FPGA-Based Bit Error Rate Performance Measurement of Wireless Systems

Amirhossein Alimohammad and Saeed Fouladi Fard

Abstract—This paper presents the bit error rate (BER) performance validation of digital baseband communication systems on a field-programmable gate array (FPGA). The proposed BER tester (BERT) integrates fundamental baseband signal processing modules of a typical wireless communication system along with a realistic fading channel simulator and an accurate Gaussian noise generator onto a single FPGA to provide an accelerated and repeatable test environment in a laboratory setting. Using a developed graphical user interface, the error rate performance of single- and multiple-antenna systems over a wide range of parameters can be rapidly evaluated. The FPGA-based BERT should reduce the need for time-consuming softwarebased simulations, hence increasing the productivity. This FPGA-based solution is significantly more cost effective than conventional performance measurements made using expensive commercially available test equipment and channel simulators.

Index Terms— Baseband performance validation, bit-error rate tester (BERT), fading channel simulation, field-programmable gate array (FPGA), Gaussian noise generator (GNG), Golay code, maximum likelihood (ML).

I. MOTIVATION

T HE pace of wireless system development using the latest communication techniques is increasingly limited by the design productivity. It is critical to verify the design characteristics at the earliest possible stage of design (e.g., at the baseband level) to minimize costly design iterations. At the physical (PHY) layer, the bit error rate (BER) performance metric is widely used to measure the reliability of the communication systems. Because BER properties are not in general amenable to analysis, Monte Carlo (MC) simulation techniques have been widely used to generate BER versus a range of expected signal-to-noise ratio (SNR) conditions. However, the execution times of software-based MC simulations of the baseband layer on workstations can be extremely long, especially for increasingly complex communication systems. This is mainly because:

 Many modern techniques, such as multiple-inputmultiple-output (MIMO) systems, rely on computationally intensive signal processing at the receiver.

Manuscript received October 14, 2013; revised May 19, 2013; accepted July 25, 2013.

A. Alimohammad is with the Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA 92142 USA (e-mail: aalimohammad@mail.sdsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2013.2276010

Therefore, bit-true software-based simulation of these algorithms on workstations is becoming prohibitively time consuming. In addition, for a communication system specification with a set of target system requirements such as data throughput, received power, available bandwidth, noise statistics, and a target error performance, there are typically various potential solutions. Each solution can use a different combination of subsystem designs with different sets of input parameters. Exploring the design space to achieve an optimized overall system solution that meets the target specifications can involve a large number of options.

2) To estimate the BER performance of a communication system with the MC simulation method, we have to measure the BER over a large number of independent problem instances [1]. While simulation of digital communication systems under additive white Gaussian noise (AWGN) channels is straightforward as the system performance is averaged over a large number of independent instances of noise and data, BER performance measurement of wireless systems over time-varying fading channels requires significantly longer simulation times because of the dependence between the channel instances. To accurately estimate the BER performance of a communication system over a time-varying fading channel, the error performance needs to be averaged not only on independent instances of noise and data, but also on the fading channel samples over a long period [1], [2]. Such a performance evaluation can require several weeks or months of software simulations.

Hardware simulators can accelerate the performance evaluation of communication systems compared with software simulators by several orders of magnitude [3]-[6]. This makes hardware-accelerated prototyping and validation of the PHY layer as an increasingly attractive alternative. Published hardware-based baseband BER measurement systems [3], [7], [8] use field-programmable gate arrays (FPGAs) and use model-based systems such as Simulink [9] to integrate parameterizable IP blocks (such as conventional cores for forward error correction) onto an FPGA. While using systemlevel tools can eliminate the need for extensive hardware knowledge and will usually shorten the design time, a simulation library may include only a set of basic digital communication components and might not include modules, such as new coding algorithms, for emerging technologies. Thus, designers will still need to implement various communication

S. F. Fard is with PMC-Sierra, Calgary, AB K2E 7Z2, Canada.

2

IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS



Fig. 1. Block diagram of the implemented BERT system.

modules with compatible interfaces with other components. In addition, most of the published BER testers (BERTs) verify the performance under the linear AWGN channel [3], [7], [8], which is a rather inadequate model for wireless mobile communication systems. Fading channel models for mobile communication systems must reproduce the statistical properties of radio propagation environments [10]. Furthermore, although several accurate Gaussian noise generators (GNGs) have been reported over the last few years [11]-[13], published BERTs use a noise generator that has a relatively poor accuracy. For example, the accuracy of the GNG supplied by Xilinx [14] degrades at the tails of the probability density function (pdf) for $|n| \ge 4.8\sigma_n$ and the pdf accuracy of the AWGN in [3] is limited to the interval $[0.2\sigma_n\%, 4.8\sigma_n\%]$, where σ_n^2 is the noise power. For a reliable BER measurement, it is necessary for the GNG to accurately generate samples at the tail of the Gaussian pdf.

This paper presents the design and implementation of a parameterizable baseband MIMO BER measurement system on an FPGA. The proposed BERT integrates various signal processing modules of a typical PHY layer, such as the channel encoder, interleaver, modulator, demodulator, deinterleaver, channel decoder, and symbol detector, along with our previously published realistic fading channel simulator [15]–[18] and accurate GNG [13] onto a single FPGA to provide an accelerated and repeatable test environment in the laboratory.

The rest of this paper is organized as follows. Section II briefly presents the overall structure of the BERT, the digital source module, and the process of encoding and decoding data bits. Section III presents the architecture of the interleaver and deinterleaver. Sections IV and V briefly discuss the fading channel simulator and the AWGN generator, respectively. Section VI discusses the process of generating received signals. Section VII presents the detector architecture. Section VIII presents the BERT implementation and simulation results. Finally, Section IX makes some concluding remarks.

