

Reconfigurable Readback-Signal Generator Based on a Field-Programmable Gate Array

Jinghuan Chen, Jaekyun Moon, and Kia Bazargan, *Member, IEEE*

Abstract—We have designed a readback-signal generator to provide noise-corrupted signals to a read channel simulator. It is implemented in a Xilinx Virtex-E field-programmable gate array (FPGA) device. The generator simulates in hardware the noise processes and distortions observed in hard drives. It uses embedded nonuniform random number generators to simulate the random characteristics of various disturbances in the read/write process. The signal generator can simulate readback pulses, intersymbol interference, transition noise, electronics noise, head and media nonlinearity, intertrack interference, and write timing error according to the characteristics specified by the user. A sample implementation operates at a 70-MHz clock speed. The design can easily be scaled for different error rates. The generator can be reconfigured in real time to give the user flexibility and increase the capacity of the FPGA device. The readback-signal generator can be integrated into an FPGA read channel simulator or serve as a test bench for data-recovery circuits.

Index Terms—Field programmable gate array (FPGA), Gaussian pseudorandom number generator, magnetic recording, read channel, read channel modeling, reconfigurable computing, transition noise.

I. INTRODUCTION

INTENSIVE simulation is often carried out to investigate advanced signal processing techniques for data storage applications. The simulation is normally performed in software that may take a very long time. For instance, the simulation of a 10^{-8} bit-error rate (BER), assuming 10 bit errors are observed, can take days running on a personal computer equipped with a 1-GHz Pentium IV processor. On the other hand, the target sector retry rates [i.e., unrecoverable error rates after error-correction-code (ECC)] in commercial hard drives normally go below 10^{-12} for desktop products and 10^{-14} for server-class products. One approach to speed up the simulation process is to implement the whole simulator or part of it in hardware. Because the noise characteristics in hard drives are unique, the additive white Gaussian noise (AWGN) assumption, which is widely accepted in studying the performance of many communication systems, is not reliable. Instead, a dedicated readback-signal generator must be implemented in front of data-recovery modules in the simulator. This signal generator shall incorporate all major noise processes and distortions and be capable of

generating very low probability events according to the user-defined statistics. It shall also have the reconfiguration capability that allows the user to choose from various noise and distortion combinations. Field-programmable gate array (FPGA) becomes the choice of implementation platform because of its advantage of low-cost and reconfigurability. The FPGA-based signal generator proposed in this paper can also serve as a test bench for data-recovery circuits (usually called the “read channel”) since it is completely tunable in contrast to typical test spindle setups and is able to provide a complete set of test conditions.

A series of noise processes and distortions in hard drives degrades the performance of read channels. Statistical models of these noise processes, definition of signal-to-noise ratio (SNR), and analysis of quantization effect are provided in Section II. In Section III, the design and performance of embedded random number generators are explained. These generators produce random numbers according to the statistics specified by the user. In Section IV, we present the architecture of the design and briefly discuss time-duplexing and JBits-aided reconfiguration. In Section V, we demonstrate the noise and nonlinearity statistics in the generated signal. Conclusions are given in Section VI.

II. MODELS AND ALGORITHMS

In hard drives, each binary datum is stored in a very small area, called a bit cell, on the magnetic surface of a disk. A write head magnetizes each bit cell to one of two directions that represent 0 or 1. A read head picks up the magnetization flux emitted from the boundaries of bit cells and generates readback pulses. These pulses will be passed through a preamplifier, analog/digital filters, and data-recovery modules and converted back to binary data.

The noisy environment in hard drives is unique and harsh. These noise processes and distortions can be categorized into intersymbol interference, transition noise, electronics noise, head/media nonlinearity, timing error, and intertrack interference. Noise processes in hard-drive recording systems have been modeled in several ways. Moon [1] uses Taylor series to approximate transition jitter and width variation. Caroselli and Wolf [2] simplify the structure of recorded tracks by using a micro-track model. In our implementation, we mimic the physical reality by actually shifting the pulse position and changing the pulse shape in runtime. The behavior of the various noise processes and distortions follows the statistics laid down by the user. It should be noted that the statistics are defined by the user and do not have to be Gaussian, as has been assumed in many previous modeling works.

Manuscript received December 8, 2003; revised February 10, 2004.

J. Chen was with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA. He is now with Maxtor Corporation, Shrewsbury, MA 01545 USA (e-mail: jinghuan_chen@maxtor.com).

J. Moon and K. Bazargan are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: moon@ece.umn.edu; kia@ece.umn.edu).

Digital Object Identifier 10.1109/TMAG.2004.826913

A. Statistical Modeling of Noises and Distortions

1) *Intersymbol Interference (ISI)*: Since the read heads are band-limited sensors, readback signals will experience ISI when the recording symbol rate ($1/T$) exceeds the head bandwidth W . A sequence of readback pulses overlap and are distorted when ISI extends over a number (I) of symbols. The noiseless readback signal is given by

$$y(t) = \sum_k a_k h(t - kT) \quad (1)$$

$$a_k = b_k - b_{k-1} \quad (2)$$

where $h(t)$ is the isolated pulse and a_k is the difference of two consecutive data bit $b_k \in \{-1, 1\}$ (2). The sequence a_k takes value in $\{-2, 0, 2\}$ and a nonzero a_k indicates the existence of a transition. ISI happens when the length of pulse $L = I \cdot T$ is longer than the symbol period T .

