American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Architectural Design and Implementation of Bit Error Rate Tester on FPGA

Mann Aladwany[a]*, Zahraa Alsaegh[b]

[a,b]*Mousl University, Dept. of Electrical/Electronic, Erbil 44001, Iraq*
[a]*Email: maanaladwany@yahoo.com*
[b]*Email: z_alsaegh@yahoo.com*

Abstract

FPGAs have witnessed an increased use of dedicated communication interfaces. With their increased use, it is becoming critical to test and properly characterize all such interfaces. Bit error rate (BER) characteristic is one of the basic measures of the performance of any digital communication system. This thesis presents a scheme for BER testing in FPGAs, with a few orders of magnitude speedup compared to other tradition methods. Where the using of hardware emulation by FPGA is very interested for most researches and designers in present time, because it has properties such as flexibility in reprogramming it and prepare it according to the user need, so we chose this technique in this thesis as an implementation environment to the proposed scheme. The proposed scheme mainly consists of an essential scheme core: a bit error rate tester (BERT), by using MATLAB Simulink software in the beginning, then composition this software on FPGA chip (spartan-6 type) by using ISE 14.2 software. The results shows very good agreement between software representation and the implementation on FPGA chip.

Keywords: FPGAs; BER; BERT; spartan-6 type; ISE 14.2.

1. Introduction

In digital communications the concept of error rate measurements has been well established as a way to verify the quality of a transmission link. The transmitted signals usually tend to suffer from noise, reduction in power and other disturbances that result in erroneous bits in the receiver. This effect can be quantified by bit-error-rate (BER) which is measured as a ratio of the incorrect received bits to all bits sent.

---

* Corresponding author.

The BER measurements are useful to characterize various communication channels (wireless and wireline) and also to test hardware, in particular the digital receivers or transceivers. The existing standards define the respective test requirements in different variants. For example, radio receivers can be tested for BER in terms of sensitivity, interference, blocking or intermodulation [1]. The specified BER values tend to be small such as $1e^{-4}$… $1e^{-8}$ (and even $1e^{-15}$ in case of SerDes devices) so the test signals capable of measuring BER are usually long bit sequences. As a rule of thumb an adequate test sequence should comprise at least (10...100) × BER^{-1} bits.

Bit error rate (BER) is the ratio of the number of incorrect to the total number of received bits. For qualifying the reliability of an entire digital communication system from “bits in” to “bits out”, BER characteristic is the fundamental measure of the performance of a digital communication system. As shown in figure 1, a digital communication system consists of a transmitter, a channel, and a receiver. The transmitter changes the raw information (sequences of binary digits) into a format that is matched to the characteristics of the channel. Depending on applications, the transmitter may consist of a source encoder, an encryptor, a channel encoder, a carrier modulator or a spread-spectrum modulator. The receiver accepts the signal from the channel and recovers the transmitted binary digits. The recovered digits are usually processed to permit interfacing with the final destination, such as a computer monitor or the human ear. The channel is the physical medium used to send the signal from the transmitter to the receiver. The medium may be the air, wire lines, optical fiber and so on. One essential feature of the communication channel is that the transmitted signal is corrupted in a random manner by a variety of possible mechanisms, such as additive thermal noise generated by electronic devices; man-made noise, e.g., automobile ignition; and atmospheric noise, e.g., electrical lightning discharges during thunderstorms.

Figure 1: block diagram of a digital communication system

In a digital communication system, either the channel or the communicating devices (sending and/or receiving end) can introduce distortion or cause errors. As modern communication interfaces are quite complex, besides inherent device and timing imperfections, the correctness and performance of communication interfaces depend on many design choices, such as types of waveforms used to transmit the information over the channel, the transmitter power, the characteristics of the channel (i.e., the amount of noise, the nature of the interference), and the method of demodulation and decoding. BER is a fundamental measure of the communication system performance, whose importance has been widely recognized [2]. 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, compared with the input of the transmitter.

1.1. Structure of BERT mechanism

All BER testers use the same basic principle: known test patterns (e.g. PRBSs, or PRWSs) are sent to a DUT, and the patterns are compared bit by bit with the output of the DUT (device under test) after a certain period of time, as shown in figure 2. The comparison process is synchronized at the start of the measurement. BER testing methods include software simulation and hardware emulation. In software simulations, each component of the communication system is built using software models; while with hardware emulation, all components are built in hardware.

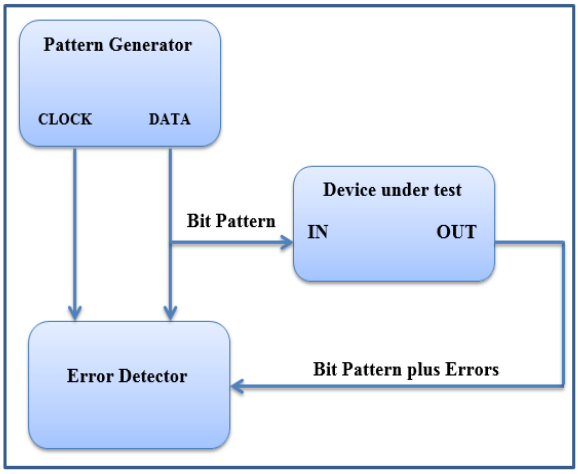

Figure 2: The basic block diagram of a BERT (BER tester) design

The DUT can be any communication interface or system that receives bit or word sequences and then restores the sequences after some signal processing or format changes, such as a transceiver (a transmitter and a receiver), the combination of a modulator and a demodulator, or the integration of an encoder and a decoder. Then the BER is measured by the ratio:

$$\text{BER measured} = \frac{Error\ Cout}{Transmitted\ Bits} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

1.2. Proposed BER Testing Scheme

1.2.1. Serial BER tester design

The practical design which has been proposed for the BERT is shown in Figur 3. It was built for two types, Serial BERT(where data transmit is serial) and Parallel BERT(where data transmit is parallel). It was built using simulation models by MATLAB software.

Then we use the field programmable gate array (FPGA) for the implementation of this model, because they provide the flexibility and processing speed as well as the lack of cost [3]. As the FPGA technology is the perfect solution for the implementation of the models that require implementation in real time, so that is characterized by its propensity to parallel execution as well as the possibility of re-programmed and configured to suit any possible alteration [4]. The proposed design produce an ease examination of BER for different types of communication interfaces.

