

# A Versatile High Speed Bit Error Rate Testing Scheme

Yongquan Fan, Zeljko Zilic, and Man Wah Chiang  
Microelectronics and Computer System Laboratory, McGill University  
{yfan, zeljko, manwah}@macs.ece.mcgill.ca

## Abstract

The quality of a digital communication interface can be characterized by its bit error rate (BER) performance. To ensure the quality of the manufactured interface, it is critical to quickly and precisely test its BER behavior. Traditionally, BER is evaluated using software simulations, which are very time-consuming. Though there are some standalone BER test products, they are expensive and none of them includes channel emulators, which are essential to testing BER under the presence of noise. To overcome these problems, we present a versatile scheme for BER testing in FPGAs. This scheme consists of two intellectual property (IP) cores: the BER tester (BERT) core and the additive white Gaussian noise (AWGN) generator core. We demonstrate through case studies that the proposed solution exhibits advantages in speed and cost over existing solutions.

## 1. Introduction

While modern interconnect evolves to gigabit domain, high speed serial communication interfaces are widely used in backplane and chip-to-chip applications, as they eliminate the clock skew problem by encoding the clock signal in data streams. High speed serial interfaces are increasingly used in many FPGAs and ASICs. According to International Technology Roadmap for Semiconductors (ITRS), the requirements for the test of these interfaces are continually putting pressure on the test industry. Currently, testing the functionality of the serial interfaces is done by standalone pattern-generators and bit-error-rate detectors [1], which are expensive and time consuming.

For a digital communication system, the *channel* serves as the physical medium used to send the signal from the transmitter to the receiver. One essential feature of the communication channel is that the transmitted signal is corrupted in a random manner. As a measure of how well the overall communication system performs, BER is the probability of a bit-error at the output of the receiver, whose importance has been widely recognized [11]. Traditionally, BER has been evaluated using software simulations, where the real communication system is emulated by its software model. Although software simulations are easy to set up, they are time consuming. A hardware-based solution is commonly 100,000 to 1

million times faster than the best simulation software at the same abstraction level [2]. While there exist products available for BER testing [3], [4], they are expensive and none of them includes AWGN channel emulators. Such testers are difficult to set up for BER testing under the presence of noise.

In order to solve the above problems, in this paper we propose a versatile low cost, high speed scheme for BER testing in FPGAs, suitable for design and manufacturing quality control. The scheme can be used to test the performance of a wide range of communication devices, including native clock data recovery (CDR) interfaces, as well as various user-defined modulation, spread spectrum and error correcting cores. Two challenging testing cases are successfully conducted using the scheme.

The paper is organized as follows: Section 2 first outlines the background of BER testing, and then exposes our scheme for BER testing. We present the detailed implementation of the BERT core and the AWGN core in Section 3 and Section 4, respectively. In Section 5, we demonstrate the advantages of our scheme through case studies. Conclusions are presented in Section 6.

## 2. Proposed scheme for BER testing

### 2.1 AWGN channel model

The physical channel of a digital communication system may be a pair of wires, an optical fiber, or any other communication medium. A mathematical model can be used to capture the characteristics of the transmission media. In communication system analysis and design, AWGN channel represents a broad class of physical channels. The AWGN mathematical model is shown in Figure 1 [5].

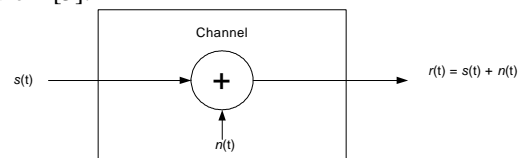


Figure 1: The AWGN channel model

In this model, the transmitted signal  $s(t)$  is corrupted by noise  $n(t)$ . The noise is introduced by the channel, as well as by electronic components, including amplifiers at the receiver. This type of noise is most often characterized as a thermal noise, or statistically as a Gaussian noise.