II. BERT STRUCTURE, DIGITAL SOURCE, AND ENCODING/DECODING PROCESS

Fig. 1 shows the block diagram of the BER performance measurement system. In the implemented BERT, source bits are encoded using an extended Golay channel code [19] and



Fig. 2. Logic diagram of the CTG258 generator.

interleaved with a length 16 383 pseudorandom interleaver [20]. Then, the interleaved bits are modulated to 4-quadraticamplitude modulation (4-OAM) symbols and passed through the MIMO channel, where they are affected by spatiotemporally correlated MIMO fading variates, and corrupted with AWGN. In the receiver, a maximum likelihood (ML) detector tries to estimate the transmitted bits. In addition, perfect channel state information is assumed to be available to the receiver. After ML detection, the bit stream is deinterleaved, decoded, and compared with the transmitted bit stream. To demonstrate the fading channel effects on the transmitted symbols, we also implemented a single-antenna transmitter where the bits can be modulated using different schemes (BPSK, QPSK, 4-pulse-amplitude modulation, 4-QAM, 8-phase-shift keying (PSK), 16-PSK, 16-QAM, circular 8-QAM, and circular 16-QAM [20]). As shown in Fig. 1, the output of the singleinput-single-output (SISO) channel can be passed to an oscilloscope through a digital-to-analog converter. In addition, the corrupted fading samples with AWGN can be monitored on the oscilloscope screen as well.

For the digital information source, it is common to use a pseudorandom number generator (PRNG). Linear feedback shift registers (LFSRs) are undoubtedly the best known register-based PRNGs [21]. Because of their compactness, fast bit-level operations, and the exponential increase of their period with the width of the shift register, LFSRs have enjoyed success in many hardware-based simulations and digital circuit testing. However, LFSRs produce only a single bit at each clock cycle, which is rather slow for very longrunning simulations. On the other hand, the sequences using LFSRs with parallel outputs have undesirable correlations [22]. In this system, we used combined Tausworthe generators (CTGs), which has improved statistical properties [23]. The logic diagram of a 64-bit CTG with five components and a period of $\rho \approx 2^{258}$ (CTG258) is shown in Fig. 2, where \ll s denotes a left shift of s bits, $\gg 1$ denotes a right shift of 1 bits, C_i , j = 1, ..., 5, are the five constant values, and S_i are the five 64-bit state variables initialized with five separate seeds. It is recommended that the initial seeds S_i be large different values [24].

The generated data source is encoded using the Encoder module. Channel codes or error control codes are used in ALIMOHAMMAD AND FARD: FPGA-BASED BER PERFORMANCE MEASUREMENT

Algorithm 1 IMLD decoding for the extended Golay code

Compute the syndrome x = wP.
 If (weight(x) ≤ 3) then set e = [x, 0], and go to 8.
 If (weight(x + b_i) ≤ 2) for some row b_i of B then set e = [x + b_i, o_i], and go to 8.
 Compute the second syndrome xB.
 If (weight(xB) ≤ 3) then set e = [0, xB], and go to 8.
 If (weight(xB + b_i) ≤ 2) for some row b_i of B then set e = [o_i, xB + b_i], and go to 8.

- 7: The error pattern cannot be determined. Exit.
- 8: The decoded vector is $\hat{\mathbf{v}} = \mathbf{w} + \mathbf{e}$. Exit.

communication systems to detect and possibly correct the errors that occur during data transmission. This is accomplished by adding redundant data to the transmitted message. In our prototype BERT system, we used the extended binary Golay code, which is a linear binary block code [19]. This code can be generated by the 12×24 generator matrix $\mathbf{G} = [\mathbf{I}, \mathbf{B}]$, where \mathbf{I} is the 12×12 identity matrix and \mathbf{B} is given by

The code rate *R* for the (24, 12) extended Golay code is 1/2. This code has minimum distance $d_{\min} = 8$ and can correct up to three errors.

Encoding the data bits using the Golay code is straightforward. Assuming $\mathbf{u} = [u_{11}, u_{10}, \dots, u_0]$ to be the vector of 12 source bits, the coded bits can be calculated as $\mathbf{v} = [\mathbf{u}, \mathbf{p}] =$ uG in the Galois field of two elements GF(2) [19], where **p** is the length 12 row vector of parity bits. To decode the extended binary Golay code, we used the imperfect maximum likelihood decoding (IMLD) algorithm [25], which attempts to find all of the error patterns e of weight at most three. The error pattern \mathbf{e} is denoted as $\mathbf{e} = [\mathbf{e}_1, \mathbf{e}_0]$, where \mathbf{e}_0 and \mathbf{e}_1 are the lower and upper parts of e, each with 12 bits. Assume that $\mathbf{w} = [w_{23}, w_{22}, \dots, w_0]$ represents the received vector, \mathbf{b}_i denotes the *i*th row of **B**, and let \mathbf{o}_i be a row vector of length 12 with a one in the *i*th position and zeros elsewhere. The IMLD algorithm tries to find the error pattern of the received vector by computing the syndrome $\mathbf{x} = \mathbf{w}\mathbf{P}$, where $\mathbf{P} = \mathbf{G}^T$ is the parity check matrix and $(\cdot)^T$ denotes the transpose of a matrix. Algorithm 1 describes the IMLD decoding for the extended Golay code, which can correct all of the error patterns with one, two, and three errors. More error patterns can also be detected and reported for requesting retransmission. The pipelined datapath of the imperfect ML decoder based on Algorithm 1 is implemented at the behavioral level using the Verilog hardware description language.

(a) Read After Write (RAW) mode



(b) Write After Read (WAR) mode



Fig. 3. Datapath of the implemented (a) interleaver and (b) deinterleaver.

III. INTERLEAVER AND DE-INTERLEAVER

A burst of errors, which typically arises in the channel when the signal experiences deep fades, can be overwhelming for the error control code that can only correct a certain number of errors in a block of data samples. This problem can be alleviated by randomizing the distribution of errors in a block of data using interleavers [20]. Interleaving spreads the transmitted data over time, results in significant improvements in finding and correcting errors at the error correction decoder.