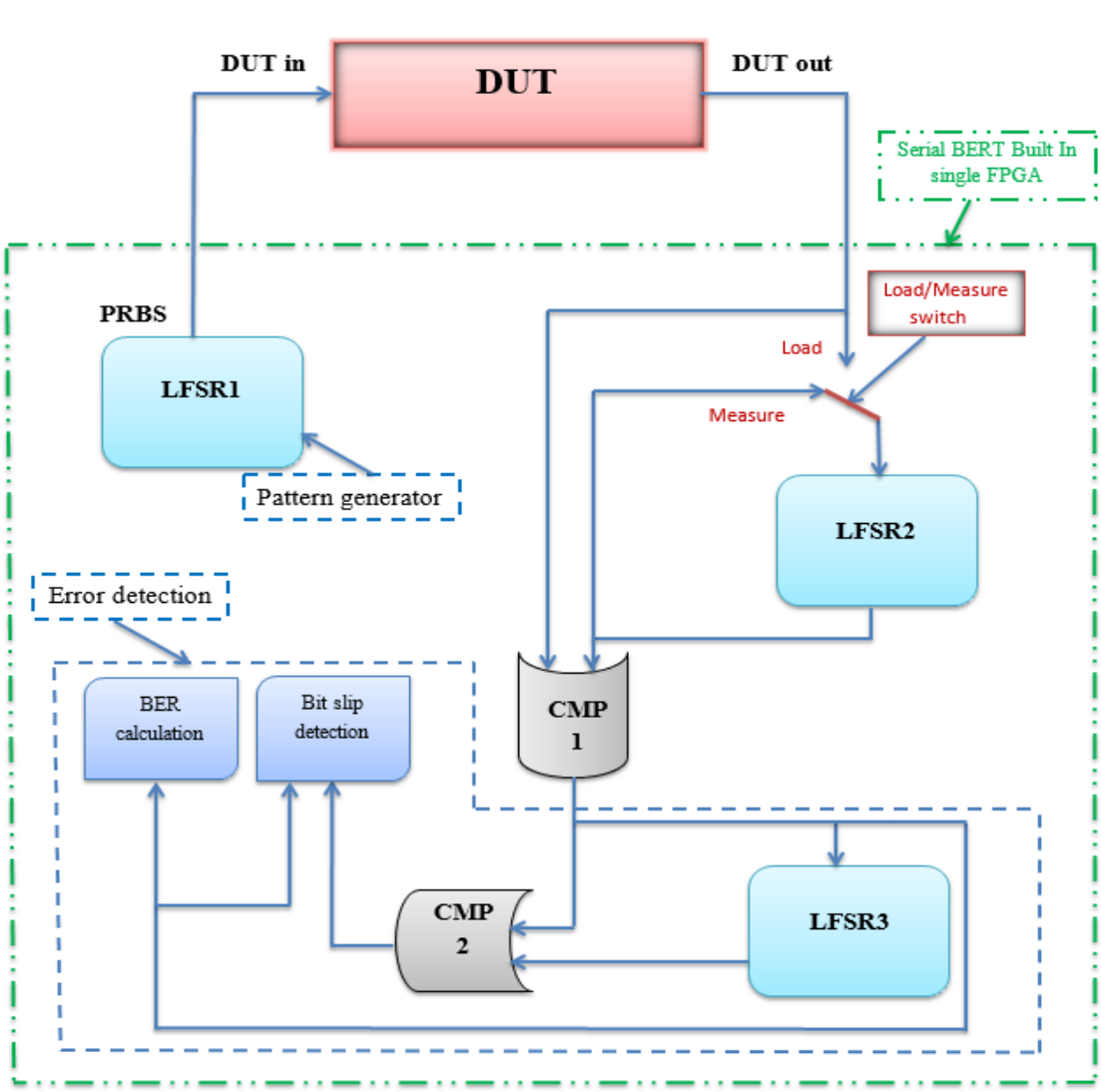

Figure 3: The structure of serial BERT (BER tester)

In figure 3, The LFSR1 generates a pseudo random bit sequence (PRBS), and the sequence is sent to the DUT. Before a measurement begins, the load/measure switch is set to be in load state. When BER measurement begins, the switch is changed to measure state. LFSR2, the switch and the comparator CMP1 are used for synchronization by replicating a delayed PRBS as the reference pattern. The comparator CMP2, comparing the pattern from the DUT with the reference pattern.

The BERT core design adressed by three issues:

1. test sequence generation
2. synchronization
3. bit error detection

1. test sequence generation

In BER measurement, a pseudorandom data sequence is used, as we can not create a truly random signal using deterministic methods.To simulate the actual measuring BER by the BERT, the pseudo-random sequence of bits (PRBS) is used to simulate the transmitted signal. Over an endless period of time, we can assume that the data transmission is a random process. Pseudo random sequence generator (PRSG) is used in the BER measurements. From the design scheme in figure 3, it is clear that we use Linear Feedback Shift Register (LFSR), where LFSR1 generate PRBS as shown in figure 4, and this sequence sent to the DUT [5]. Before starting the process of measurement, the (load / measure) switch has been set on the loading case, and when the process of measurement start, the status of the switch is change to measure case.

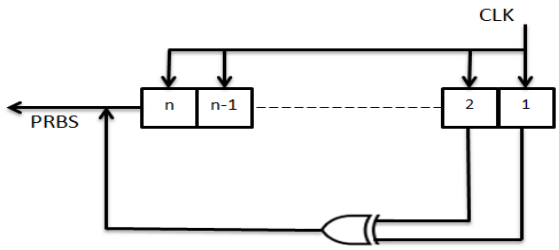

Figure 4: The Circuit for PRBS Generation by using LFSR

The sequence repeats periodically after a certain number of bits. In figure 4, n is the stage number of the shift register, determines the length of the PRBS. The maximum period of the sequence is 2^n -1.

2. synchronization

LFSR2 and the switch are used for the synchronization process by repeating the delayed PRBS as a reference pattren.The synchronization between the transmitted data pattrens and reference pattren is achived by loading LFSR2 and LFSR3 by transmitted PRBS before the switch turns from loading case to measurement case. For the purpose of explaining the work of serial BERT we will assume that n = 3 (LFSR length) and suppose that the DUT is an adjustable shift register, which gives outputs and delays which are controlled by a 2-bit signal as shown in figure 5, and called error slip signal (err-slip) where there will be four possibilities as shown in table 1.

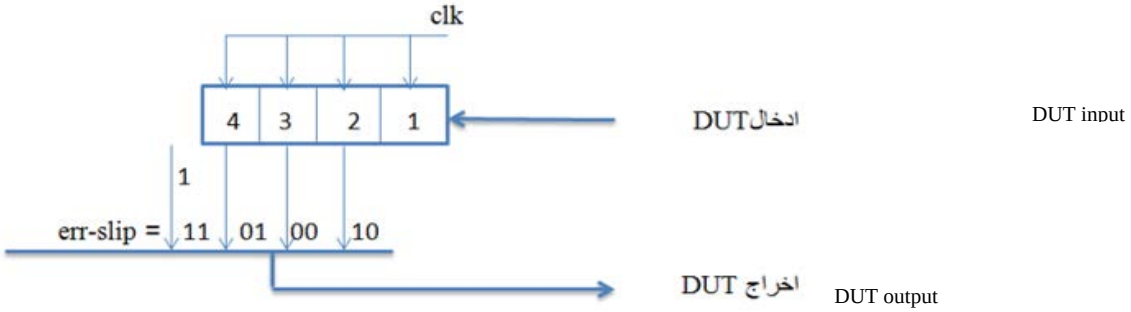

Figure 5: error slip signal

The four states are added to the DUT to demonstrate the bit slip detection function:

when err-slip = "00", the DUT exhibits a delay of 3 clock cycles, so it emulates the normal state;

when err-slip = “10”, the DUT exhibits a delay of 2 clock cycles, so it emulates a bit loss;

when err-slip = "01", the DUT exhibits a delay of 4 clock cycles, so it emulates a bit repeat;

and when err-slip = "11" , the output of the DUT is always set to be 1s, so it emulates an error burst.