2) *Transition and Electronics Noise*: The zigzag boundary between two opposing magnetizations causes the position of a readback pulse to shift in time (*transition jitter*) and its shape to deform (*width variation*) [3]. The noise introduced by this sort of phenomenon is called *transition noise* (TN) and is data dependent in the sense that noise arises only in the presence of transitions that make up a data pattern. Both of these noise processes can be modeled using random variables. We rewrite (1) to reflect the transition noise. Now the isolated pulse $H(t)$ is not deterministic but a random process where the sample space is a series of pulses that have different pulse width and amplitude. We note that a wider pulse has reduced amplitude so as to make the area under the pulse remains constant. This is necessary if the pulse widening is due to the broadening of the magnetic transition. Position jitter δ_j is a random variable falling in the range of $(-T, T)$, where we assume T is the limit of jitter on each side.

Electronics noise $n_e(t)$ normally is modeled as an additive white Gaussian noise (AWGN) band-limited by W . Its power spectral density is $(N_0/2)$. Now the noise-corrupted read signal is written as

$$y(t) = \sum_k a_k H(t - kT + \delta_j) + n_e(t). \quad (3)$$

3) *Head and Media Nonlinearities: Magnetoresistance (MR) read heads* are not linear sensors. When incorrectly biased, they can cause an unsymmetrical head sensitivity function [4]. This effect is called the *head nonlinearity* (HNL). It amplifies the readback signal in a nonlinear fashion across the range of amplitude and results in disparity between positive and negative pulses. It can be characterized by a nonlinear function $F(x)$, which can be measured from a read head.

Two more types of media nonlinearity also occur during the write process: *nonlinear transition shift* (NLTS) and *partial erasure* (PE). They generally create additional transition position shift and amplitude loss. Simplistic assumptions have been made on the data dependence in [5] and [6]. We consider only the first-order PE effect and combine them as follows:

$$y(t) = F \left\{ \sum_k r_k a_k H \left(t - kT + \delta_j - \frac{\varepsilon T}{4} a_k a_{k-1} \right) \right\} \quad (4)$$

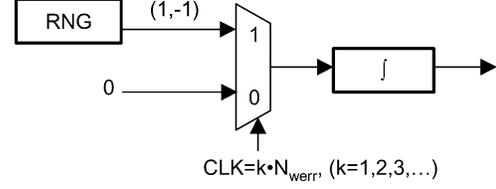


Fig. 1. Generation of write timing error.

where r_k is the PE parameter and εT is the amount of nonlinear transition shift. The NLTS component reduces to zero effectively unless two consecutive transitions occur, i.e., $a_k a_{k-1} \neq 0$.

4) *Write Timing Error*: A slowly varying write timing error (WERR) is added to model accumulated phase jitter in the write-clock synthesizer circuit. It can be simplified and modeled as a slow-paced “*random walk*.” As shown in Fig. 1, one write error is uniformly generated from $\{1, -1\}$ in every N_{werr} clock cycles. The effect is integrated so it will influence the write position of all later pulses. The average of the drift is zero in a very long run and the pace is controlled by N_{werr} .

5) *Intertrack Interference*: A read head does not always position itself with 100% accuracy on the top of a track. It consequently senses the signal from the neighboring track and is subject to the *crosstalk*, which is also called *intertrack interference* (ITI). ITI is brought into the picture by adding up two weighted output signals from two independent generators.

B. SNR and Statistics of Noise Processes

The performance of data-recovery schemes is usually compared across a range of SNR. A relationship between SNR and statistics of generated random noise shall be established so that the effectiveness of signal processing effort can be evaluated.

We adopt the SNR definition in [7] because it includes data-dependent transition noise while eliminating the dependency on symbol density. We briefly rephrase the derivation as follows:

$$\begin{aligned} \text{SNR} &= \frac{E_t}{M_0 + N_0} \\ &= \frac{E_t}{M_0} \cdot \frac{\alpha}{100} \end{aligned} \quad (5)$$

where E_t is the energy of an isolated pulse given as

$$E_t = \int_{-\infty}^{\infty} h(t)^2 dt \quad (6)$$

M_0 is twice the average energy of the transition noise in each transition, $(N_0/2)$ is the power spectral density of the electronics noise, and α is the percentage of transition noise in total noise

$$\alpha = \frac{M_0}{M_0 + N_0} \cdot 100(\%). \quad (7)$$

Assuming the position jitter and width variation are uncorrelated, we can decompose M_0 as

$$M_0 = M_{0,j} + M_{0,w} \quad (8)$$

where $M_{0,j}$ and $M_{0,w}$ are the power of position jitter and width variation, respectively. When the *probability density function* (PDF) or *cumulative distribution function* (CDF) of position

jitter and width variation are predefined, M_0 can be easily calculated and, hence, the SNR is known.