For our MIMO communication system, we implemented a pseudorandom interleaver of length 16 383. A pseudorandom interleaver is a variation of a block interleaver where coded bits are written linearly into a memory and read out randomly based on a pseudorandom sequence. Fig. 3 shows the datapath of the implemented pseudorandom interleaver and deinterleaver. In the interleaver, a 14-bit counter is used to write the coded input bits into a 16 384×1 memory. This counter counts linearly from 1 to 16 383 and goes back to 1. At the output, a 14-bit LFSR is used to read out the coded bits randomly from the memory to decrease the correlation between the encoded samples. Notice that the counter does not generate 0 as zero is not among the values that are generated by an LFSR, hence the interleaver length is $16\ 384 - 1$. In the deinterleaver, the reverse operation is performed, where the received bits are written randomly into a memory using the same pseudorandom sequence and later read out using a circular counter that counts from 1 to 16 383.

When a new bit *bin* is passed to the interleaver (shown by the *newBit* signal connected to the write enable WE port), it is written into the memory location addressed by the counter CNTR and the counter is incremented for the next cycle. Then, an output bit *Do* is read out from the location determined by the current state of the LFSR. The LFSR is then updated for 4

IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS



Fig. 4. 2×2 MIMO channel.

the next cycle. Note that the input bits must be written into the memory before reading the output bits to maintain the integrity of the data sequence. The inverse operation happens in the deinterleaver. When a new bit is ready to be written into the memory, the deinterleaver first reads one bit from CNTR location of the memory and informs the next stage using the *bitReady* signal. Here, the data is read out of the memory and stored in the *bOut* register before writing the input bit. Then, the deinterleaver writes the input bit into the memory and updates the LFSR for the next cycle. The counter CNTR is also updated for the next cycle.

IV. SPATIOTEMPORALLY CORRELATED FADING CHANNEL SIMULATOR

The radio propagation environment introduces significant impairments and sources of interference into wireless communications. To provide a realistic fading channel model and repeatable experimental results in a laboratory setting, a baseband fading channel simulator must accurately reproduce the statistical properties of time-varying radio propagation environments. Fig. 4 shows a 2×2 MIMO channel, where the element $h_{jk}(t)$, $j = 1, \ldots, n_R$, $k = 1, \ldots, n_T$, represents the complex-valued fading gain between the kth transmit antenna and the *j*th receive antenna at any time t, and n_T and n_R are the number of transmit and receive antennas, respectively. A fading channel gain is commonly modeled as a complex Gaussian wide-sense stationary uncorrelated scattering process $h_{jk}(t) = h_{jk,i}(t) + jh_{jk,q}(t)$, where $h_{jk,i}$ and $h_{jk,q}$ denote the in-phase and quadrature components of the fading channel gain, respectively, and the envelope $|h_{ik}(t)|$ follows the Rayleigh distribution [26]. The amplitude statistics for each complex-valued fading coefficient $h_{ik}(t)$ may follow a Rician distribution if a line-of-sight path is present [27]. The Nakagami-m distribution has also been used to model fading that is more or less severe than Rayleigh fading [28].

To generate temporally correlated complex-valued Gaussian fading gains $\{h_{jk}(t)\}\)$, we used the sum-of-sinusoids (SOS) fading channel model, where the fading channel gain is formed by superimposing N sinusoidal waveforms with amplitudes, frequencies, and phases that are selected appropriately to generate the desired statistical properties of radio propagation environments accurately [29]. Using the novel SOS-based fading channel model in [15]–[18], each complex discrete-time Rayleigh fading process $h_{jk}[m] = h_{jk,i}[m] + jh_{jk,q}[m]$ is described as

$$h_{jk,i}[m] = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} \cos\left(2\pi f_D T_s \cos(\alpha_n[m])m + \varphi_n[m]\right)$$

$$h_{jk,q}[m] = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} \cos\left(2\pi f_D T_s \sin(\alpha_n[m])m + \psi_n[m]\right)$$

where $m = 0, 1, 2, \cdots$ is the discrete time index, f_D is the maximum Doppler frequency for any path, T_s is the symbol period, $\alpha_n[m] = (2\pi n - \pi + \theta[m])/4N$ is the angle of arrival of the *n*th sinusoid, and $\varphi_n[m]$ and $\psi_n[m]$ are the phases of the in-phase and quadrature components, respectively, of the *n*th sinusoid. Note that in this model, the $\psi_n[m]$, $\varphi_n[m]$, and $\theta[m]$ are mutually independent and uniformly distributed random walk processes (RWPs) over $[-\pi, \pi)$ for all *n*, instead of uniformly distributed random variables, which are widely used in most of the SOS-based fading channel models. It was shown in [15] and [16] that the time-averaged statistical properties of the generated fading samples do indeed match the reference theoretical properties when θ , φ_n , and ψ_n are RWPs.

In a typical wireless communication scenario, the Doppler frequency f_D is significantly smaller than the signal sample rate $F_s = 1/T_s$. This allows us to design much of the fading channel simulator at a much lower sample rate and, thereby reduce the required hardware resources. Fading samples generated at the slower sample rate are then interpolated to provide samples at the desired output sample rate. Therefore, instead of generating fading samples directly, the discrete differences between subsequent fading samples (i.e., d_{ik} = $h_{ik}[m+1] - h_{ik}[m]$) are generated at a low sample rate using a time-multiplexed datapath. Fading channel simulator is a relatively complex module. The detailed description of the fading channel model and the statistical analysis of generated fading variates can be found in [15] and [16]. The compact hardware architecture of the fading channel simulator were presented in [17] and [18].