Table 1: bit slip detection function

err-slip	DUT state	DUTout
"00"	Normal	3 clock cycle delay
"01"	bit repeat	4 clock cycle delay
"10"	bit loss	2 clock cycle delay
"11"	error burst	always 1

In this part we will assume the normal state where the adjusted signal err-slip in order to be in a state of probability "00" meaning that DUT output is the input signal delayed by 3 clock cycle. In this case, the synchronization circuit shown in figure 6.

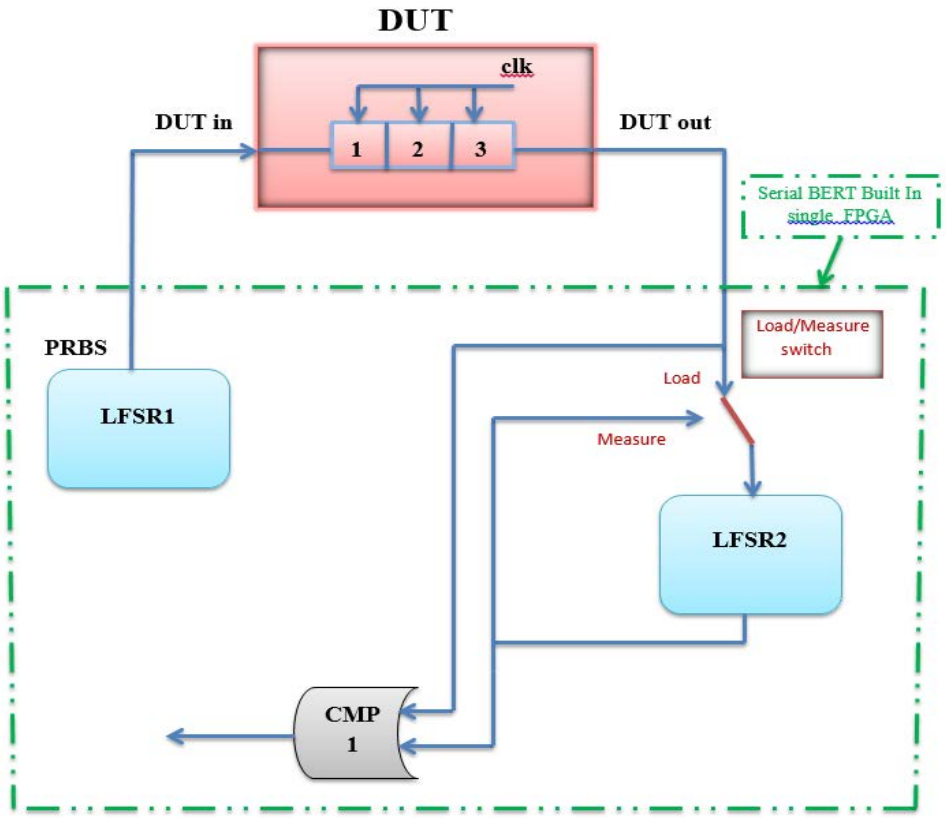

Figure 6: Synchronization circuit (n=3, DUT delay=3)

At the beginning of BER measurements process, the load / measure switch will be in the case of load, and with the assumption that the initial state of shift registers is not "000", the sequence send to the DUT (DUT in) will be repeat every 7 clock cycles (2^3- 1 = 7) after the confirmation signal reset.

According to the figure 6 and to achieve Primitive polynomial, each bit of the sequence DUTin is the XOR operation of the two bits that are immediately two bits before the bit. If the DUT in symbolized:

X1 X2 X3 X4 X5 X6 X7 X1 X2 X3 X4 X5 X6 X7 X1,…….

Then we would have:

X4 = X1 ⊕ X2, X5 = X2 ⊕ X3, X6 = X3 ⊕ X4, X7 = X4 ⊕ X5, X1 = X5 ⊕ X6, X2 = X6 ⊕ X7,

X3 = X7 ⊕ X1,…….

Table 2 shows the outputs of the circuit in figure 6 for the first 16 clock cycles after the reset signal is asserted.

Table 2: The Outputs of the first 16 clock cycles in figure 6

Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DUT in	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_1	X_2
DUT out	--	--	--	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_1	X_2	X_3	X_4	X_5	X_6
LFSR2out	--	--	--	--	--	--	X_4	X_5	X_6	X_7	X_1	X_2	X_3	X_4	X_5	X_6
CMP1	--	--	--	--	--	--	0	0	0	0	0	0	0	0	0	0

As can be seen from the table, that after a certain number of the clock cycles: LFSR2out = DUTout, therefore CMP1 = 0, where: DUT out ⊕ LFSR2out = CMP1.This particular number of clock cycles is the sum of delay of DUT and n, and here we have n = 3, and also delay of DUT = 3, then the number of clock cycles is 6.

In figure 6, when the LFSR2 is fully loaded with the transmitted PRBS, the LFSR2out equals to DUTout, and so the switch can be changed to measure state without affecting the work state of the whole circuit. In this case, LFSR1 and LFSR2, the two LFSRs generate the same PRBS, but the sequence from LFSR2 is d clock cycles later in timing than that from LFSR1 , where d is the delay of the DUT in terms of the number of clock cycles. Therefore, if the test patterns from LFSR1 are correctly transmitted by the DUT, then the two inputs of CMP1 should be the same value in each clock cycle. If a ‘1’ is detected, a transmission error is counted; otherwise, the transmission is error-free.

3. bit error detection

Error Detection (ED) is the scrutiny of binary data as they are sent to detect the error if it happened, as in the example in figure 7, and when the error is discovered , the special procedure is usually taken in a re-send binary data or an error message is displayed.