The statistics of simulated noise processes can be determined in two ways. The user can provide the CDFs of the noise processes and the corresponding SNR can be calculated. Or, when the observed noise can be satisfactorily approximated as the Gaussian distribution, in which the first- and second-order statistics can provide all the necessary information about the noise process, the user of the signal generator will only need to specify the SNR. Now the SNR becomes an input that effectively defines the variance of the noise. Additionally, the user shall define the undistorted isolated pulse $h(t)$, the percentage of the transition noise power, and composition of the position jitter and the pulse variation. We will focus on the second approach in the following paragraphs since it is widely used in the modeling of readback signals.

As given in [7], when we assume first-order transition noise, we have

$$M_0 = 2\sigma_j^2 I_t + 2\sigma_w^2 I_w \quad (9)$$

where σ_j^2 and σ_w^2 are the variances of transition position jitter and width variation, respectively. I_t and I_w are integrals defined as follows:

$$I_t = \int_{-\infty}^{\infty} \left[\frac{\partial h}{\partial t} \right]^2 dt \quad (10)$$

$$I_w = \int_{-\infty}^{\infty} \left[\frac{\partial h}{\partial w} \right]^2 dt. \quad (11)$$

The quantity of two components in M_0 is controlled by a parameter that specifies the jitter noise power expressed as a function of the total medium noise power P_j

$$P_j = \frac{2\sigma_j^2 I_t}{M_0}. \quad (12)$$

From (5)–(12), we can obtain the variance of the noise processes as

$$\sigma_j^2 = \frac{P_j \cdot E_t \cdot \alpha}{2I_t \cdot \text{SNR}} \quad (13)$$

$$\sigma_w^2 = \frac{(1 - P_j) \cdot E_t \cdot \alpha}{2I_w \cdot \text{SNR}} \quad (14)$$

$$\sigma_e^2 = \frac{E_t \cdot (100 - \alpha)}{2\text{SNR}} \cdot \frac{1}{W}. \quad (15)$$

All random variables we have modeled till now are continuous-time. In the FPGA signal generator, which is a digital device, each random variable will be represented by a finite number of bits. Loss associated with this quantization process will be discussed in the next section. If we represent each symbol period T with n_T discrete numbers and when $n_T \gg 1$, E_t , I_t , and I_w can be approximated to

$$E_t \cong \sum_{i=1}^L h_{1,i}^2 \cdot \Delta t \quad (16)$$

$$I_t \cong \sum_L \left[\frac{\Delta h}{\Delta t} \right]^2 \Delta t = \sum_{i=1}^L (h_{1,i+1} - h_{1,i})^2 \frac{1}{\Delta t} \quad (17)$$

$$I_w \cong \sum_L \left[\frac{\Delta h}{\Delta w} \right]^2 \Delta t = \sum_{i=1}^L \frac{(h_{2,i} - h_{1,i})^2}{\Delta w^2} \Delta t \quad (18)$$

where $h_{1,i}$ and $h_{2,i}$ are the first (undistorted) and second pulses in the distorted pulse lookup table, respectively, Δt is the unit shift of the pulse, and Δw is the unit change of pulse width between two pulses in the table. We can design the pulse lookup table so that $\Delta w = \Delta t = (T/M \cdot n_T)$, where $M \geq 1$. Then, (18) becomes

$$I_w = \sum_{i=1}^L (h_{2,i} - h_{1,i})^2 \frac{1}{\Delta t}. \quad (19)$$

By substituting E_t , I_t , and I_w in (13) and (14), we may write

$$\sigma_j = \sqrt{\frac{P_j \cdot \alpha \cdot \sum_{i=1}^L h_{1,i}^2}{2\text{SNR} \cdot \sum_{i=1}^L (h_{1,i+1} - h_{1,i})^2}} \cdot \Delta t \quad (20)$$

$$\sigma_w = \sqrt{\frac{(1 - P_j) \cdot \alpha \cdot \sum_{i=1}^L h_{1,i}^2}{2\text{SNR} \cdot \sum_{i=1}^L (h_{2,i} - h_{1,i})^2}} \cdot \Delta t. \quad (21)$$

Since the electronics noise is bandlimited by $(1/\Delta t)$, (15) becomes

$$\sigma_e = \sqrt{\frac{(100 - \alpha) \cdot \sum_{i=1}^L h_{1,i}^2}{2\text{SNR}}} \cdot \Delta t. \quad (22)$$

Now we have established the relationship between SNR and the statistics of noise processes. Zero-mean Gaussian random variables can then be generated in units of Δt with above standard deviations following the method to be given in the next section.