While the generated fading samples $\{h_{ik}(t)\}\$ are correlated in time, the fading channel simulator is extended to model the spatial correlations that exist between the antennas in MIMO systems. To obtain the space-time (ST) correlation characteristics, a temporally correlated random process, for example, generated using the fading variate generator in [18], can be followed by linear transformations to be made spatially correlated [30]. The spatial structure of the channel is commonly characterized by channel correlation matrices. Our BERT supports the four analytical narrowband models of the fading coefficients, the independent identically distributed (i.i.d.) flat-fading model [2], the Kronecker model [31], the Weichselberger model [32], and the virtual channel representation (VCR) model [33], which are widely accepted by the research community for performance analysis [34]. For example, the Kronecker model can be expressed as

$$\mathbf{H} = \mathbf{U}\mathbf{G}\mathbf{V} \tag{1}$$

where **H** is the $n_R \times n_T$ spatiotemporally correlated fading channel matrix. **G** is the $n_R \times n_T$ i.i.d. matrix with zeromean unit variance circularly -symmetric complex Gaussian distributed entries, $\mathbf{U} = \mathbf{R}_{RX}^{1/2}$, $\mathbf{V} = (\mathbf{R}_{TX}^{1/2})^T$, \mathbf{R}_{Tx} , and \mathbf{R}_{Rx} denote the $n_T \times n_T$ transmit and $n_R \times n_R$ receive correlation matrices, respectively. The Weichselberger model and the VCR model can be written as

$$\mathbf{H} = \mathbf{U}(\mathbf{W} \odot \mathbf{G})\mathbf{V} \tag{2}$$

ALIMOHAMMAD AND FARD: FPGA-BASED BER PERFORMANCE MEASUREMENT



Fig. 5. Architecture of the spatiotemporally correlated fading variate generator.

where **G** is multiplied by the $n_R \times n_T$ matrix **W** using the element-wise Schur–Hadamard multiplication operation \odot .

To introduce spatial correlations between the temporally correlated fading samples, instead of performing the matrix calculations on the high-frequency samples, similarly to the generation of temporally correlated fading samples, the matrix operations are performed on the low-frequency fading samples and later up sample the resulting streams with appropriate interpolators. For an efficient implementation of the spatial correlation characteristics of analytical MIMO fading channels, we designed a pipelined architecture, as shown in Fig. 5. The datapath receives the difference fading samples from the fading variate generator described in [18], performs the matrix operations of either (1) or (2) on the generated difference fading channel matrix $\mathbf{G} = \mathbf{D}$ and passes the spatiotemporally correlated fading samples to the next stage for interpolation.

The three RAMs uRAM, wRAM, and vRAM are used to keep the elements of the U, W, and V matrices, respectively. In addition, the dual-port memory tRAM is used as a register bank for holding the intermediate results. The *fading variate generator* module generates the discrete difference between two subsequent low-frequency fading samples (i.e., the current sample and the previous sample) for each transmit-receive antenna pair using the fading channel simulator presented in [18]. The core of this architecture is the arithmetic unit (AU) datapath, which performs basic complex arithmetic operations, such as complex products. A brief explanation of the hardware architecture for the AU can be found in [35]. The details of the interpolator architecture are presented in [15] and [18].

V. AWGN GENERATOR

To generate Gaussian noise samples, we use the Box–Muller (BM) algorithm [36]. The inputs to the BM algorithm are two independent uniformly distributed pseudorandom numbers (PRNs), $u_1, u_2 \in (0, 1)$. The outputs n_1 and n_2 are the two independent samples from a zero-mean unit-variance Gaussian



Fig. 6. (a) Plot of $f(u_1) = \sqrt{-2\ln(u_1)}$. (b) Nonlinear behavior of $f(u_1)$ in the vicinity of $u_1 = 1$.

distribution that can be obtained as

$$\begin{cases} n_1 = \sqrt{-2\ln(u_1)} \times \sin(2\pi u_2) \\ n_2 = \sqrt{-2\ln(u_1)} \times \cos(2\pi u_2). \end{cases}$$
(3)

Approximating the sine and cosine functions is straightforward. The quarter cycle of the sine (cosine) function is partitioned into 1024 uniform segments. The precomputed values are stored in two memory blocks. Using the symmetry of the trigonometric functions, the sine and cosine values over $(0, 2\pi)$ can then be approximated relatively accurately [13], [27]. To approximate $f(u_1) = \sqrt{-2\ln(u_1)}$ between $u_1 \in (0, 1)$, we use segmentation and polynomial curve fitting techniques. We note that the function $f(u_1)$ has two high-slope regions: in the vicinity of $u_1 = 0$ and close to $u_1 = 1$, as shown in Fig. 6. This means that a small input change may lead to a (very) large output change. Thus, the input domains near zero and one need smaller segments than the relatively linear regions in the middle of the domain. Therefore, we used a hybrid segmentation scheme in which both logarithmic and uniform segmentations are used [13]. First, the domain (0, 1) of u_1 is divided into two subintervals, $r_0 \in (0, 0.5)$ and $r_1 \in [0.5, 1)$. Subintervals r_0 and r_1 are then segmented logarithmically from $u_1 = 0.5$ down to zero and up to one, respectively, into $31 \times 2 = 62$ segments. Then, each logarithmic segment is subdivided uniformly into eight subsegments. The value of $f(u_1)$ within each of the 62×8 segments can then be approximated more efficiently using separate linear polynomials $f(u_1) = a \times u_1 + b$. The coefficients a and b for each segment were calculated using the orthogonal least squares fit method [37] to minimize the residual error. The number of segments depends on the desired accuracy and on the size of memory that is available to store the coefficients of polynomials.

An important point to note is that the coefficients of $f(u_1)$ have extremely large values when u_1 is in the vicinity of zero or one. Storing the large coefficient values of $f(u_1)$ onchip requires relatively large memories, increases the hardware complexity and likely slows down the variate generation rate. To overcome these problems, the suitably scaled coefficients \tilde{a} and \tilde{b} for all segments are stored in the Coefficient Memory, as shown in Fig. 7(a). For a given PRN input u_1 , the Addressing Unit calculates the scaled value of u_1 and generates a signed value of \tilde{u}_1 and also produces the segment address. Thus, the scaled coefficients of the linear piece can be addressed and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS



Fig. 7. (a) Datapath for generating properly scaled Gaussian noise samples. (b) Gaussian pdf compared with the pdf of 10^{11} generated Gaussian samples with $\sigma_n = 1$.

read directly from the Coefficient Memory to approximate $f(\tilde{u}_1) = (\tilde{a} \times \tilde{u}_1) + \tilde{b}$ [13], [27].

