10. Distribution of 14-bit output signal of implemented noise generator (thin solid line) and Gaussian distribution with $\sigma^2 = 1.797e6$ (thick dotted line).

$U_{ADCnoise} = 8.113e-5 V_{rms}$ at the transformer's primary coil. The dither signal is obtained from the signal which takes the values of -1 and 1 . This signal is filtered by the filter whose gain in the stopband is A_f , and then attenuated by the factor A_a . So obtained signal is added to $U_{ADCnoise}$, resulting in an increase of the noise level. The increase of the noise level achieved within filter's stopband is equal to

$$\Delta P = 10 \log_{10} \left[1 + \left(\frac{1 V_{rms} \cdot A_f \cdot A_a}{U_{ADCnoise}} \right)^2 \right] \text{ dB} \quad (14)$$

The value of A_f which results in the increase of system's noise floor of ΔP can be easily obtained from (14) as

$$A_f = \frac{U_{ADCnoise}}{1 V_{rms} \cdot A_a} \cdot \sqrt{10^{\Delta P/10} - 1} \quad (15)$$

The amount of ΔP which can be tolerated depends on the application at hand. For example, in spectral analysis the raise of the noise floor of 3 dB is rarely a problem because it can be compensated with the processing gain of FFT, simply by doubling the number of samples being processed. However, in a radio receiver, a large increase of the noise floor is not tolerable because it decreases receiver's sensitivity. We targeted the value of $\Delta P = 1 \text{ dB}$. It results in $A_f = 6.351e-4$, that is, in the stopband attenuation of 63.9 dB.

The noise generator was implemented using circuit in Fig. 6 with the adders operating with the word lengths of $S = 17$ bits, output resolution of $D = 14$ bits, and the length of the LFSR of $L = 36$ bits. Assuming the stopband begins at the frequency $\omega_s = 0.1\pi$, the filter was designed with the passband cutoff $\omega_c = 0.044\pi$. The required attenuation is obtained for the filter of the 75th order.

Table 2. Complexity of proposed colored noise generator and its comparison with existing white noise generators, all implemented on Xilinx FPGA devices

Design	this work	this work	Gutierrez [4]	Cheung [5]
Device	XC3S1500	XC6VLX240T	XC5VLX110T	XC4VLX100
Speed grade	-4	-3	-3	-12
Precision	16	16	16	16
LUTs	1105	1105	421	832
Flip-Flops	1255	1255	-	-
Multipliers	0	0	1	1
BRAMs	0	0	4	4
f_{\max} , MHz	199	683	462	450

The set of quantized filter coefficients used in dither generator is given in Table 1 and the corresponding magnitude response is shown in Fig. 9. Note that the passband attenuation of 3.1 dB is a consequence of the normalization of maximum output value in (9) and (10). The value of 63.9 dB is targeted as the lowest stopband attenuation. However, the stopband attenuation is approximated with ripple. Therefore, the noise floor is increased less than 1 dB within the major part of filter's stopband. The amplitude distribution obtained by the coefficients from Table 1 approximates Gaussian as shown in Fig. 10.

In the above design we assumed that the noise level within the band of interest is not increased by truncating the generator's output to the word length of $D = 14$ bits. It is correct because the noise introduced by 14-bit quantization is negligible compared to the noise generated within the filter's stopband. The use of the word length of 14 bits is therefore more than sufficient. However, a high-performance DAC is chosen because of its high linearity. Namely, if the linearity of the DAC is not sufficiently high, the components from the dither band produce spurs which might fall into the band of interest and additionally increase the noise floor. In our case this effect is not encountered because the used DAC has the SFDR which is in dither band higher than 70 dBc.

The proposed generator is able to generate noise with arbitrary shape of power spectral density, depending on the used filter. Furthermore, it can be easily modified to store the filter coefficients in registers, thus changing the band of interest while the system is operating. However, moving the dither to high frequencies should be carefully considered because the SFDR of digital to analog converters generally decreases with approaching the Nyquist frequency.