III. PSEUDORANDOM NOISE GENERATOR

Transition noise, electronics noise, and write timing errors are randomly generated as nonuniform numbers. The embedded pseudorandom number generators must be uncorrelated and bear good randomness. In our design, uniform pseudorandom numbers are first produced and then transformed to nonuniform random numbers. One effective and popular uniform pseudorandom noise generator (PRNG) is the linear feedback shift register (LFSR) [8]. Its output is a maximum-length sequence (m -sequence) with a period of $2^n - 1$, where n is the number of storage units (registers). However, if many sequences of random noise are in demand, the area consumed by LFSR will be large, a situation that cannot be afforded in an FPGA implementation. For example, with five 6-bit wide PRNGs that can generate random numbers with a period longer than 1000 ($n = 10$), the number of registers will be 300. Facing area constraints in this design, we adopted an alternative method, *linear hybrid cellular automaton* (LHCA) (for example, see [9] and [10]). The principle of a cellular automaton is that the next value of each register is calculated by a Boolean function from the current values of immediate neighbors and itself. A cellular automaton is said to have a *null boundary* condition when the left- and right-most registers in the array connect to zeros. A null boundary is preferred to a feedback boundary because it avoids long feedback paths, and therefore reduces routing

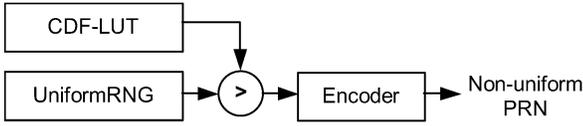
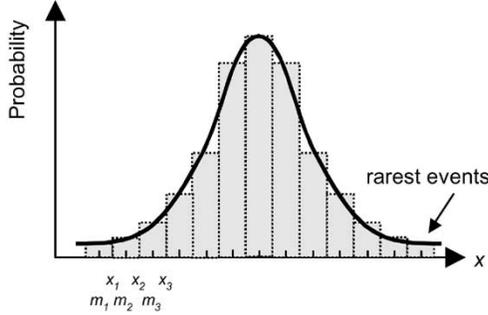


Fig. 2. Generation of nonuniform pseudorandom numbers.

Fig. 3. Splitting PDF into 2^m columns.

delay. It has been shown that when the Boolean functions are carefully selected and after the LHCA evolves for a number of initialization clock cycles, such a register array will output an m -sequence that has the same period as that of an LFSR [9]. The Boolean functions are called *computation rules* and categorized by Wolfram [13]. One of the setups that can generate m -sequences is a careful mix (hybrid) of two Boolean functions, Rule 90 and Rule 150, in the calculation of register contents [9]. Rule 90 and Rule 150 are defined as follows:

$$\text{Rule 90 : } d_i(t+1) = d_{i-1}(t) \oplus d_{i+1}(t) \quad (23)$$

and

$$\text{Rule 150 : } d_i(t+1) = d_{i-1}(t) \oplus d_i(t) \oplus d_{i+1}(t) \quad (24)$$

where $d_i(t)$ is the content of register i at time t . How to determine the positions of Rule 90 and Rule 150 in a register array can be found in [11] and [12].

One can easily see that the output from one register in the array is correlated to that of its immediate neighbor. In order to eliminate the correlation, only one bit of every $\eta + 1$ bits can be used to form a random number where η is called the *site spacing parameter* [9]. To compare with LFSR, if $\eta = 1$, the number of registers in a LHCA PRNG that can generate the same group of random numbers as mentioned above will be 60, a factor of 5 reduction in the number of registers.

Transformation from a uniform to a nonuniform pseudorandom number is illustrated in Fig. 2. An n -bit uniform random number is compared with numbers in a precalculated CDF conversion table and then encoded to an l -bit nonuniform random number. This exercise is equivalent to randomly picking points in the area under the corresponding PDF, grouping all points in a column, and substituting them with the one number m_i (Fig. 3). Attention shall be paid to the choice of n and l since they determine the accuracy and precision of the PRNG. The width n of a uniform random number limits the smallest probability we can simulate, which is $1/2^{n-1}$. Encoding from an n -bit to an l -bit random number effectively quantizes a continuous random variable X to an l -bit discrete

random variable X_q , when $n \gg l$. The width l controls the mean square loss associated with the quantization process.

Max [14] showed that the optimal quantizer is nonuniform when the PDF of the random variable is not uniform. However, the digital nature of the FPGA device determines the quantization levels $m_j (j = 1, 2, \dots, q)$ to be uniform where q is the number of levels, which is 2^l . We need to compute the boundaries $x_j (j = 1, 2, \dots, q - 1)$ for the suboptimal quantizer in term of minimizing the quantization loss. The quantization loss N_q is given in (25) and $f_X(x)$ is the PDF of X

$$N_q = E[(X - X_q)^2] = \sum_{i=1}^q \int_{x_{i-1}}^{x_i} (x - m_i)^2 f_X(x) dx. \quad (25)$$

In order to minimize the quantization loss, let

$$\frac{\partial N_q}{\partial x_i} = 0 \quad (i = 1, 2, \dots, q). \quad (26)$$

It can be easily found that

$$x_j = \frac{1}{2}(m_j + m_{j+1}). \quad (27)$$

Therefore, the suboptimal quantizer is a uniform quantizer. If we assume zero-mean Gaussian noise, we can get the close-form expression for N_q

$$N_q = \sum_{i=1}^q \int_{x_{i-1}}^{x_i} (x - m_i)^2 f_X(x) dx = \sum_{i=1}^q N_{q,i} \quad (28)$$

where

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{x^2}{2\sigma^2}} \quad (29)$$

and $Q(x)$ is the well-known tail-integral of the unit variance Gaussian PDF.