## 2.2 BER and SNR

Traditionally, BER is related both theoretically and practically to the signal-to-noise ratio (SNR). SNR is the fundamental input quantity that determines the channel capacity  $C$  for a given bandwidth  $B$ , according to the fundamental Shannon law:

$$C = B \log_2(1 + SNR)$$

In practice, communication system designers balance between bandwidth and SNR to maximize the channel capacity for an acceptable BER performance. There are several types of communication systems in which this balancing act is played in different ways.

In *baseband* transmission, the data and clock are transmitted as digital waveforms. Baseband schemes, such as commonly used non-return-to-zero (NRZ) CDR encoding, combine clock and data signals at the transmitting side and decouple them at the receiver. Careful timing extraction leads to a reduction in the number of transmission errors, which is equivalent to an increase in the system SNR. The theoretical relationship between BER and SNR can be characterized by

$$BER = Q(\sqrt{SNR})$$

where  $Q(\cdot)$  is  $Q$  function, which represents the area under the tail of the Gaussian density function [5].

Another class of communication systems employs *modulation* schemes for communication over a given portion of spectrum. The modulator at the transmitter performs the function of mapping the digital sequence into sinusoidal signal waveforms. The BER performance of receivers varies widely, depending on the modulation scheme. For example, assuming a Gray code is used, the BER performance of the Quadrature Phase Shift Keying (QPSK) modulation can be theoretically characterized by

$$BER \approx Q(\sqrt{2SNR})[1 - 0.5Q(\sqrt{2SNR})]$$

In the QPSK scheme, as the baseband digital signal is modulated by a complex exponential (sine and cosine waves), two real-valued data streams appear and have to be processed separately. They are referred to as  $I$  (In phase) channel and  $Q$  (Quadrature) channel.

*Spread spectrum* technique is yet another implementation of the Shannon law by which the transmitted signal bandwidth  $B$  is much greater than the information bandwidth  $C$ . This excess bandwidth is used to protect the signal from interference caused by multiple users, as well as from the intentional jamming.

In all such implementations, it is desirable to obtain quickly the  $BER$  performance of a manufactured device.

## 2.3 Our Scheme for BER Testing

To facilitate testing various communication interfaces, we have proposed a scheme for BER testing in FPGAs. The block diagram of the scheme is shown in Figure 2.

The solution combines a BERT and an AWGN generator in a single FPGA device. The communication channel is emulated by adding AWGN variables to the transmitted streams. The amplitude of the noise is adjustable. Hence, we can emulate an AWGN channel, in which signals are transmitted with different SNRs.

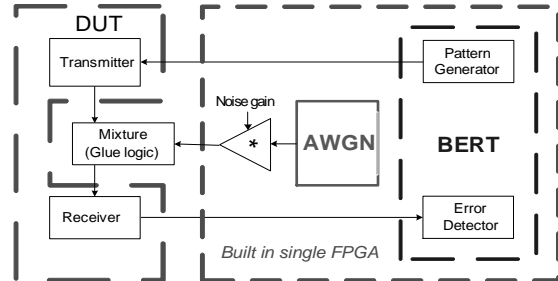


Figure 2: Block diagram of our BER testing scheme

## 3. BERT core design

### 3.1 Serial BERT design

The structure of our serial BERT is shown in Figure 3. In this scheme,  $shift\_reg1$  and  $XOR1$  forms the pattern generator, which generates pseudo random bit sequences (PRBSs) and sends them to the DUT.

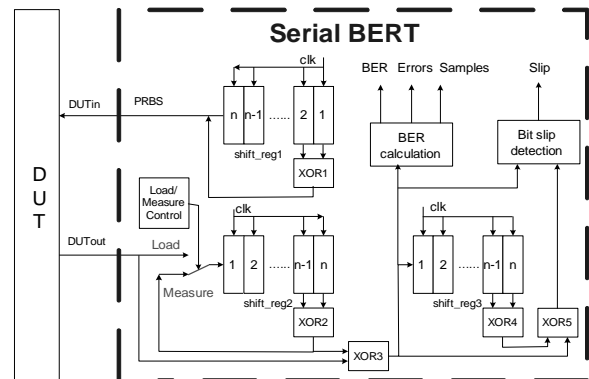


Figure 3: Block diagram of the serial BERT

Before a measurement begins, the *load/measure* switch is set in *load* state until  $shift\_reg2$  is fully loaded with the contents of  $shift\_reg1$ . The switch,  $shift\_reg2$  and  $XOR2$  are used for synchronization by replicating delayed PRBSs as the reference pattern. During the synchronization process, it is assumed that all the bits are correctly transmitted. Otherwise, it should be repeated. The gate  $XOR3$  serves as the error detector. Its output is monitored every clock cycle: if a '1' is detected, a transmission error is counted; otherwise, the transmission is error-free.