Table 2 shows the resources occupied by the implementation of the described noise generator on Spartan-3 [26] and Virtex-6 [28] FPGA devices. The results are given for fully pipelined structure shown in Fig. 6. Note that such a structure implements FIR filter which operates with data clock equal to the systems clock. However, the price

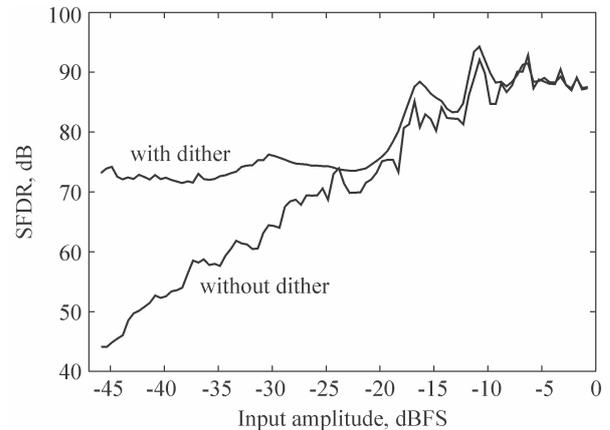


Fig. 11. Spurious free dynamic range of high dynamic system for digital processing of continuous-time signals, measured without dither and with dither added in the band $|\omega| < 0.1\pi$.

is paid in a high number of flip-flops. For less demanding applications, the structure without pipeline shown in Fig. 3 might be used, which requires only 96 flip-flops. Partial pipelining is also possible to achieve a compromise between the operating speed and the complexity.

The last two columns in Table 2 show the implementation results of the noise generator presented in [4] and [5]. Clearly, these results are obtained for white-noise Gaussian sources. However, we included them in the comparison to illustrate that color noise generators can also be implemented with reasonable complexity and high operating speed, which are comparable to those obtained by high-performance white-noise generators.

6.3 Results

In this section we provide the results of measurements which we performed to verify the suitability of the described noise generator for mitigation of the effects of nonlinearities in analog to digital converters. In these measurements the input attenuator was not used to avoid the influence of its nonlinearity.

The measurements were carried out using a sine input signal with the frequency of 4 MHz and spurious components below 96 dBc. Its level varied from -1 dBFS to -46 dBFS with the step of 0.5 dB. The analysis was performed in the frequency domain, using the FFT with 4 million points. Keeping in mind the ADC's SNR is 74 dB, such an FFT length allows the estimation of the level of the harmonic components down to -120 dBFS. The rectangular window was used, and the interpolation with factor 10 was applied to decrease the scalloping loss.

Fig. 11 shows the spurious free dynamic range of the described system obtained with and without dither. The

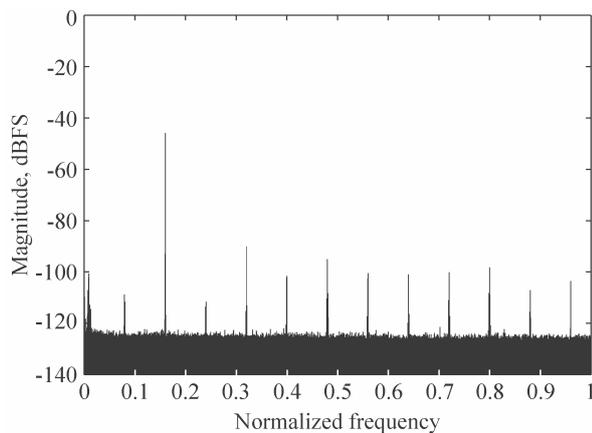


Fig. 12. Magnitude spectrum of signal sampled without dither.

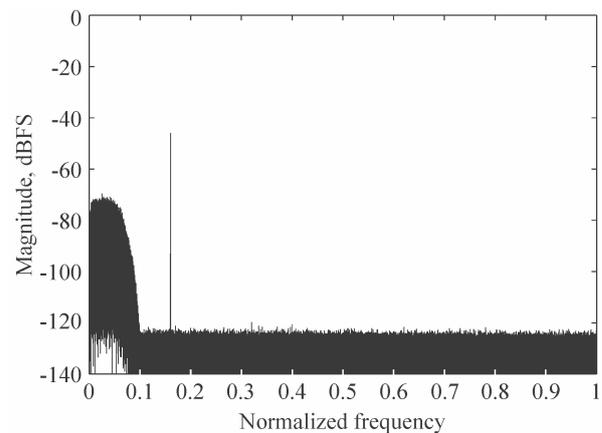


Fig. 13. Magnitude spectrum of signal sampled with dither.