Since quantization noise can be seen as an additional perturbation to the random variable, we can define a signal to quantization noise ratio (SQNR) to evaluate the loss due to quantization as follows:

$$\begin{aligned} N_{q,i} &= \int_{x_{i-1}}^{x_i} (x - m_i)^2 f_X(x) dx \\ &= -\frac{\sigma}{\sqrt{2\pi}} \left(x_i e^{-\frac{x_i^2}{2\sigma^2}} - x_{i-1} e^{-\frac{x_{i-1}^2}{2\sigma^2}} \right) \\ &\quad + \sqrt{\frac{2}{\pi}} m_i \sigma \left(e^{-\frac{x_i^2}{2\sigma^2}} - e^{-\frac{x_{i-1}^2}{2\sigma^2}} \right) \\ &\quad + (m_i^2 + \sigma^2) \left[Q\left(\frac{x_{i-1}}{\sigma}\right) - Q\left(\frac{x_i}{\sigma}\right) \right] \end{aligned} \quad (30)$$

$$\text{SQNR} = \frac{\sigma_{\text{RV}}^2}{N_q}. \quad (31)$$

σ_{RV}^2 is the variance of the continuous random variable. For example, when $q = 64$ and $\sigma_{\text{RV}}^2 = 9$, SQNR is 20.4 dB. This means that as long as we use sufficiently many quantization levels the quantization loss can be suppressed down to a negligible level. The tradeoff is between the resolution and the resources consumed.

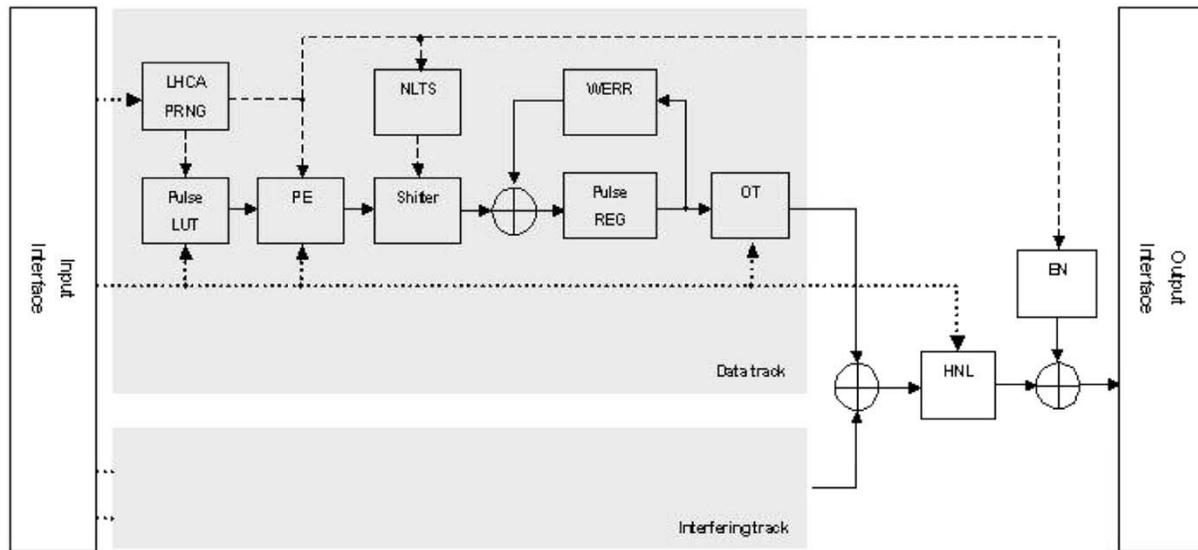


Fig. 4. Block diagram of the readback-signal generator.

IV. IMPLEMENTATION AND RECONFIGURATION

The design has been implemented on a Xilinx Virtex-E XCV-1000E device that is mounted on an Annapolis Micro Systems FIREBIRD PCI board to verify functionality and performance. A scaled-down version also has been synthesized and laid out targeting a Xilinx Virtex XCV-1000 device in order to demonstrate reconfigurability. The design is synthesized using Synplify's Synplify 6.2, simulated using ModelSim 5.5, and implemented using Xilinx Foundation 3.1. Since achievable resolution of the implementation is constrained by the size of Virtex-E XCV-1000E device, we modularized the design so it is easy to scale up when a device with more resources is available. Alternatively, when only some of the noise processes are required the capacity of the device can be effectively increased through reconfiguration, which will be briefly introduced in this section. The user can balance between satisfactory performance and consumed resource by modifying a set of design parameters. The architecture of the readback-signal generator is modularized and pipelined. Its structure is illustrated in Fig. 4. The data-track and interfering-track modules are almost identical except the random numbers from LHCA-PRNG are independent and the *off-track* (*OT*) ratios can be different. The output of data-track is multiplied by the ratio r while the interfering track is multiplied by a different ratio r' , ($r' \leq 1 - r$). ITI to the data track is realized through adding two weighted signals. The values of r and r' are determined by the off-track position of the read head [15]. In the diagram, the solid arrows mark the signal paths, the dashed ones indicate the random number paths, and the dotted lines are the control signal paths. The remaining seven modules in each track process the noise corruption and distortions on the pulses that are described in Section II. An L -bit-long isolated pulse is chosen from the *pulse lookup table* that is implemented in on-chip memory. The lookup table contains N pulses that have different pulse shapes and widths. The amplitude and the shape of the selected pulse are altered in the PE modules. In the NLTS module, nonlinear transition shift is combined with