To evaluate the error rate over various SNR values, we normalize the average energy of constellation symbols to unity (i.e., $E_s = 1$) and only the power of the generated White Gaussian noise samples (with zero mean and variance $\sigma_n^2 = 1$) need to be scaled for the desired noise variance $\sigma_n = \sqrt{(2N_o/E_b \log_2 M)}$, where *M* is the size of the signal constellation and E_b is the energy per bit. One can see that the SNR is related to E_b/N_o as

$$\frac{E_b}{N_o} = \frac{E_s}{\sigma_n^2} \cdot \frac{2}{\log_2 M} = \frac{2 \operatorname{SNR}}{\log_2 M}$$

where $E_s = \log_2 M E_b$ is the energy per symbol and the noise power (variance) is $\sigma_n^2 = N_o/2$. Therefore, the White Gaussian noise samples are multiplied by σ_n to support Gaussian variates with variable noise variances. Fig. 7(b) superimposes the pdf of 10^{11} generated Gaussian samples on top of a pdf plot of the ideal normal distribution. The two plots are indistinguishable over $\pm 6.6\sigma_n$. Various statistical characteristics of the generated Gaussian samples are also evaluated and confirmed using multiple standard statistical goodness-of-fit tests in [13].

VI. GENERATION OF RECEIVED SAMPLES

In the implemented 2×2 MIMO system, 4-QAM modulated transmitted symbols **s** are passed through the fading channel **H**. The complex-valued received samples $\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{n}$ can be expressed as

$$\begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$$
(4)

where h_{jk} , $j, k \in \{1, 2\}$ is the complex-valued channel gain between *k*th transmit and *j*th receive antenna, s_j is the transmitted 4-QAM symbol from the *j*th transmit antenna, and $\{n_j\}$ are the AWGN samples. Decomposing the complex received samples into their in-phase and quadrature components, we can rewrite (4) as

$$\begin{pmatrix} r_{1,i} \\ r_{2,i} \\ r_{1,q} \\ r_{2,,q} \end{pmatrix} = \begin{pmatrix} h_{11,i} & h_{12,i} & -h_{11,q} & -h_{12,q} \\ h_{21,i} & h_{22,i} & -h_{21,q} & -h_{22,q} \\ h_{11,q} & h_{12,q} & h_{11,i} & h_{12,i} \\ h_{21,q} & h_{22,q} & h_{21,i} & h_{22,i} \end{pmatrix} \begin{pmatrix} s_{1,i} \\ s_{2,i} \\ s_{1,q} \\ s_{2,q} \end{pmatrix} + \begin{pmatrix} n_{1,i} \\ n_{2,i} \\ n_{1,q} \\ n_{2,q} \end{pmatrix}.$$

$$(5)$$



Fig. 8. Datapath and timing diagram of the received symbol generation. Note that add1 = add2 and add3 = add4.

The quadrature components of the fading gains $h_{jk,i}$ and $h_{jk,q}$, the input samples $s_{j,i}$ and $s_{j,q}$, and the noise samples $n_{j,i}$ and $n_{j,q}$ are assumed to be available and constant during each cycle. The fading gains and the AWGN samples are generated using the fading variate generator and the GNG presented in Sections IV and V, respectively.

Fig. 8 shows the datapath and the timing diagram of the received symbol generation. Because the in-phase and the quadrature components of the 4-QAM symbols comprise only +1 and -1 values, the **Hs** expression can be implemented without using multipliers. As Fig. 8 shows, this block has been implemented with four accumulators with add/subtract inputs. The accumulators acc1, acc2, acc3, and acc4 are reset at the beginning of each cycle. The accumulator adds (subtracts) the input to (from) its current value if the *add* input is 1(0). For the quadrature components of the input, the digital value one is assumed to represent arithmetic +1 and the digital value zero represents arithmetic value -1. To calculate $r_{1,i}$, the accumulator acc1 adds/subtracts the fading gains $h_{11,i}$.



Fig. 9. Datapath of the implemented ML detector for the 2×2 MIMO system with 4-QAM modulated symbols.

 $h_{12,i}$, $h_{11,q}$, $h_{12,q}$ according to $s_{1,i}$, $s_{2,i}$, $not(s_{1,q})$, $not(s_{2,q})$, respectively, and adds $n_{1,i}$. As shown in Fig. 8, $r_{1,q}$, $r_{2,i}$, and $r_{2,q}$ are calculated similarly.

VII. ML DETECTOR

The task of the MIMO detector is to estimate the symbol vector \mathbf{s} from the received signal vector \mathbf{r} . At the receiver, assuming that the channel matrix \mathbf{H} is known (or estimated perfectly), an ML detector computes an estimate $\hat{\mathbf{s}}$ for each transmitted ST symbol