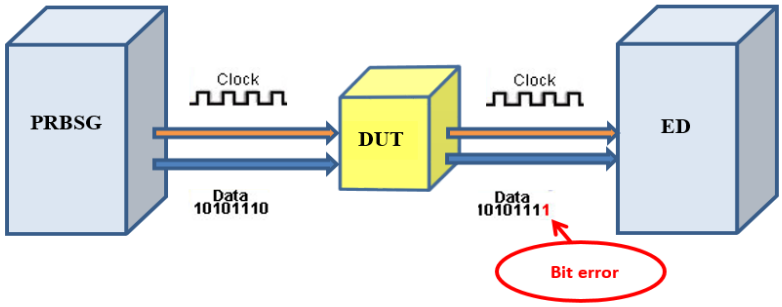

Figure 7: example of error bit happened when sending data

In real communications systems it is possible coincidence errors during the transfer process, and errors as possible be: single errors, bursts error, or bit slips (both the bit loss and the bit repeat are called bit slip). After the synchronization process, both error bursts and bit slips can result in a large number of errors. For errors resulting from error bursts, they should be counted in BER calculation. For errors resulting from bit slip, they should not be totally counted in BER calculation; only the bits lost or repeated should be counted as errors. When bit slips happen, the number of bit errors measured will be infinite due to the phase shift between the received and reference patterns. In this case, the measurement must be interrupted and the BERT must be resynchronized. The bit slip detection circuit is shown in figure 8.

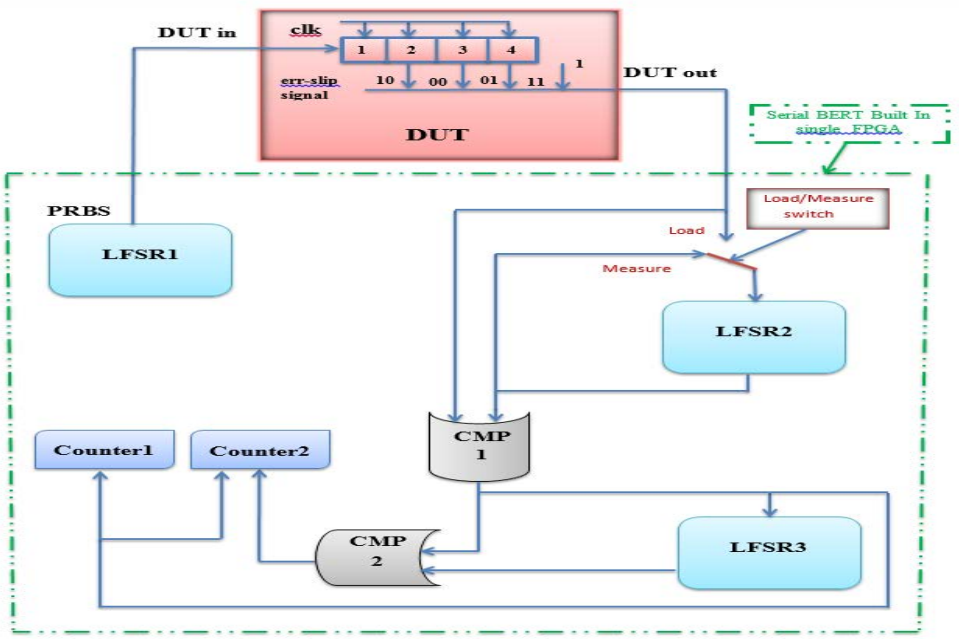

Figure 8: Circuit for Bit Slip Detection

When a large number of errors are encountered, the BERT must be able to determine whether the errors are caused by bit slips or error bursts. If bit slips happen, the BERT should be interrupted and resynchronized; otherwise, the measurement should continue. After the synchronization process, the BERT works in measure state. As discussed before, if all bits are correctly transmitted, the output from LFSR2 and DUTout are the same PRBS in terms of both the value and the phase. Therefore, the outputs from CMP1 and CMP2 are all zero. If a bit slip happens, the two patterns from LFSR2 and the DUT are shifted in phase. To illustrate this case, we assume that in measure state err-slip is switched from "00" to "10" in clock cycle t, which emulates a bit loss.From clock cycle t on, the output sequence of LFSR2 is set to be X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_1 X_2 X_3 X_4 X_5 X_6 X_7 …… .The output of DUT and CMP1 are listed in Table 3.

Table 3: the output of DUT and CMP1

Cycle	t-2	t-1	t	t+1	t+2	t+3	t+4	t+5	t+6	.....
LFSR2	X_6	X_7	X_1	X_2	X_3	X_4	X_5	X_6	X_7	.....
DUTout	X_6	X_7	X_2	X_3	X_4	X_5	X_6	X_7	X_1	.....
CMP1	0	0	X_4	X_5	X_6	X_7	X_1	X_2	X_3	.....

As can be seen from Table 3, when a bit slip happens, the output of CMP1 is the same PRBS value as it transmitted PRBS value, but may have different phase. The above is for the situation that only one bit is lost. It is easy to understand that when more bits are lost or bit repeat happens, the output of CMP1 is always the PRBS value, but with different phase. According to the synchronization principle as discussed before, when DUTout is a PRBS and the switch is in load state (figure8), the pattern from the output of LFSR2 is the same as DUTout after n clock cycles, which results in zeros from the output of CMP1. Applying this principle, we know that, if the output of CMP1 is a PRBS, the two sequences from the outputs of CMP1 and LFSR3 are the same after n clock cycles, where n is the length of the shift registers. Therefore, the output of CMP2 is all zeros. When error bursts happen, as the real bit errors are random and the synchronization is still maintained, the output of CMP1 is random. The output of CMP1 cannot be in phase with the output of LFSR3, so the output of CMP2 is not all zeros. The differentiate between bit slips and error bursts is done by adding two counters, counter1 and counter2, as shown in figure 8.

Counter1 counts burst error or normal error and counter2 counts bit slip, where they are linked to the output of CMP1 and CMP2. When measurements taking place, if the difference between the two counters is zero or very little difference , the error is burst error or normal error, either If the difference between them make a big difference, the error is a bit slip. Of course, in the absence of an error, the counters originally not counted any number.

1.2.2. Parallel BER tester design

The design of the parallel BERT is based on the serial BERT presented in previous section. Basically, a k-bit parallel BERT can be built using k independent serial BERTs that have the same load time, where k is the width

of the parallel data (bit0~bit (k-1)). The parallel BERT sends pseudo-random word sequences (PRWSs) to the DUT. The parallel BERT is shown in figure 9.