In the above scheme, a bit slip (loss or repeat) will result in infinite errors, as the output of  $XOR3$  is PRBSs. Based on the fact that the addition or superimposition of

two PRBSs shifted in phase produces another PRBS [6], *XOR4*, *XOR5* and *shift\_reg3* serve for bit slip detection. Bit slips can be detected by monitoring the outputs of *XOR3* and *XOR5*. In cases of bit slips, the synchronization needs to be reconstructed.

### 3.2 Parallel BERT design

A parallel BERT is used to test parallel communication interfaces by sending pseudo-random word sequences (PRWSs) to DUTs. Basically, a  $k$ -bit ( $0 \sim k-1$ ) parallel BERT can be built by combining  $k$  independent serial BERTs that have the same load time. Figure 4 shows the block diagram of the proposed parallel BERT, in which only the serial BERT circuitry for bit0 is used for *load/measure* control and word slip detection.

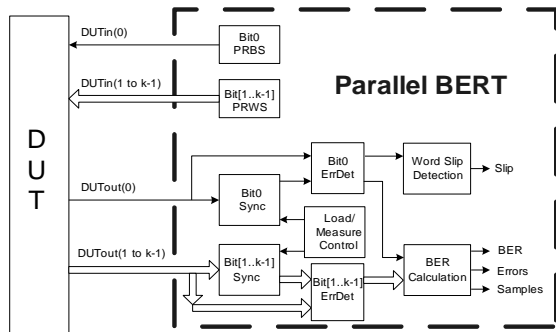


Figure 4: Block diagram of the parallel BERT

### 3.3 Synthesis Results

The BERT designs are built in VHDL, and can target almost any FPGA devices. The synthesis has been done using Quartus II tools by Altera. Table 1 shows the synthesis results of the parallel BERT design based on the Altera Mercury FPGA EMP120 device.

Table 1: Synthesis results of the BERT

Logic Elements	ESB Bits	fmax
384/4800( 8% )	0/49152 ( 0 )	160.3 MHz

As can be seen, the BERT only occupies a small part of the FPGA device. There are enough resources in the FPGA to implement other user-specified functions.

## 4. AWGN core design

### 4.1 Method overview

Existing methods of generating AWGN in digital hardware include CLT method, Polar method, mixed method and CA-based methods.

The CLT method is based on central limit theorem (CLT), where a large number of uniformly distributed random variables (UDRVs) are added to generator a

Gaussian variable. The CLT method usually employs an accumulator [7], which greatly limits the output rate, so it is not suitable for high-speed applications.

As presented in [8], the Box-Muller method is based on Box-Muller algorithm. The design requires many considerations regarding the number of sampling recursions and relative position of points, as well as the precision of implementation. Efficient implementation is hence not straightforward. The mixed method uses the combination of the CLT method and the Box-Muller method for high accuracy, but it slows down the output rate. For the generator in [8], when  $N=4$ , the output rate is reduced to 24.5MHz, while its clock rate reaches 98MHz.

CA-based methods use a more efficient scheme called cellular automata (CA) [9] instead of LFSRs to generate a large number of UDRV. The UDRVs are then transformed to Gaussian variables based on CLT or other algorithms [10]. However, the transformation is difficult to implement for applications where high speed and high precision are required.

In order to overcome the disadvantages of the existing methods, we propose our method for AWGN generation. Our method consists of Polar method as shown in Algorithm 1 and our CLT method.

