An Open Source, FPGA-Based Bit Error Ratio Tester for Serial Communications

M. J. Basford, A. E. Pena-Quintal, S. Greedy, M. Sumner, D. W. P. Thomas
George Green Institute for Electromagnetic Research
University of Nottingham, NG7 2RD
Nottingham, UK
eexmjbas@nottingham.ac.uk

***Abstract*—This paper discusses the design of a FPGA-based Bit Error Ratio (BER) measurement system in a serial communication link under electromagnetic emissions. The BER measurement is an important figure for the reliability of digital communication systems. Generally, the equipment involved to measure BER is prohibitively expensive. This paper demonstrates the design, construction and verification of an Open Source System (OSS), with all design files being made freely accessible to the public for use in general or highly specialized BER measurements. The methods used for programming the Hardware Description Language (HDL) modules were done as simple as possible to facilitate easy integration into experiments for academics with limited experience in HDL usage.**

***Index Terms*—FPGA, BER, electromagnetic emissions, OSS, OSH, serial communication.**

I. Introduction and Background

Electromagnetic Compatibility (EMC) is a field that attracts increasing attention in research, as the consequence of a failure of any of the increasing number of data systems in everyday life may result in ever-increasing consequences. It is therefore imperative that researchers may have access to equipment that will allow them to determine how resilient a system is to failure, or interference, in the presence of electromagnetic noise. There are two key approaches towards understanding the resilience of a system. The first is to measure the change of induced voltages and currents on a wired network and comparing the responses to that of current standards such as [1] and [2]. The second approach is to determine how the data itself is affected by sources of interference by finding the ratio of correctly transmitted bits, or BER, to incorrectly transmitted bits. Examples of standards which use this approach include [3] and [4], which are designed for use in the telecommunication industry, primarily for mobile networks.

A Bit Error Ratio Tester (BERT) is used to calculate this ratio. Unlike many other types of measurement, examining the BER permits an understanding of the full end to end performance of a communication system including the transmitter and the receiver, all within a single figure [5]. However, the cost for equipment which is capable of performing measurements of this error ratio presents a large barrier for some in the academic community as prices for such systems can easily reach in excess of £90,000 [6] for a basic system. Furthermore, many commercially available BERTs have very high bandwidth capabilities, with even the most basic systems being capable of measuring data rates in excess of hundreds of Megabits per second (Mbps) [7]. Such high data rates are not necessarily required by every academic, such as the research presented in [8].

As part of the investigation into the BER phenomena observed in [8], the authors needed to perform the BER measurements using a randomized stream of bits. However, the aforementioned cost of procuring a BERT prevented this measurement from being taken. It was therefore decided to develop a simplified BERT which would allow for BER measurements to be taken, whilst keeping the costs of the instrument to a minimum. In the spirit of facilitating ready access to such instrumentation for other academics and persons at large, it was also decided that all documentation including circuit schematics, HDL, software and Printed Circuit Board (PCB) files should all be made Open Source (OS), without any restrictive licenses. The design and operation of the BERT (known as OS BERT) as it is today is presented in this paper demonstrating the flexibility of the system. All of this is achieved with a budget of less than £150.

II. Design of the BERT

Fundamentally, a BERT must find the ratio of incorrectly transmitted bits to transmitted bits, as given by equation 1.

$$BER = \frac{N_{err}}{N_{bits}} \tag{1}$$

where N_{err} is the number of incorrect bits whilst N_{bits} is the total number of bits sent in a time period. When data is transmitted over a data link, there is a possibility of bit errors being introduced into the system. If this occurs the system integrity may be compromised [9] with potentially severe repercussions. The likelihood of a given bit being incorrectly determined by a receiver, and therefore the value of the BER, is affected by many factors including the signal to noise ratio of the signal, distortion, and jitter [10].

At the heart of OS BERT is an Intel FPGA Cyclone IV. The use of an FPGA permits the parallelisation of processes within the same device, ensuring that absolute-time dependant operations may happen repeatably and without unexpected interruptions, as might be expected from the use of a microcontroller. The Quartus Prime Integrated Development

Environment from Intel is selected to create the programming files for the FPGA. The Lite edition of this software is freely available for download, at time of writing [11]. This software gives the user the possibility to create HDL projects by using the many included, free-to-use tools and IP blocks, such as: clocks, memories, Phase Lock Loop (PLLs), communication interfaces, I/O interfaces as well as digital filters. Quartus Prime also includes the NIOS II soft-CPU IP files, as well as an integrated form of the Eclipse-based programming environment to program C code for use on the created processor. The combination of these two platforms can generate a reliable hardware and software platform, which allows for the high-speed, precise timing, and parallelisation capabilities of the FPGA hardware along with the ease of programming sequentially in the C programming language on a CPU. While using system level tools can eliminate the need for extensive hardware knowledge and will usually shorten the design time, it is important to note that a simulation library may include only a set of basic digital communication components and may not include modules, such as new coding algorithms, for emerging technologies [12]. If proper timing analysis is not performed on a given design, then the device may not work as expected under all operating conditions. OS BERT makes use of only widely available modules from Intel, as well as those created by the authors which have undergone timing analysis.

III. HDL Modules

Figure 1 shows a high-level block diagram of OS BERT, split into the major HDL modules. A NIOS II soft-CPU was utilised to handle the communication with a PC through a serial interface, as well as control the non-critical timing functions of the system such as resetting data buffers. The fundamental BERT processes of random bit generation, transmission and comparison are handled within the module marked BERT. To permit the adjustment of data throughput, the BERT module is fed a clock sourced from a frequency divider, controlled by the soft-CPU. This is itself fed by the Phase-Locked-Loop (PLL) module which takes the 48 MHz input clock and multiplies it to 320 MHz output clock allowing for the generation of data at speed far exceeding that which a purely CPU-based system would allow.