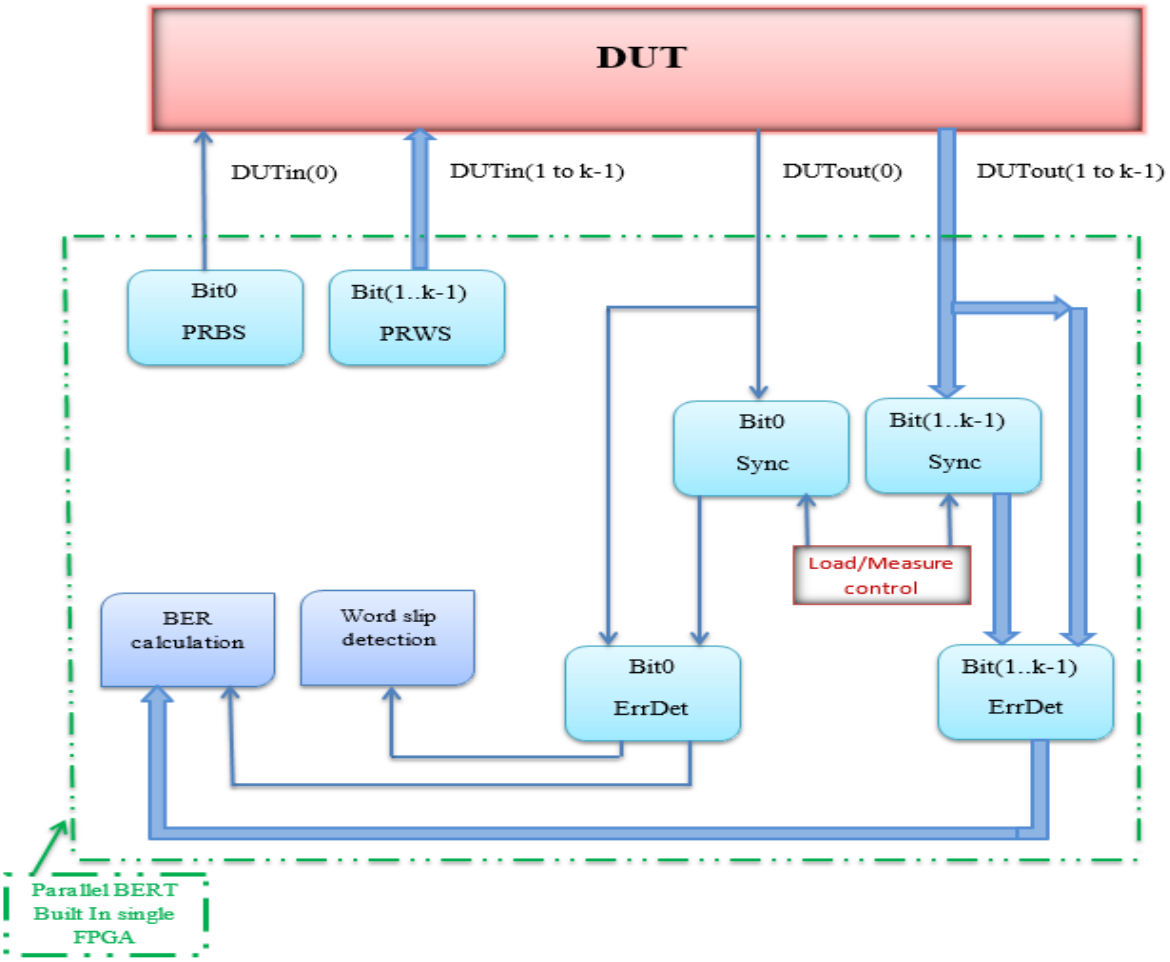

Figure 9: The structure of parallel BERT

In order to qualify randomness of the generated sequences, the independence of each of the serial BERTs is very important. That means the length of the shift registers in each of serial BERTs should be different. In the design, the width of the parallel BERT is parameterized, which ranges from 1 to 10. It can also be easily expanded to a width more than 10. For all the lengths of the shift registers in the serial BERTs, being prime to each other can achieve a maximum period of the PRWS, 2^m-1, where m is the sum of all the lengths. When k independent serial BERTs are directly put together to build a parallel BERT, each of the serial BERTs has circuits for the load/measure switch control and bit slip detection. The circuits for the load/measure switch control of each bit of the parallel data should change load/measure state at the same time, and the parallel BERT should be capable of distinguishing between error bursts and word slips instead of bit slips in a serial BERT. Therefore, only one of the k such control circuits is needed for switch control and word slip detection. From the figure 9 note that the design of the Serial BERT for bit0 circuit are used to control all switches of loading and measurement in synchronization circuits and word slip detection circuits. Also note that parts: Bit0 PRBS, Bit0 Sync, Bit0 ErrDet, load / measure control and word slip detection impose the same Serial BERT circuit in figure 3. The

parts: Bit (1 ... k-1) PRWS, Bit (1 ... k-1) Sync and Bit (1 ... k-1) ErrDet are a set of k-1 independent units. One of these units is shown in figure 10.

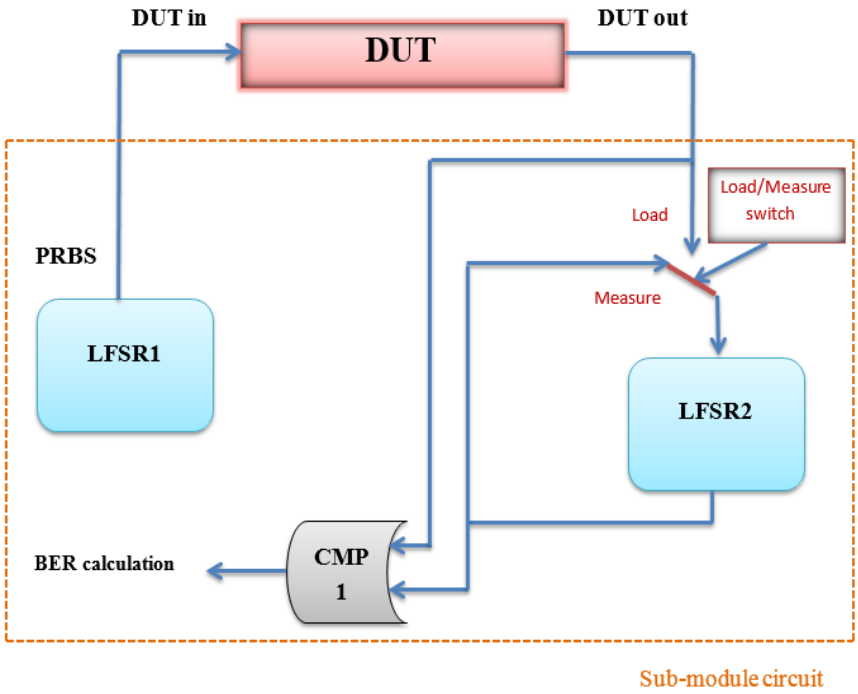

Figure 10: The structure of the basic circuit used in parallel BERT

Since each circuit has the same design but a length of each shift register different from the other, so as to keep randomness of the word sequence. The design is the same as in serial BERT, as LFSR1 generate PRBS, while LFSR2 and switch constitute one bit synchronization circuit and CMP1 constitute a 1- bit error detector circuit.

2. Simulation of BERT design

The design of BERT is done by using MATLAB Simulink which is an ideal tool for simulate digital communication systems [6], and then install the software on-chip FPGA type Spartan-6 using ISE 14.2 software. The results showed a match in both the programmatic representation on the computer and when implemented on the FPGA chip.

2.1. The use of Xilinx System Generator in the simulation

Xilinx Company provided tools to the possibility of designing and implementing the system fully, the most important Xilinx System Generator, which provides a set of Simulink blocks (models) in the software environment, so as to implement several processes in the hardware environment in which it is possible to represent on different types of Xilinx FPGAs. System Generator reduces the time it takes to simulate the design, on the other hand, have a flexible design and can change the design parameters and examine the impact of the change on the performance of the system in a fast manner [7].