the transition jitter. The actual pulse movement is executed by a two-level Barrel shifter. After passing through the shifter, the pulses are superimposed, as described in (4), and the write error is added to the pulse sequence in the WERR module. Pulse amplitude is then multiplied by the OT ratio to simulate out-track reading. Since each multiplier in OT has an operand that does not change in the process of signal generation (OT ratio does not change during runtime), these multipliers are implemented as lookup tables to improve the run speed. Finally, head nonlinearity is processed and electronics noise is added before the signal is sent to the output interface.

Three approaches of software-hardware mixed reconfiguration have been evaluated: 1) the embedded PRNG can be reinitialized in real time in order to simulate on-track variation; 2) FPGA bit streams that contain different combinations of noise processes can be reloaded into the device on-the-fly; and 3) dynamically reconnect modules that are needed or even build the modules in real time according to the used-specified parameters. Approach 1) lets us simulate different noise statistics, but it does not help reduce the resource consumption. The drawback of approach 2) is that it is not realistic to implement all the combinations into bitstreams. The third approach provides the user most flexibility to achieve a balance between performance and resource, and it can be realized through Xilinx's JBits package [16]. The dynamic reconfiguration issues are beyond the intended scope of this paper.

The readback-signal generator can operate at clock speeds up to 70 MHz on Virtex-E XCV-1000E. It takes about 4 h to generate 10^{12} pulses on the FPGA-based generator, compared to about 10 000 h in software. The runtime on a software-based generator is estimated on a PC equipped with one 1-GHz Pentium IV processor and 256 MB RAM. A sample output signal is shown in Fig. 5. All the design parameters in Table I can be easily scaled up when a more precise generator is in demand and a larger FPGA device is available. Every module can be easily removed from the design and a new module can be inserted without modification of others.

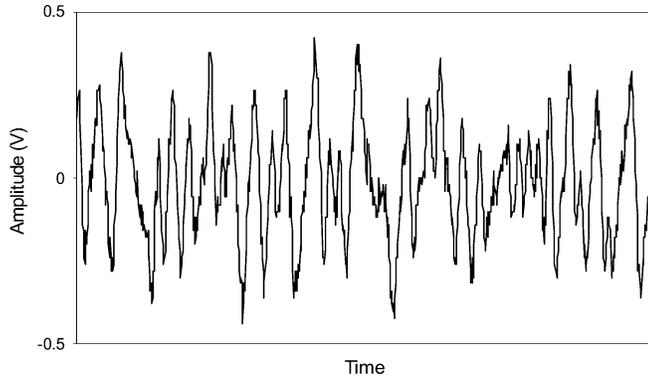


Fig. 5. Generated readback signal.

TABLE I
PARAMETERS OF SAMPLE IMPLEMENTATIONS

Parameter	Symbol	Value
Pulse look-up table size	N	16
ISI length	I	6
Symbol period	T	8
NLTS	ϵT	2
Signal-to-noise ratio	SNR	10 dB
Percentage of TN in total noise	α	90%
Fraction of jitter noise power in TN	P_j	0.75
LHCA length	L_{LHCA}	244
Site spacing parameter	η	1
Quantization levels	q	64
Uniform RN width	n	11
OT ratios	r and r'	0.9 and 0.1
Head nonlinearity	HNL	1 for pos. pulses; 0.9 for neg. pulses

V. ANALYSIS OF GENERATED SIGNALS

As we have discussed in Section II, many noise processes, such as transition jitter and NLTS, in magnetic recording systems have unique characteristics. In this section, we demonstrate that some of their known characteristics have been successfully captured in the signal generator.

Example 1: Supralinear Increase of Integrated Noise Power at High Recording Density: The relationship between integrated transition noise power and linear recording density, especially the supralinear behavior at high densities, has been systematically studied in [17]–[19]. It has been shown that the integrated transition noise power increases linearly at low densities since the data-dependent transition noise associated with each transition is unchanged. At higher densities, where bit length between consecutive transitions is shorter, there is more media noise in each individual transition, so the integrated transition noise power will no longer increase linearly with density. This phenomenon can be simulated as follows: 1) fix SNR, thus σ_j , when $PW50/T$ increases from 1.2 to 2, and linearly increase σ_j by 5% when $PW50/T$ increases from 2 to 2.6; 2) generate readback signals using an *all-one*'s data pattern; 3) calculate the power spectral density of generated signal and remove the data harmonics; 4) calculate the integrated noise power. During this signal generation process, we assume transition jitter noise is dominant ($\alpha = 99\%$ and $P_j = 99\%$).