1. Do
2. Generate two UDRV,  $U_1$  and  $U_2$ , between  $[0,1]$
3. Set:  $V_1 = (2*U_1)-1$  and  $V_2 = (2*U_2)-1$
4. Set:  $S = V_1^2 + V_2^2$
5. If  $S >= 1$ , go back to line 2
6. Loop until  $S < 1$
7. Set:  $W = \sqrt{-2\ln(s)/s}$
8. Generate two Gaussian variables

$$X_1 = V_1 * W \quad X_2 = V_2 * W$$

Algorithm 1: Polar method [16]

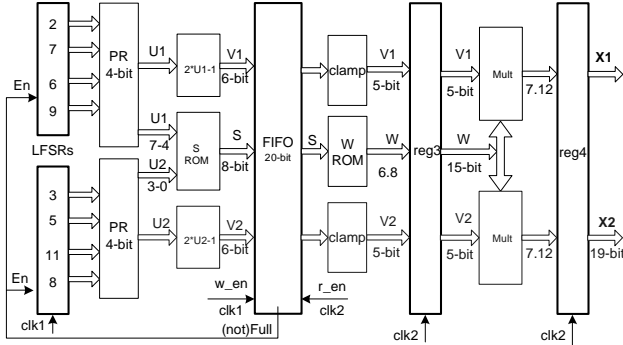
The Polar algorithm is faster than the Box-Muller algorithm because it uses fewer transcendental functions. Polar method generates two independent Gaussian variables at the same time. This is especially suitable for applications like QPSK transmission where two communication channels are needed.

### 4.2. Architecture

**4.2.1 Polar method.** Based on Algorithm 1, the block diagram of Polar method for two AWGN generators is developed and shown in Figure 5. It can easily be simplified to implement a single generator.

We use eight independent LFSRs to generate two 4-bit UDRVs,  $u_1$  and  $u_2$ . All bits represent fractional parts, so  $u_1$  and  $u_2$  are uniformly distributed over  $[0, 0.9375]$ .  $v_1$

and  $v_2$  are generated using signed adders. As computing  $S$  involves lots of additions and multiplications, we use ROM-based design, where the concatenation of  $U_1$  and  $U_2$  is set to be the address and  $S$  is set to be the data stored in the ROM. If computed  $S \geq 1$ , zero is stored in the ROM.



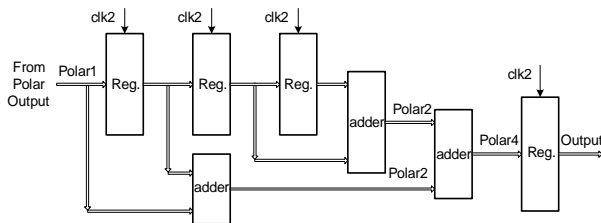
**Figure 5: Block diagram of Polar method**

On average, the Do Loop is executed 1.3 times of line 7 and line 8 in Algorithm 1. We insert a synchronizing FIFO to achieve a constant output rate. Two clock signals are used in the FIFO,  $clk1$  for writing and  $clk2$  for reading. Writing is enabled when  $S$  is not equal to 0, while reading is always enabled. The LFSRs are disabled when the FIFO is full. By setting the depth of the FIFO to be 16 and  $clk2$  to be half of  $clk1$ , we achieve a constant output rate.

We also use ROM-based design to calculate  $W$ , where  $S$  denotes the address and  $W$  denotes the data stored in the ROM. As  $S$  is between  $[0.00390625, 0.99609375]$ ,  $W$  is between  $[54.2835, 0.0886]$ . In binary code, we use 14 bits to represent  $W$ , 6 for the integer and 8 for the fraction.

Finally, two signed multipliers are used to generate two Gaussian variables,  $x_1$  and  $x_2$ . Each variable is 19 bits in width, 1 bit for the sign, 6 bits for the integer and 12 bits for the fraction. The width can be truncated according to applications.

**4.2.2 Our CLT Method.** Traditionally, CLT method uses an accumulator, which slows down the output speed by a factor of  $N$ , where  $N$  is the number of accumulated variables. We propose our CLT method shown in Figure 6, which does not exhibit the speed penalty while improving accuracy.