2.1.1. Simulation of serial BERT design

The final serial BERT simulation design is shown in figure 11:

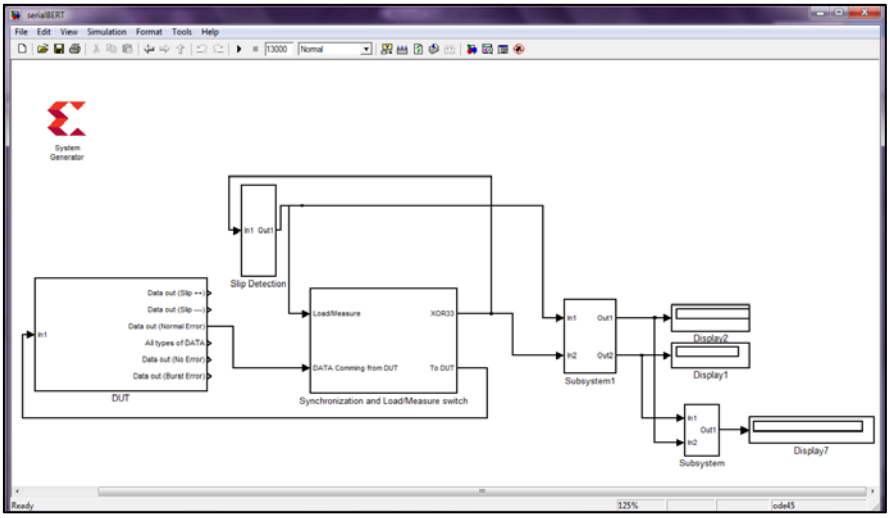

Figure 11: serial BERT simulation design

2.1.2. Simulation of parallel BERT design

The final simulation design of an example of 3 bit-parallel BERT (k=3) is shown in figure 12:

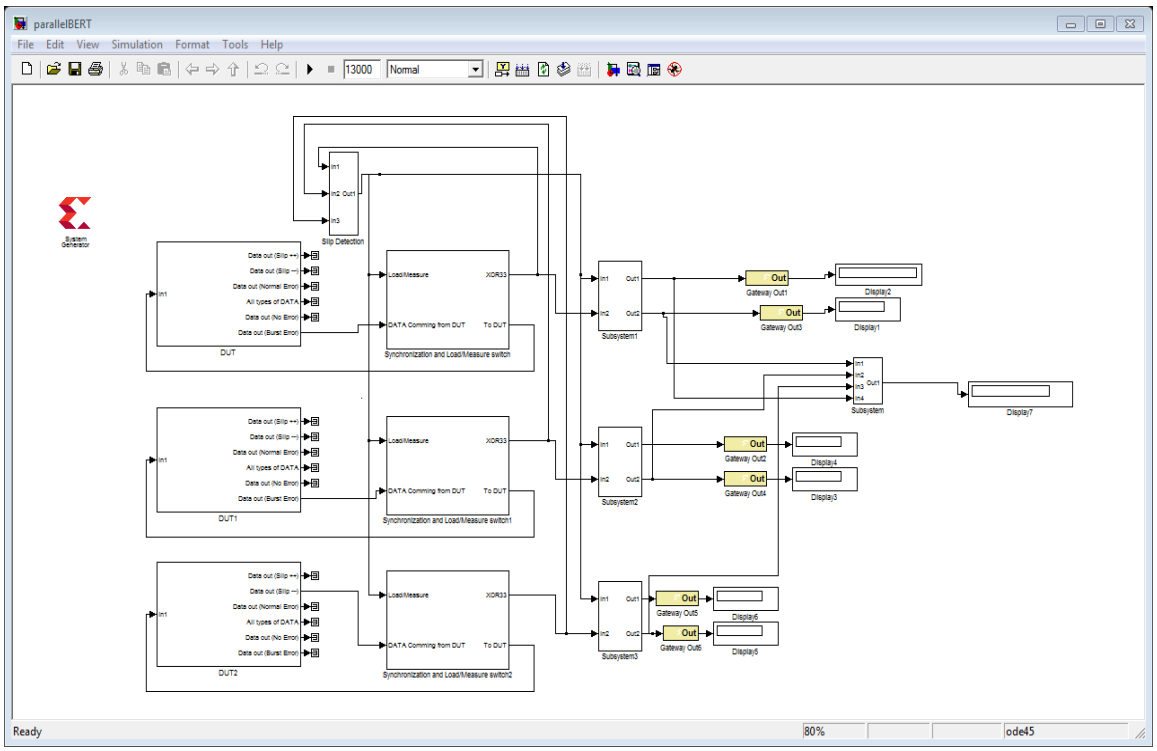

Figure 12: 3-bit parallel BERT simulation design

3. Simulation results and Emulation

3.1. Simulation results

After designing models using Matlab Simulink and for the purpose of implementation of any model and then open the model window, the number of bits used in the implementation process is set and then implement the simulator model of serial BERT as shown in figure 13. Where the result of implementing the simulation shows the BER and the total number of bits and the error bits. Also the implementation of the simulator model of 3-bit parallel BERT as shown in figure 14.