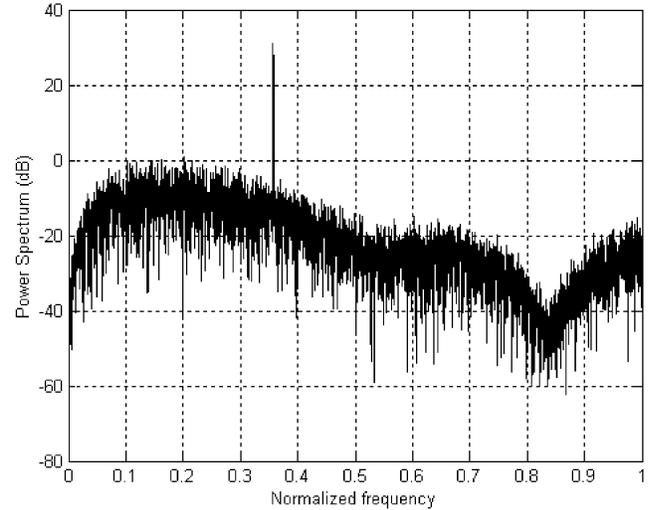


Fig. 6. Power spectrum of simulated readback signal.

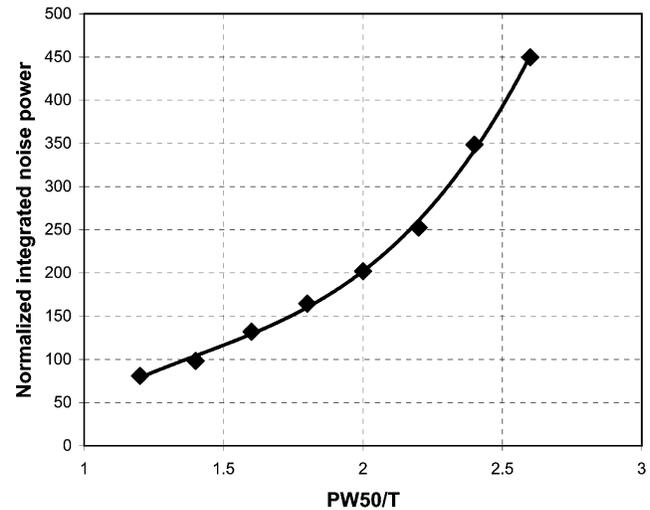


Fig. 7. Integrated transition noise power versus density. (The noise power is normalized to $V_{\text{isolated pulse}}^2 * T$.)

Fig. 6 shows the power spectrum of the simulated readback waveform at density $PW50/T = 1.4$. The characteristic low frequency hump due to transition noise is evident. We also plot the integrated noise power versus density in Fig. 7 in which a supralinear region after $PW50/T = 2$ can be clearly seen.

Example 2: Measurement of Nonlinear Transition Shift: Now we measure NLTS in the generated signal using the widely used 127-bit pseudorandom sequence method. Its theory and detailed procedures are described in [20]. The amount of NLTS can be calculated as the ratio of the amplitude of echoes (P1 and P2) to that of the main pulse (P0):

NLTS caused by the first adjacent transition:

$$NLTS_1 = \frac{2P1}{P0} \quad (32)$$

NLTS caused by the second adjacent transition:

$$NLTS_2 = \frac{2P2}{P0}. \quad (33)$$

TABLE II
NLTS DEFINITIONS

Adjacent transition	Second adjacent transition	NLTS
yes	no	37.5%
yes	yes	25%
no	yes	12.5%

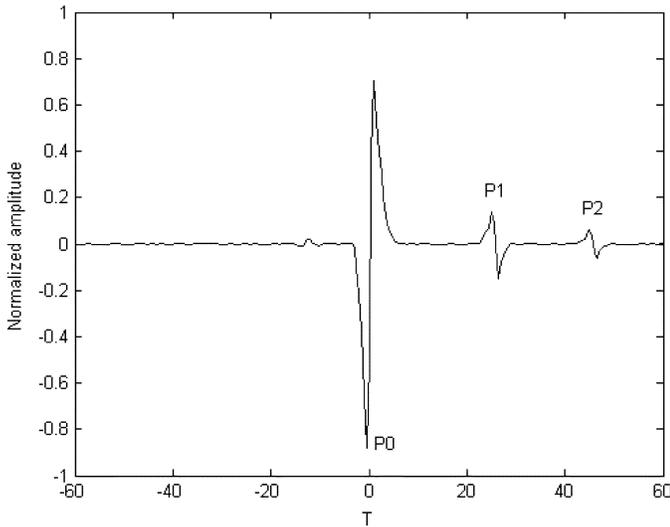


Fig. 8. Dipulse and echoes.