**Figure 6: Our CLT method (N=4)**

## 4.3 Experiment results

**4.3.1 Statistical Properties.** Statistical properties of an AWGN generator should be based on evaluating samples from at least one period. The period of our generator may reach around  $2^N$ , where  $N$  is the sum of the lengths of all LFSRs; in our case,  $N$  equals to 51. Statistically evaluating  $2^{51}$  samples requires a lot of hardware resources and time. Our experiments demonstrate that statistical results of thousands of samples are a good approximation. Table 2 shows the  $Q(x)$  accuracy of our generators with 10,000 and 500,000 samples.

**Table 2:  $Q(x)$  relative error of our generators**

$x$	Theory $Q(x)$	Relative Error of Our Generators		
		Figure 5 10,000	Figure 5 + Figure 6	
			10,000	500,000
0	0.5000	2.76 %	1.02 %	0.24%
0.2	0.4207	-2.50 %	0.50 %	0.42%
0.4	0.3446	1.69 %	0.26 %	0.55%
0.6	0.2743	-0.10 %	1.06 %	0.80%
0.8	0.2119	1.88 %	1.74 %	1.09%
1.0	0.1587	4.70 %	3.21 %	1.20%
1.2	0.1151	-7.17 %	3.99 %	1.49%
1.4	0.0808	-4.29 %	4.95 %	1.85%
1.6	0.0548	-3.06 %	6.57 %	2.37%

As can be seen, our method with the parameters shown in Figure 5 (simplified to a single generator) implements a high precision AWGN generator even with a limited number of samples. Our CLT method shown in Figure 6 can further smoothen the variation of the distribution. Also note that the relative error of  $Q(x)$  decreases when the number of samples increases. Limited number of samples is the main reason for the error.

**4.3.2 Synthesis results.** Table 3 shows the synthesis results with the structure shown in Figure 5.

**Table 3: Synthesis results**

FPGA Device	Logic Elements	ESB Bits	Output Rate
EP1M120F4 84C7	336/4800 (7%)	5856/49152 (11%)	73.48 MHz

## 5. Case studies

### 5.1. High speed serial interface testing

In this section, we present a low cost scheme to test gigabit serial interfaces included in Altera Mercury FPGA devices. The Altera Mercury gigabit transceiver is implemented in the high-speed differential interface (HSDI) to transmit and receive high-speed serial data

streams (up to 1.25Gbps). Figure 7 shows the block diagram of one of the eight HSDI transceiver channels (channel 4) of a Mercury EP1M120 device [12]. Each HSDI transceiver consists of a transmitter and a receiver. The transmitter includes a synchronizer and a serializer; the receiver includes a clock recovery unit (CRU), a deserializer, and a synchronizer. The HSDI PLL circuitry is dedicated to providing clock signals for the transceiver. Based on its structure, the setup to test the functionality of the transceiver is developed and shown in Figure 8.