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s} \in \mathbb{Q}^{n_T}} \left\{ \|\mathbf{r} - \mathbf{H}\,\mathbf{s}\| \right\}$$
(6)

where \mathbb{Q} denotes the signal constellation and the minimum is sought over all possible n_T -element ST symbols $\mathbf{s} \in \mathbb{Q}^{n_T}$ [38]. ML detection provides an optimal error rate performance. Assuming that the transmitted symbols are modulated with 4-QAM scheme, the ML detector for a 2 × 2 MIMO channel can be expressed as

$$\left\| \begin{pmatrix} r_{1,i} \\ r_{2,i} \\ r_{1,q} \\ r_{2,q} \end{pmatrix} - \begin{pmatrix} h_{11,i} & h_{12,i} & -h_{11,q} & -h_{12,q} \\ h_{21,i} & h_{22,i} & -h_{21,q} & -h_{22,q} \\ h_{11,q} & h_{12,q} & h_{11,i} & h_{12,i} \\ h_{21,q} & h_{22,q} & h_{21,i} & h_{22,i} \end{pmatrix} \begin{pmatrix} s_{1,i} \\ s_{2,i} \\ s_{1,q} \\ s_{2,q} \end{pmatrix} \right\|^{2}$$

$$= \left(r_{1,i} - s_{1,i}h_{11,i} - s_{2,i}h_{12,i} + s_{1,q}h_{11,q} + s_{1,q}h_{12,q} \right)^{2}$$

$$+ \left(r_{2,i} - s_{1,i}h_{21,i} - s_{2,i}h_{22,i} + s_{1,q}h_{21,q} + s_{1,q}h_{22,q} \right)^{2}$$

$$+ \left(r_{1,q} - s_{1,i}h_{11,q} - s_{2,i}h_{12,q} - s_{1,q}h_{11,i} - s_{1,q}h_{12,i} \right)^{2}$$

$$+ \left(r_{2,q} - s_{1,i}h_{21,q} - s_{2,i}h_{22,q} - s_{1,q}h_{21,i} - s_{1,q}h_{22,i} \right)^{2} .$$

$$(7)$$

Fig. 9 shows the datapath of the implemented ML detector. In this figure, the Section cost function calculates $c(\mathbf{s}) = \|\mathbf{r} - \mathbf{Hs}\|$ according to (7). The quadrature components of the tentative samples, (i.e., $s_{1,i}$, $s_{1,q}$, $s_{2,i}$, and $s_{2,q}$) are modulated with the fading gains and subtracted from the received input signal. For example, the first branch of the cost function Section (including the adder/subtracters U0, U4, U8, U12 and the multiplier U16) calculates $(r_{1,i} - s_{1,i}h_{11,i} - s_{2,i}h_{12,i} + s_{1,q}h_{11,q} + s_{1,q}h_{12,q})^2$.

For a 2×2 MIMO system with 4-QAM modulated symbols, there are 4^2 tentative symbols in the search space. According to (7), four multiplications are required for calculating the cost of each of the tentative symbols. Because of the resource constraints in the chosen FPGA, as shown in Fig. 9, we used only four multipliers for the calculation of the cost function and shared the pipelined datapath for calculating the 16 costs. The 16 clock cycle latency of the ML detector is the bottleneck that limits the symbol transmission rate of the MIMO communication system to $F_{\rm clk}/16$ symbols per second, where $F_{\rm clk}$ denotes the clock frequency.

The FIFO section in the ML datapath delays each of the tentative transmitted symbols according to the latency of the cost function datapath so that the cost of each tentative symbol can be augmented with its corresponding symbol. The section search, finds the symbol sHat with the minimum cost, which is the output of the ML detector. Notice that three comparators are used in the search section for finding the tentative symbol with the minimum cost. It was due to the one clock cycle latency of the comparators that the sequence of the costs of tentative symbols was divided into two substreams. In Fig. 9, the minimum costs of the two substreams (along with the tentative symbols corresponding to the minimum costs) are stored in the M1 and M2 registers. The final ML solution, sHat, is chosen based on the minimum cost by comparing the final values of M1 and M2.

IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS



Fig. 10. Test setup for evaluating the baseband performance of MIMO systems on an FPGA board.

VIII. HARDWARE IMPLEMENTATION AND SIMULATION RESULTS

Fig. 10 shows the test setup for evaluating the baseband performance of the implemented MIMO system on the GVA-290 board [39] along with the power source, oscilloscope, and the control computer. This board contains two Xilinx Virtex XCV2000E-BG560-6 FPGAs, two Xilinx Spartan-II FPGAs, four 100 Ms/s analog-to-digital converters, and four 100 Ms/s digital-to-analog converters. The GVA-290 board is interfaced with the control computer through the parallel port. We implemented the entire parameterizable BERT for a 2×2 MIMO system on only one of the Xilinx Virtex-E FPGAs [40], [41]. This FPGA includes 19 200 configurable slices, 160 block memories (BRAMs) with no built-in dedicated multipliers. The parameterizable BERT enables the designers to verify and optimize the transmitter and the receiver algorithms under a wide variety of system parameters such as channel conditions, noise models, modulations, and coding schemes. We developed a GUI through which the BERT can be easily configured for different test scenarios. For example, we can vary the SNR and set the spatial correlation parameters of the analytical MIMO fading channel models. The BERT supports different sample rates and modulation schemes. We can also compare different signal processing algorithms to quantify the BER performance versus system resource tradeoffs. The program can be configured to stop the simulation based on a combination of different criteria including the number of transmitted bits, number of errors, and transmission time. The measured BER performance is exported to MATLAB for graphical display.

Fig. 11 shows the BER performance measurement of the implemented MIMO system. In addition, the floating-point computer simulation of the i.i.d. channel model is shown. To estimate each BER point at each SNR, we measured the performance over at least 1023 seconds of signal transmission on the hardware platform, and when at least 100 errors were collected from the Golay decoder output. In Fig. 11, we can verify that the hardware generated BER results accurately



Fig. 11. BER performance of a 2×2 coded MIMO system measured using the FPGA-based BERT for different channel models.