As an example, in the signal generator, we define the NLTS as in Table II.

Two echoes at 25.5 and 45.5 T can be easily seen in the dipulse extraction plot (Fig. 8). The measured NLTS is calculated as:

NLTS caused by the first adjacent transition:

$$\text{NLTS}_1 = \frac{2P1}{P0} = 36.4\%$$

NLTS caused by the second adjacent transition:

$$\text{NLTS}_2 = \frac{2P2}{P0} = 16.04\%.$$

VI. CONCLUSION

A readback-signal generator has been designed and implemented in Xilinx FPGA. It can generate input signals to a hardware-based read channel simulator. The noise processes and dis-

tortions have been modeled statistically. The design is modular so that the simulator can be easily reconfigured in runtime. The signal generator has been demonstrated running at a much faster speed than a software-based simulator. It also provides the user with flexibility in simulation while reducing the waste on resources through dynamic reconfiguration.

REFERENCES

- [1] J. Moon, "Discrete-time modeling of transition-noise-dominant channels and study of detection performance," *IEEE Trans. Magn.*, vol. 27, pp. 4573–4578, Nov. 1991.
- [2] J. Caroselli and J. K. Wolf, "Applications of a new simulation model for media noise limited magnetic recording channels," *IEEE Trans. Magn.*, pt. 1, vol. 32, pp. 3917–19, Sept. 1996.
- [3] C. D. Mee and E. D. Daniel, *Magnetic Recording Technology*, 2nd ed. New York: McGraw-Hill, 1996.
- [4] N. Curland and R. J. Macheliski, "Integrating an MR head into a peak detection channel," *IEEE Trans. Magn.*, vol. 30, pp. 309–315, Mar. 1994.
- [5] T. Yamauchi and J. M. Cioffi, "A nonlinear model for thin film disk recording systems," *IEEE Trans. Magn.*, vol. 29, pp. 3993–3995, Nov. 1993.
- [6] B. Lin, E. Hild, and J. R. Cruz, "Magnetoresistive read/write channel models," *IEEE Trans. Magn.*, vol. 35, pp. 4528–4531, Nov. 1999.
- [7] J. Moon, "SNR definition for magnetic recording channels with transition noise," *IEEE Trans. Magn.*, vol. 36, pp. 3881–3883, Sept. 2000.
- [8] S. W. Solomon, *Shift Register Sequences*. Laguna Hills, CA: Aegean Park, 1982.
- [9] P. D. Hortensius, R. D. McLeod, W. Pries, D. M. Miller, and H. C. Card, "Cellular automata-based pseudorandom number generators for built-in self-test," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 842–859, Aug. 1989.
- [10] P. D. Hortensius, R. D. McLeod, and H. C. Card, "Parallel random number generation for VLSI systems using cellular automata," *IEEE Trans. Comput.*, vol. 38, pp. 1466–1473, Oct. 1989.
- [11] S. Zhang, D. M. Miller, and J. C. Muzio, "Determination of minimal cost one-dimensional linear hybrid cellular automata," *Electron. Lett.*, vol. 27, pp. 1625–1627, Aug. 1991.
- [12] K. Cattell and S. Zhang, "Minimal cost one-dimensional linear hybrid cellular automata of degree through 500," *J. Electron. Test.: Theory Applicat.*, vol. 6, pp. 255–258, Apr. 1995.
- [13] S. Wolfram, "Statistical mechanics of cellular automata," *Rev. Mod. Phys.*, vol. 55, pp. 601–644, July 1983.
- [14] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Inform. Theory*, vol. IT-6, pp. 7–12, Mar. 1960.
- [15] J. Chen and J. Moon, "Detection signal-to-noise ratio versus bit cell aspect ratio at high areal densities," *IEEE Trans. Magn.*, vol. 37, pp. 1157–1167, May 2001.
- [16] S. A. Guccione, D. Levi, and P. Sundararajan, "JBits: A Java-based interface for reconfigurable computing," in *Proc. 2nd Annu. Military and Aerospace Applications of Programmable Devices and Technologies Conf.*, 1998.
- [17] R. A. Baugh, E. S. Murdock, and B. R. Natarajan, "Measurement of noise in magnetic media," *IEEE Trans. Magn.*, vol. MAG-19, pp. 1722–1724, Sept. 1983.
- [18] N. Berk, P. George, and G. Mowry, "Noise in high performance thin-film longitudinal magnetic recording media," *IEEE Trans. Magn.*, vol. MAG-21, pp. 1350–1355, May 1985.
- [19] M. Madrid and R. Wood, "Transition noise in thin-film media," *IEEE Trans. Magn.*, vol. MAG-22, pp. 892–894, Sept. 1986.
- [20] D. Palmer, P. Ziperovich, R. Wood, and T. Howell, "Identification of nonlinear write effects using pseudorandom sequences," *IEEE Trans. Magn.*, vol. MAG-23, pp. 2377–2379, Sept. 1987.