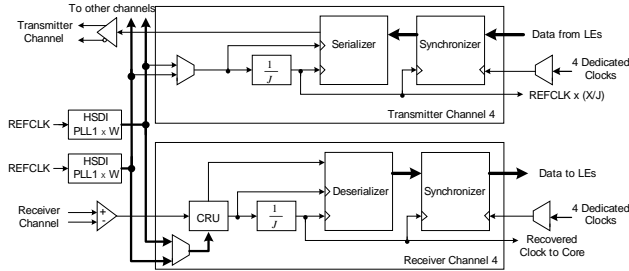


Figure 7: HSDI transceiver block diagram

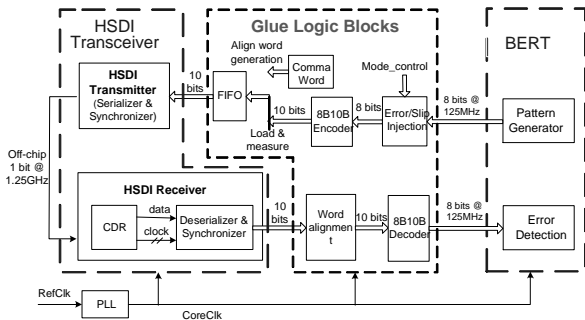


Figure 8: Testing setup for HSDI transceiver

In the testing setup, the PLL, the HSDI transmitter and the HSDI receiver can only be built by instantiating mega functions [13], [14]. The data width of the BERT is 8 bits. The glue logic is developed to interface the BERT and the transceiver. The *Error/Slip Injection* block inserts errors or word slips for the purpose of demonstration. The *8b10b encoder* encodes the 8-bit sequences to 10-bit sequences to ensure enough bit transitions [15]. A FIFO is used to make sure that there is always data ready for transmission after a testing begins. Comma words are inserted at the start of the testing for word alignment. The *8B10B Decoder* recovers the 8-bit PRWSs sent by the BERT.

The testing setup is implemented in VHDL, targeting the EP1M120F484C7AES device using Quartus II. The synthesized results are downloaded onto an Altera Mercury HSDI CDR Demo board. The outputs of the transmitter are connected to the inputs of the receiver by two SMA cables.

We obtained zero BER from both simulations and running real test in the board when no error or bit slip was injected. The zero BER experiment results demonstrate

the functional correctness of the HSDI transceiver and the feasibility of the testing setup.

## 5.2 Baseband transmission testing

Based on the AWGN model discussed in Section 2.1, the testing setup for a digital baseband communication system is developed and shown in Figure 9.

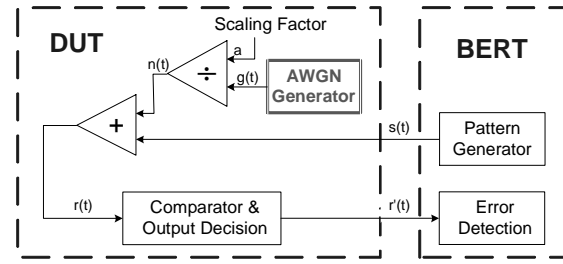


Figure 9: BER testing setup for digital baseband

In this system, one is used to transmit data ‘1’ and zero is used to transmit data ‘0’. Assuming that data 1s and 0s have equal occurring probability, the average energy of transmitted signals is

$$E = (E_0 + E_1) / 2 = 0.5$$

In the testing setup, the channel is emulated by scaling the AWGN generator with zero mean and unity variance by a factor of  $a$ . In this case, the energy of the noise is

$$N_o = 2 / a^2$$

Hence, we have

$$SNR = E / N_o = a^2 / 4$$

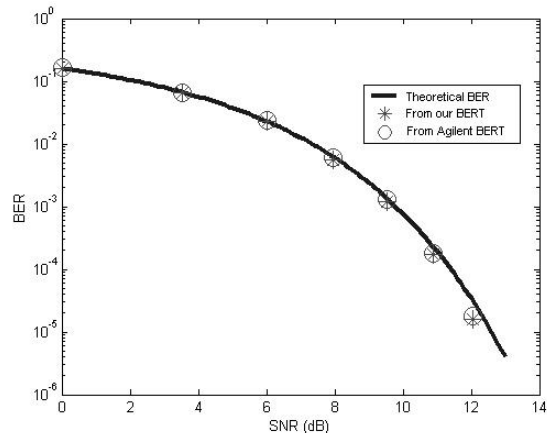
As shown in Figure 9, the transmitter consists of the pattern generator, and the receiver consists of the comparator and output decision block. If the noise corrupted signal  $r(t)$  is bigger than 0.5 (threshold voltage),  $r'(t)$  is set 1; otherwise,  $r'(t)$  is 0.

Table 4 lists the test results. The measurements are taken while running the testing setup in an Altera Mercury FPGA board with a clock of 50MHz. The measurement is first done using our BERT, and then an Agilent BERT.