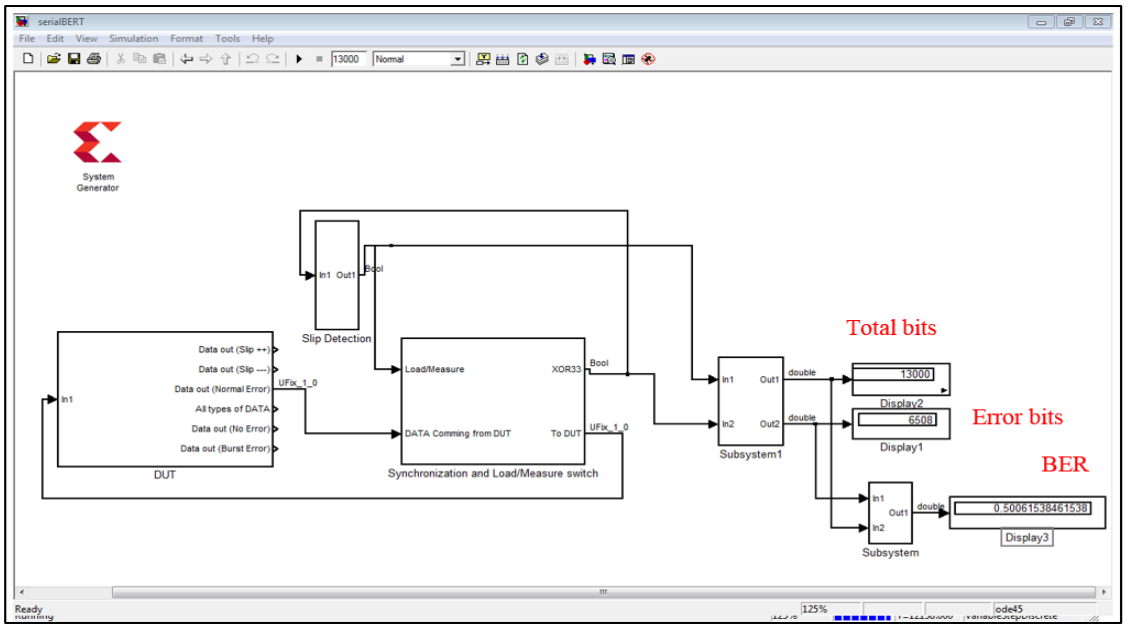

Figure 13: implement serial BERT simulation

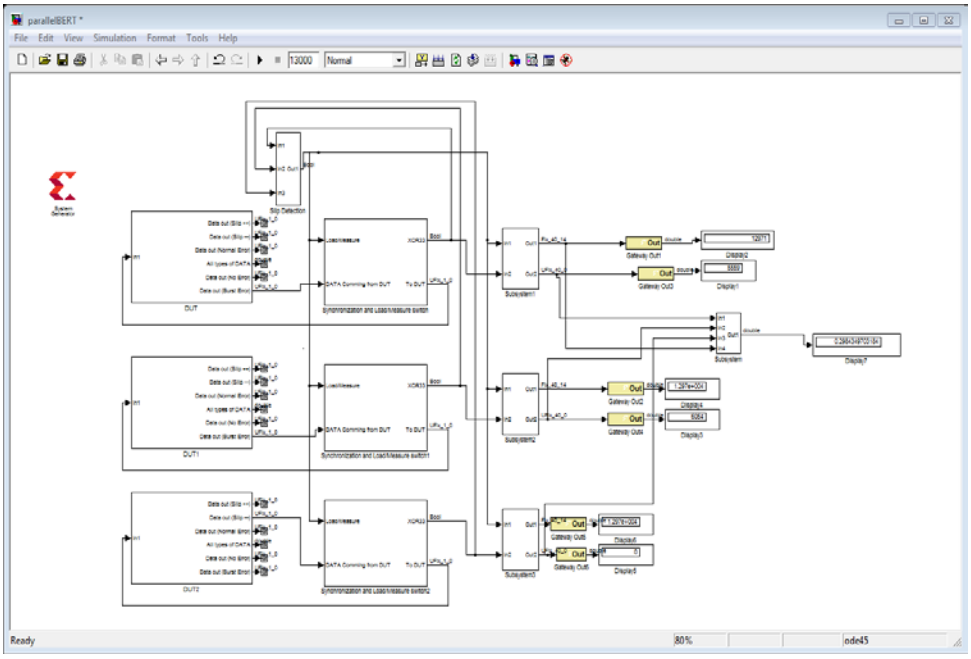

Figure 14: implement 3-bit parallel BERT simulation

3.2. Emulation

FPGA chip comes with different types, but all almost share the same configuration. Figure 15 shows the type of FPGA chip of the type Spartan- 6 through which carried out the models designed [8]. The results were identical to the results of simulations in both types, the serial BERT and parallel BERT. The figures 16, 17 shows the emulation of the serial BERT and parallel BERT simulation designs on FPGA chip respectively.