SFDR obtained without dither decreases rapidly with the decrease of the input signal. If dither is added, the SFDR is improved. The improvement is significant for signals whose level is less than the dither level. It is acceptable because the SFDR is already high for the signals approaching ADC's full scale. Improvement in SFDR generally increases with the decrease of input signal and exceeds 25 dB for the signals around -45 dBFS. Further increase is expected for smaller signals. However, such a case is rarely interesting in practice.

Fig. 12 and Fig. 13 show the magnitude spectrum of the signal with the level of -46 dBFS, sampled with and without dither. It is clear from the figures that the dither spreads the spurs along the entire Nyquist band, thus increasing the SFDR.

7 CONCLUSION

Colored noise with approximately Gaussian distribution can be efficiently generated by filtering of 1-bit random signal. A generator was presented which has a simple structure, containing no general-purpose multipliers and no memories. The structure can be easily pipelined resulting in a high-speed implementation.

The presented generator is suitable for applications in which colored noise is required with arbitrary power spectral density, as for example in generators of dither signals.

ACKNOWLEDGMENT

The work described in this paper was supported by the Ministry of Science, Education and Sports of the Republic of Croatia under grant no. 036-0362214-2217, and the ACROSS project funded by the Seventh Framework Programme of the European Union under grant no. 285939 FP7-REGPOT-2011-1.

REFERENCES

- [1] D.-U. Lee, J. D. Villasenor, W. Luk and P. H. W. Leong, "A Hardware Gaussian Noise Generator Using the Box-Muller Method and Its Error Analysis," *IEEE Trans. Comput.*, vol. 55, no. 6, pp. 659–671, June 2006.
- [2] D. B. Thomas, W. Luk, P. H. W. Leong and J. D. Villasenor, "Gaussian Random Number Generators," *J. ACM Comput. Surveys (CSUR)*, vol. 39, no. 4, pp. 11:1–11:38, Oct. 2007.
- [3] J. S. Malik, J. N. Malik, A. Hemani and N. D. Gohar, "Generating High Tail Accuracy Gaussian Random Numbers in Hardware Using Central Limit Theorem," in *Proc. 2011 IEEE/IFIP Int. Conf. VLSI System-on-Chip*, Hong Kong, China, pp. 60–65, Oct. 2011.
- [4] R. Gutierrez, V. Torres and J. Valls, "Hardware Architecture of a Gaussian Noise Generator Based on the Inversion Method," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 59, no. 8, pp. 501–505, Aug. 2012.
- [5] R. C. C. Cheung, L. Dong-U, W. Luk and J. D. Villasenor, "Hardware Generation of Arbitrary Random Number Distributions From Uniform Distributions Via the Inversion Method," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 8, pp. 952–962, Aug. 2007.
- [6] M. D. McDonnell, "Is Electrical Noise Useful?," *Proc. IEEE*, vol. 99, no. 2, pp. 242–246, Feb. 2011.
- [7] R. A. Wannamaker, S. P. Lipshitz, J. Vanderkooy and J. N. Wright, "A Theory of Nonsubtractive Dither," *IEEE Trans. Signal Process.*, vol. 48, no. 2, pp. 499–516, Feb. 2000.
- [8] H. Pan and A. A. Abidi, "Spectral Spurs due to Quantization in Nyquist ADCs," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 51, no. 8, pp. 1422–1439, Aug. 2004.
- [9] N. C. Davies, "A High Performance HF Software Radio," in *Proc. 8th Int. Conf. HF Radio Systems and Techniques*, Guildford, U.K., pp. 249–256, 2000.
- [10] M. Vucic, G. Molnar and M. Butorac, "FPGA Implementation of High-Frequency Software Radio Receiver," in