Table 4: BER measurements for digital baseband

$a$	SNR (dB)	Total bits	Our BER	Agilent BER	Error (%)
2	0	2024	1.62e-1	1.65e-1	1.85
3	3.52	12448	6.58e-2	6.61e-2	0.45
4	6.02	12448	2.32e-2	2.40e-2	3.33
5	7.96	20000	5.65e-3	5.32e-3	4.32
6	9.54	1000000	1.20e-3	1.31e-3	8.40
7	10.88	1000000	1.76e-4	1.83e-4	3.83
8	12.04	10000000	1.62e-5	1.78e-5	8.99

Figure 10 shows the plot of the theoretical BER, the measured BER using our BERT and an Agilent 81200 BERT as a function of input SNR.



**Figure 10: Plot of measured BER and theoretical BER**

As can be seen, the measured BER using our BERT perfectly coincides with that from the expensive Agilent BERT, and closely matches the theoretical BER. This match can be further improved by optimizing the threshold voltage setting or increasing the number of samples. The plot demonstrates the validity of our scheme for BER testing. In the above testing, it takes less than one second to generate the point at  $1.62e-5$  BER; in software simulations, this usually takes days. The proposed scheme exhibits astronomical advantage in speed over traditional software simulation methods.

Although the experiment is based on testing a digital baseband system, the proposed BER testing scheme applies to any AWGN digital transmission systems. Furthermore, the AWGN module can be modified by adding appropriate filters to emulate more complex channels, such as Rayleigh channels.

## 6. Conclusions

In this paper, a versatile high speed BER testing scheme is presented, suitable for the testing of a wide range of digital communication interfaces. Compared to traditional software simulations, the proposed scheme is a few orders of magnitude faster. Compared to traditional standalone BERT and ATE equipment, the proposed solution is much cheaper. Furthermore, FPGA-based solution makes it easy to interface different DUTs.

## References

[1] ITRS, *The International Technology Roadmap for Semiconductors*, 2001 Edition

[2] M. Courtoy, "Rapid System Prototyping for Real-time Design Validation," *Proc. Ninth International Workshop on Rapid System Prototyping*, pp. 108-112, 1998.

[3] Agilent Technologies, *Agilent 81200 Data Generator/Analyzer User Guide, Revision 3*, Palo Alto, California, 2002

[4] Anritsu Corporation. *48 Gb/s BER Test System Datasheet*. Atsugi-shi, Kanagawa, Japan, 2002

[5] J. G. Proakis, *Digital Communications*, McGraw-Hill High Education, 2001

[6] R. Kiefer, *Test Solutions for Digital Networks*, Huthig GmbH, Heidelberg, 1998

[7] P. Atinirami, *Design and implementation of an FPGA-based adaptive filter single-use receiver*, Master Thesis, Department of Electrical Engineering, Virginia Polytechnic Institute and State University, 1999

[8] A. Gazel, E. Boutillon, J.L. Danger, G. Gulak and H. Lamaari, "Design and Performance Analysis of a High Speed AWGN Communication Channel Emulator", *IEEE PACRIM Conference*, Victoria, B.C., Canada, Aug. 2001

[9] S. Wolfram, "Random Sequence Generation by Cellular Automata," *Advances in Applied Mathematics*, vol. 7, pp. 123-169, June 1986

[10] J. Chen, J. Moon and K. Bazargan, "A Reconfigurable FPGA-based Readback Signal Generator for Hard-drive Read Channel Simulator," *Proceedings of 39th Design Automation Conference*, pp. 349-354, June 2002

[11] E.A. Newcombe, and S. Pasupathy, "Error Rate Monitoring for Digital Communications", *Proc. IEEE*, vol. 70, no.8, pp. 805-825, Aug. 1982.

[12] Altera Corporation, *Mercury Programmable Logic Device Family Data Sheet*, San Jose, California, Jan. 2003

[13] Altera Corporation, *Introduction to Quartus II Manual*, San Jose, California, July 2003

[14] Altera Corporation, *Mercury Gigabit Transceiver MegaCore Function User Guide*, San Jose, California, February 2002

[15] Altera Corporation, *8B10B Encoder/Decoder MegaCore Function User Guide*, San Jose, California, December 2002

[16] D. E. Knuth, *The Art of Computer Programming*. Addison-Wesley, 1998