TABLE I Characteristics of Different Modules on a Xilinx Virtex-E FPGA

Module	Freq. (MHz)	Slices	BRAMs
Source	299	64 (0.3%)	0
Encoder	126	50 (0.3%)	0
Interleaver	117	49 (0.3%)	4 (2.5%)
ML detector	76	947 (4.9%)	0
De-interleaver	115	50 (0.3%)	4 (2.5%)
Decoder	80	438 (2.3%)	0
Fading generator	70	3601 (18.8%)	7 (4.4%)
Noise generator	99	634 (3.3%)	8 (5.0%)
Entire system ^a	52	13436 (70.0%)	47 (29.4%)

match the computer generated BER performance. Even though software development of fixed-point baseband signal processing modules is faster than hardware design and implementation of the BERT, accurate performance measurement of complex baseband sections using a bit-true model on time-varying channels is a computationally daunting process. For example, 10 s of BER measurements using our hardware platform take more than three days of a bit-true software simulation in C running on a 3.6-GHz dual-core Pentium processor with 1 GB of RAM. This corresponds to a speed-up of over 25 000. As shown in Fig. 11, due to the great computational complexity of the bit-true BER performance measurement using software-based simulation, the computer simulation results were given up at 20 dB SNR. Note that simulation times vary significantly and depend on various factors, such as the complexity of the baseband signal processing algorithms (e.g., modulation order, error control codes, detection algorithms, and so on) and the accuracy of the implemented fading channel models [42].

Table I shows the FPGA implementation results of different components. The synthesis results are provided for the Xilinx Virtex-E FPGA. From the results presented in this table, we can conclude that the implemented MIMO communication system (source, encoder, interleaver, detector, deinterleaver, and decoder) uses less than 9% of the available configurable



Fig. 12. Constellation plot of the distorted symbols generated by the (a) SISO channel and (b) 2×2 MIMO channel.

slices on a Virtex-E FPGA while the rest of the system (fading simulator, BER performance measurement, initialization, and interfacing modules) consume a much larger portion of the available resources (more than 60%). On a Xilinx Virtex-E FPGA, our parameterizable BERT for a 2×2 MIMO system uses 69% of the configurable slices, 29% of the BRAMs, and it operates at 52 MHz.

The effects of fading can be conveniently visualized using a constellation plot on an oscilloscope. Fig. 12 shows the two outputs of the fading simulation platform on the oscilloscope screen. The Doppler frequency for these simulations was set to $f_D = 0.5$ Hz so that the changes in the scatter-plot could be easily followed. In Fig. 12(a), the oscilloscope screen shows the scatter-plot of the noisy output of a SISO channel. In this figure, 8-PSK modulated samples are passed through a two-path SISO fading channel and corrupted with AWGN. Fig. 12(b) shows the two noisy outputs of a the 2 × 2 MIMO channel where the transmitted bits are modulated with 4-QAM and the SNR is set to 20 dB.

The implemented fading simulation platform and the BER performance measurement cores along with the analog and digital access to different parts of the system on a GVA-290 board can be used for testing and validation of more complex wireless communication systems. More specifically, with one Virtex-E FPGA dedicated to fading simulation and interfacing, the other on-board FPGAs can be used for a rapid prototyping of wireless communication systems. In addition, the implemented fading simulation and BER performance measurement platform can be easily adapted to faster and more recent FPGA boards for rapid prototyping of wireless communication system in baseband and intermediate frequency.

IX. CONCLUSION

Hardware-accelerated validation is essential to speed up the characterization of computationally intensive and rapidly evolving modern wireless communication systems. This paper presented a parameterizable BERT for a typical single- and multiple-antenna digital baseband communication system on a single FPGA. The BERT uses a MIMO realistic fading channel simulator and a high-quality GNG for faithful performance validations in a laboratory setting. By mapping the computationally intensive signal processing algorithms in the simulation chain to dedicated hardware, the simulation time was reduced by over four orders of magnitude. This BERT system is flexible enough to be reconfigured for adapting the new specifications of emerging standards and is scalable to support various configurations. In addition, this measurement system demonstrates how rapid prototyping can be used to minimize reliance on expensive test equipment and timeconsuming field trials.

REFERENCES

- M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, Simulation of Communication Systems: Modeling, Methodology, and Techniques. Boston, MA, USA: Kluwer, 2000.
- [2] T. S. Rappaport, Wireless Communications: Principles and Practice. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [3] V. Singh, A. Root, E. Hemphill, N. Shirazi, and J. Hwang, "Accelerating bit error rate testing using a system level design tool," in *Proc. 11th Annu. IEEE Symp. FCCM*, Apr. 2003, pp. 62–68.
- [4] Y. Guo and D. McCain, "Compact hardware accelerator for functional verification and rapid prototyping of 4G wireless communication systems," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, 2004, pp. 767–771.
- [5] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "Hardware-based error rate testing of digital baseband communication systems," in *Proc. IEEE ITC*, Oct. 2008, pp. 1–10.
- [6] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "A flexible layered architecture for accurate digital baseband algorithm development and verification," in *Proc. DATE*, 2009, pp. 45–50.
- [7] L. G. Barbero and J. S. Thompson, "Rapid prototyping system for the evaluation of MIMO receive algorithms," in *Proc. IEEE EUROCON*, Nov. 2005, pp. 1779–1782.
- [8] Hardware Acceleration of 3GPP Turbo Encoder/Decoder BER Measurements Using System Generator, Xilinx, San Jose, CA, USA, Dec. 2006.
- [9] Using Simulink, The Mathworks Inc., Natick, MA, USA, 2007.
- [10] Y. R. Zheng and C. Xiao, "Improved models for the generation of multiple uncorrelated Rayleigh fading waveforms," *IEEE Commun. Lett.*, vol. 6, no. 6, pp. 256–258, Jun. 2002.
- [11] D.-U. Lee, W. Luk, J. D. Villasenor, and P. Y. K. Cheung, "A Gaussian noisegenerator for hardware-based simulations," *IEEE Trans. Comput.*, vol. 53, no. 12, pp. 1523–1534, Dec. 2004.
- [12] 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, Jun. 2006.
- [13] A. Alimohammad, S. F. Fard, B. F. Cockburn, and C. Schlegel, "A compact and accurate Gaussian variate generator," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 5, pp. 517–527, May 2008.
- [14] Additive White Gaussian Noise (AWGN) Core v1.0, Xilinx Inc., San Jose, CA, USA, 2002.
- [15] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "A compact Rayleigh and Rician fading simulator based on random walk processes," *IET Commun.*, vol. 3, no. 8, pp. 1333–1342, 2009.
- [16] A. Alimohammad, S. F. Fard, B. F. Cockburn, and C. Schlegel, "A novel technique for efficient hardware simulation of spatiotemporally correlated MIMO fading channels," in *Proc. IEEE Int. Conf. Commun.*, May 2008, pp. 718–724.
- [17] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "An accurate MIMO fading channel simulator using a compact and high-throughput reconfigurable architecture," *IET Commun.*, vol. 5, no. 6, pp. 844–852, Apr. 2011.
- [18] S. F. Fard, A. Alimohammad, and B. F. Cockburn, "An FPGA-based simulator for high path count Rayleigh and Rician fading," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2725–2734, Jul. 2010.
- [19] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2004.
- [20] B. Sklar, Digital Communications, Fundamentals and Applications. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
- [21] R. C. Tausworthe, "Random numbers generated by linear recurrence modulo two," *Math. Comput.*, vol. 19, no. 90, pp. 201–209, 1965.
- [22] I. Vattulainen, K. Kankaala, J. Saarinen, and T. Ala-Nissila, "A comparative study of some pseudorandom number generators," *Comput. Phys. Commun.*, vol. 86, no. 3, pp. 209–226, 1995.
- [23] P. L'Ecuyer, "Good parameter sets for combined multiple recursive random number generators," *Annal. Operations Res.*, vol. 47, no. 1, pp. 159–164, 1999.