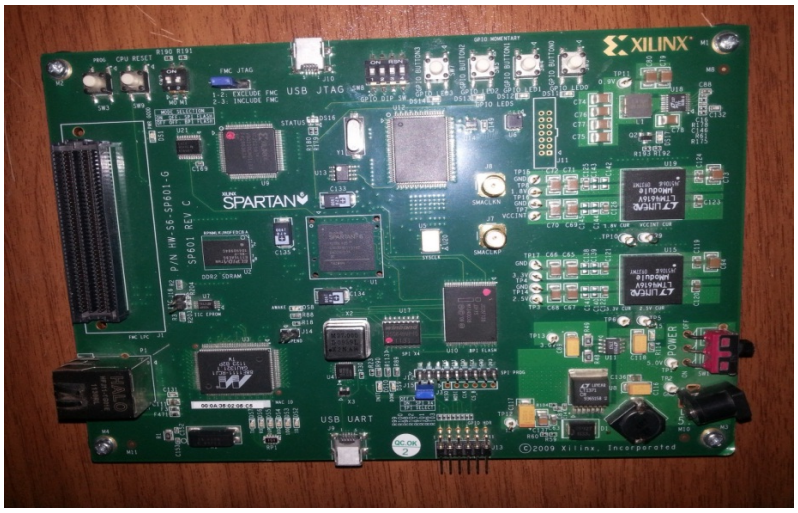

Figure 15: FPGA chip of the type Spartan- 6

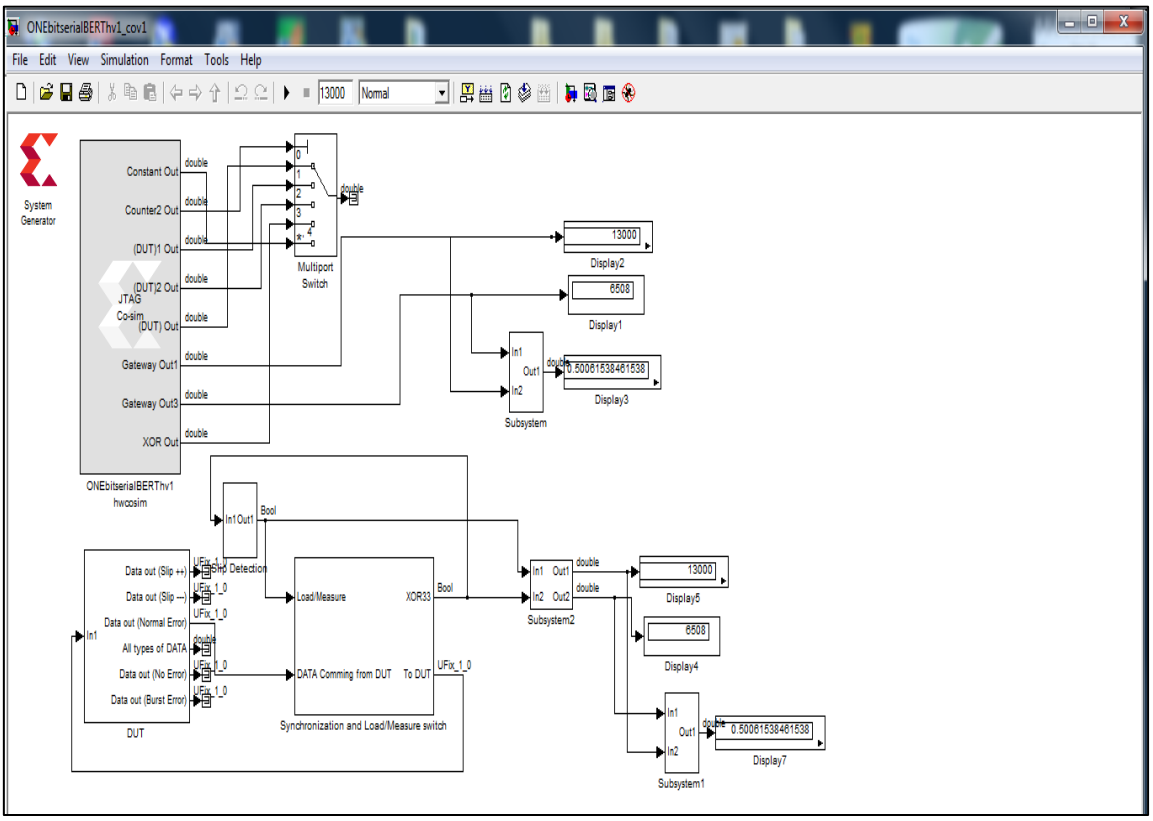

Figure 16: Implementation of serial BERT simulation on FPGA chip

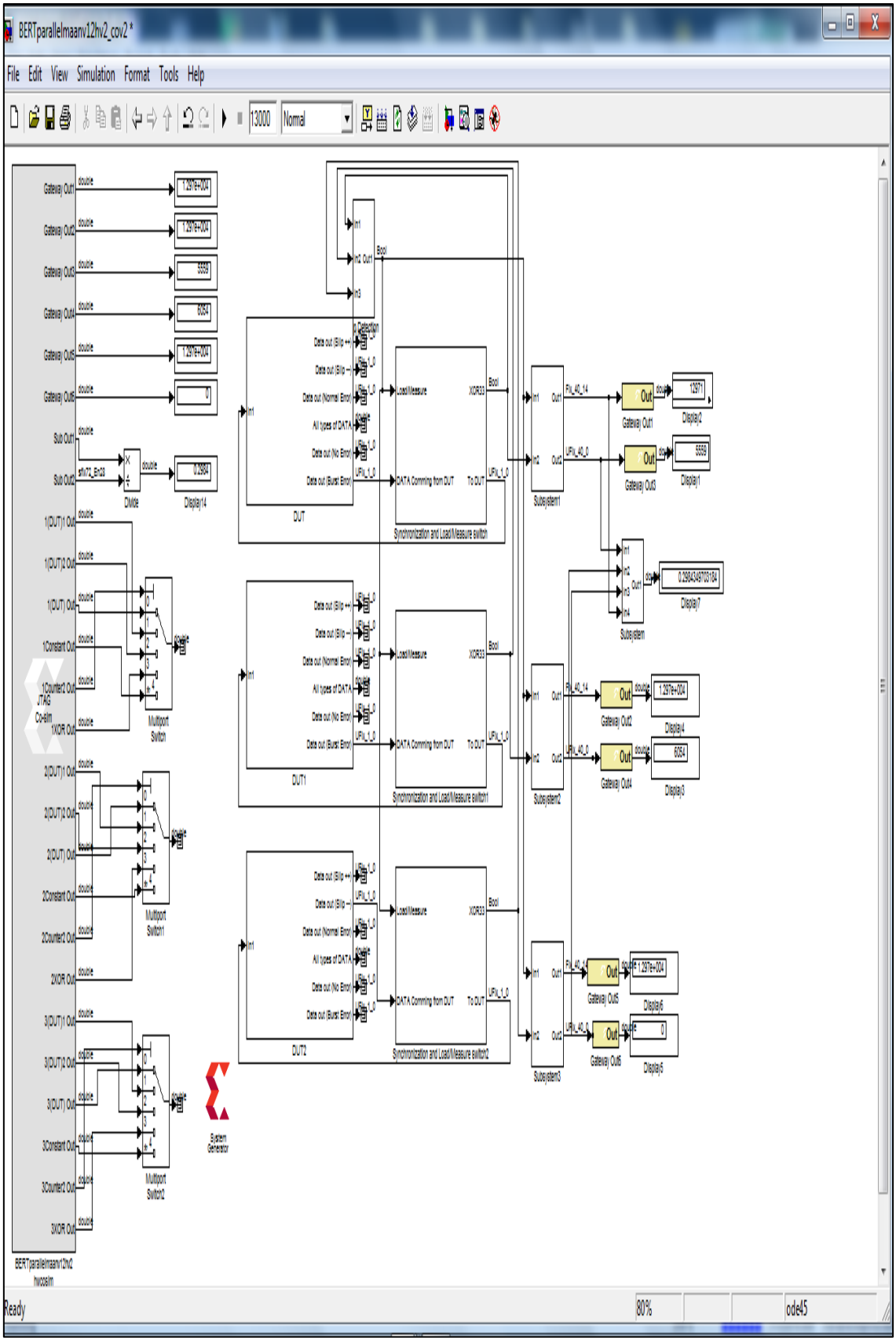

Figure 17: Implementation of 3-bit parallel BERT simulation on FPGA chip

4. Conclusion and Recommendation

We have provided in this thesis BERT scheme depending on the FPGA chip , the proposed scheme as possible be used to measure the BER performance of a wide range of digital communications systems. Compared with the traditional simulation software, the BERT proposed scheme is somewhat faster. Compared with the traditional BERT rate, the proposed solution is much less expensive, as well, this design makes it easy to examine the different types of DUTs.

The examination of Jitter is a challenge issue and its importance began to widen with the increasing of transmission speed and to achieve this, the Additive White Gaussian Noise (AWGN) core is added to the design of BERT which together constitute a good starting point for Jitter testing research, with the continuing enhancement of FPGA performance, it is possible to build a jitter testing scheme in a single FPGA.

References

[1].K.B. Schaub, J. Kelly, “Production Testing of RF and SoC Devices for Wireless Communications”, Artech House Inc., 2004**.**

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

[3] .Y. Fan, Z. Zilic, Accelerating Test, Validation and Debug of High Speed Serial Interfaces , DOI 10.1007/978-90-481-9398-1_7, © Springer Science +Business Media B.V. 2011.

[4]. A. Deshpande, M. Deshpande, D. Kayatanavar," FPGA Implementation of AES Encryption and Decryption", Control, Automation, Communication and Energy Conservation, 2009. INCACEC 2009. 2009 International Conference on. IEEE, 2009.

[5]. "Efficient Shift Registers, LFSR Counters, and Long Pseudo Random Sequence Generators ", URL: http://www.xilinx.Com/bvdocs/appnotes/xapp052, XAPP 052 July 7, 1996 (Version 1.1)

[6]. the Math Works, Inc. Natick, Massachusetts. http://www.mathworks.com

[7]. Xilinx Inc., "System Generator for DSP", UG640 (v14.1), 424pp. April 24, 2012

[8]."Spartan-6 FPGA Configurable Logic Block", URL: www.xilinx.com, UG384 (v1.1) February 23, 2010.