- Proc. 2009 IEEE Int. Symp. Circuits Syst.*, Taipei, Taiwan, pp. 1040–1043, May 2009.
- [11] B. Brannon, “Overcoming Converter Nonlinearities with Dither,” Appl. Note AN-410, Analog Devices, 1995.
- [12] B. Brannon and R. Reeder, “Understanding High Speed ADC Testing and Evaluation,” Appl. Note AN-835, Analog Devices, 2006.
- [13] M. Levin, “Generation of a Sampled Gaussian Time Series Having a Specified Correlation Function,” *IRE Trans. Information Theory*, vol. 6, no. 5, pp. 545–548, May 1960.
- [14] S. M. Kay, “Efficient Generation of Colored Noise,” *Proc. IEEE*, vol. 69, no. 4, pp. 480–481, Apr. 1981.
- [15] D. M. Brookes, “Coloured Noise Generation Without Filter Startup Transient,” *Electron. Lett.*, vol. 37, no. 4, pp. 255–256, Feb. 2001.
- [16] D. Derrien and E. Boutillon, “Gaussian Noise Generator Hardware Implementation Based on Statistical Properties,” in *Proc. IEEE Int. Symp. Signal Processing and Information Technology*, 2002.
- [17] K. Mochizuki and M. Uchino, “Efficient Digital Wide-Band Coloured Noise Generator,” *Electron. Lett.*, vol. 37, no. 1, pp. 62–64, Jan. 2001.
- [18] M. Goresky and A. Klapper, *Algebraic Shift Register Sequences*. Cambridge, UK: Cambridge University Press, 2012.
- [19] P. Alfke, “Efficient Shift Registers, LFSR Counters, and Long Pseudo-Random Sequence Generators,” Xilinx Appl. Note XAPP 052, 1996.
- [20] S. K. Mitra, *Digital Signal Processing – A Computer-Based Approach*. New York, NY: McGraw-Hill, 2010.
- [21] S. B. Matthews, “Generation of Pseudorandom Noise Having a Gaussian Spectral Density,” *IEEE Trans. Comput.*, vol. 17, no. 4, pp. 382–385, Apr. 1968.
- [22] P. Z. Peebles, Jr., *Probability, Random Variables, and Random Signal Principles*. New York, NY: McGraw-Hill, 2000.
- [23] *MATLAB: Filter Design Toolbox*, MathWorks, Inc., Natick, MA, USA, 2010.
- [24] *ADS5542 14-Bit, 80 MSPS Analog-To-Digital Converter Data Sheet*. Texas Instruments Inc., Dallas, TX, USA, rev. Feb. 2007.
- [25] *DAT-31-SP+ Digital Step Attenuator Data Sheet*. Mini-Circuits, New York, NY, USA, Rev. G.
- [26] *Spartan-3 FPGA Family Data Sheet*. Xilinx Inc., San Jose, CA, USA, 2008.
- [27] *DAC904 14-Bit, 165MSPS Digital-To-Analog Converter Data Sheet*. Texas Instruments Inc., Dallas, TX, USA, May 2002.
- [28] *Virtex-6 Family Overview*. Xilinx Inc., San Jose, CA, USA, Jan. 2012.



Mladen Vučić was born in Karlovac, Croatia, in 1965. He received the Diploma Engineer, M.Sc., and Ph.D. degrees in electrical engineering from the University of Zagreb, Zagreb, Croatia, in 1989, 1993 and 1999, respectively. Since 1989, he has been with the Department of Electronic Systems and Information Processing of the Faculty of Electrical Engineering and Computing, University of Zagreb. Currently, he is a Professor of Electrical Engineering. His research interests include circuit theory, analog and digital signal processing, optimization theory and applications, digital system design and embedded systems. He has published 46 papers in journals and conference proceedings. He is a member of KoREMA and IEEE.



Marko Butorac was born in Rijeka, Croatia in 1984. He received the Diploma Engineer degree in electrical engineering from the University of Zagreb, Zagreb, Croatia, in 2007. He is currently working toward the Ph.D. degree in electrical engineering at the same university. Since 2007, he has been a Research Assistant with the Department of Electronic Systems and Information Processing of the Faculty of Electrical Engineering and Computing, University of Zagreb. His interests include digital signal processing, software defined radio, digital system design and embedded systems.

AUTHORS' ADDRESSES

Prof. Mladen Vučić, Ph.D.
Marko Butorac, Dipl. Ing.
University of Zagreb,
Faculty of Electrical Engineering and Computing,
Department of Electronic Systems and Information Processing,
Unska 3, 10000 Zagreb, Croatia
email: mladen.vucic@fer.hr, marko.butorac@fer.hr

Received: 2013-05-20
 Accepted: 2013-09-02