10

IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS

- [24] P. L'Ecuyer, "Tables of maximally equidistributed combined LFSR generators," *Math. Comput.*, vol. 68, no. 225, pp. 261–269, 1999.
- [25] D. G. Hoffman, D. A. Leonard, C. C. Lindner, K. T. Phelps, C. A. Rodger, and J. R. Wall, *Coding Theory, The Essentials*. New York, NY, USA: Marcel Dekker, 1991.
- [26] P. Bello, "Characterization of randomly time-variant linear channels," *IEEE Trans. Commun.*, vol. 11, no. 4, pp. 360–393, Dec. 1963.
- [27] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "Hardware implementation of Rayleigh and Ricean variate generators," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 99, no. 4, pp. 1–5, Jan. 2010.
- [28] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "Hardware implementation of Nakagami and Weibull variate generators," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 7, pp. 1276–1284, Jul. 2012.
- [29] M. Pätzold, Mobile Fading Channels. New York, NY, USA: Wiley, 2002.
- [30] M. Kiessling and J. Speidel, "Statistical transmit processing for enhanced MIMO channel estimation in presence of correlation," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2003, pp. 2411–2415.
- [31] J. P. Kermoal, L. Schumacher, K. I. Pedersen, P. E. Mogensen, and F. Frederiksen, "A stochastic MIMO radio channel model with experimental validation," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 6, pp. 1211–1226, Aug. 2002.
- [32] W. Weichselberger, M. Herdin, H. Ozcelik, and E. Bonek, "A stochastic MIMO channel model with joint correlation of both link ends," *IEEE Trans. Wireless Commun.*, vol. 5, no. 1, pp. 90–99, Jan. 2006.
- [33] A. M. Sayeed, "Deconstructing multiantenna fading channels," *IEEE Trans. Signal Process.*, vol. 50, no. 10, pp. 2563–2579, Oct. 2002.
- [34] B. Holter and G. E. Oien, "On the amount of fading in MIMO diversity systems," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2498–2507, Sep. 2005.
- [35] S. F. Fard, A. Alimohammad, and B. F. Cockburn, "A versatile fading simulator for on-chip verification of MIMO communication systems," in *Proc. IEEE Int. SOCC*, Sep. 2009, pp. 412–415.
- [36] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *Anna. Math. Stat.*, vol. 29, no. 2, pp. 610–611, 1958.
- [37] N. Chernov, C. Lesort, and N. Simanyi, "On the complexity of curve fitting algorithms," J. Complex., vol. 20, no. 4, pp. 484–492, Aug. 2004.
- [38] J. G. Proakis, *Digital Communications*, 4th ed., New York, NY, USA: McGraw-Hill, 2001.
- [39] GVA-290 Xilinx VirtexE Hardware Accelerator, GV & Associates Inc., Anchorage, AK, USA, 2009.
- [40] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "FPGA-accelerated baseband design and verification of broadband MIMO wireless systems," in *Proc. IEEE 1st Int. Conf. Adv. Syst. Testing Validation*, Sep. 2009, pp. 135–140.

- [41] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "FPGA-based accelerator for the verification of leading-edge wireless systems," in *Proc. IEEE Int. DAC*, Jul. 2009, pp. 844–847.
- [42] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "Reconfigurable performance measurement system-on-a-chip for baseband wireless algorithm design and verification," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 84–91, Dec. 2012.



Amirhossein Alimohammad received the M.Sc. degree from the University of Tehran, Tehran, Iran, and the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada.

He is an Assistant Professor with the Electrical and Computer Engineering Department, San Diego State University, San Diego, CA, USA. He was the Co-Founder and Chief Technology Officer of Ukalta Engineering, Edmonton, from 2009 to 2011. He was a Post-Doctoral Fellow with the University of

Alberta from 2007 to 2009. He was a Hardware Engineer with Get2Chip GmbH, a Research Fellow in the Institute of Microelectronics, University of Ulm, Ulm, Germany, and Atmel Wireless, Germany. His current research interests include digital VLSI systems, reconfigurable architectures, wireless communication circuits, and signal processing algorithms.



Saeed Fouladi Fard received the B.Sc. degree in electrical engineering from the Faculty of Communications, Tehran, Iran, in 2000, the M.Sc. degree in electrical engineering from the University of Tehran, Tehran, in 2003, and the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2009.

He was the VP of engineering with Ukalta Engineering, Edmonton. His current research interests include signal processing, communications, reconfigurable computing, VLSI design and testing, mul-

tiple antenna transceivers, fading simulation, and efficient hardware computation techniques.