The sub-modules which form the CPU block in Figure 1 are presented in 2. At the heart of the design is the NIOS II soft-processor module. A UART module is used to communicate with a user's PC. At the time of writing, this set to a baud rate of 9600bps as this connection does not require a high throughput. The CPU module in the current version comes with 16 kB of on-chip RAM, in order to simplify the construction processes of the BERT and to reduce the cost of including extra external memory. This is more than sufficient for the operation of OS BERT, though users of OS BERT who develop their own applications may wish to keep this memory limitation in mind.

Note that the design of OS BERT deliberately refrains from using a general or multi purpose data/control Parallel

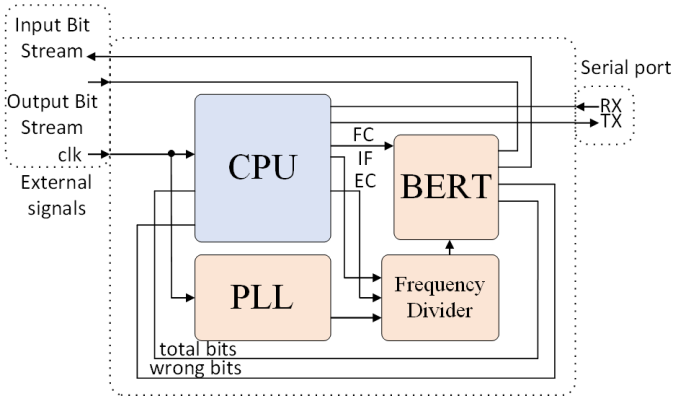

Figure 1. OS BERT block diagram, showing the primary connections between the authors' HDL modules (orange) and soft CPU (blue). FC: Frequency Control, IF: Input Frequency and EC: External Connection.

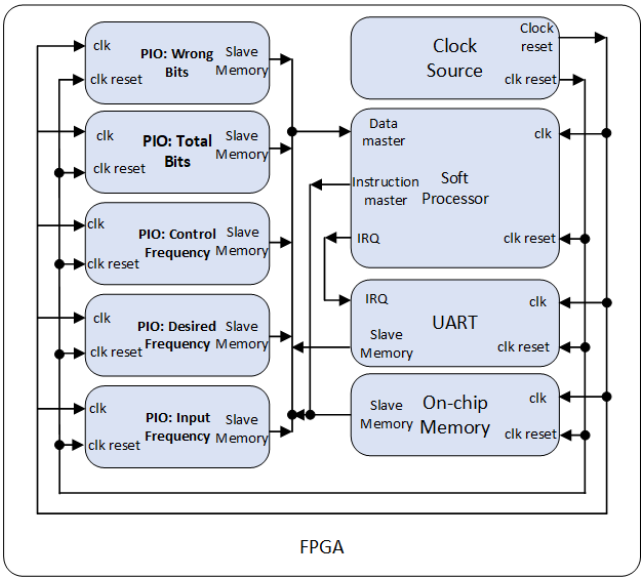

Figure 2. Structure of the Soft CPU in the BERT

Input/Outputs (PIO). This not only helps to simplify the implementation of the software within the CPU as each important parameter can be directly accessed by the CPU with no abstraction, but it also permits the direct implementation of generalised HDL modules with little to no modification required for their inclusion.

Figure 3 shows the key components of the BERT module from Figure 1. In short, a clock signal is used by a 31-bit Linear-Feedback Shift Register (LSFR) to generate a pseudo-random bit on each clock cycle. The connections of the LSFR are based on tables published in XCELL and later Xilinx Data Books, but can be more readily sourced from [13]. The clocked data stream is then split into two paths. One acts as a victim channel, in other words it will be the data channel subject to interference and is therefore routed out of the FPGA. The returning data from this victim channel is then fed into an XOR gate for comparison with the unaffected data stream. The exact point of digital state determination is governed by clocking

the incrementing Incorrect Bits and Total Bits. By inverting the clock signal (thereby offsetting the rising edges by 180°), shared with the LSFR, the sample will be taken during the exactly midpoint of each data bit. This was suitable for the purposes in the author's experiments, but for experiments which require alternative positioning of the sampling point, section VI offers some alternative implementations.

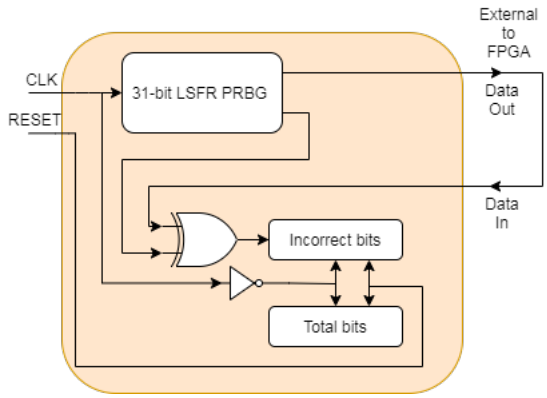

Figure 3. Structure of the Soft CPU in the BERT

Before a measurement, the soft-CPU will reset the values in the both the incorrect bits and total bits registers before enabling the frequency divider output. As each bit is received, the total bits register is incremented and an X-OR operation applied with the received data bit and the known-good bit. The output of the operation is then summed with the contents of the incorrect bits. When the CPU is ready to stop the counting, the clock signal is disabled and the two registers are read from their PIO connections, shown in Figure 2. A maximum measurement duration is not necessarily set by any time limitation on OS BERT, but rather by the size of the Total bits register. As a 32-bit register, this limits the maximum number of bits to just over 2.14×10^9. Depending on the data rate, this may facilitate measurement periods of up to 43 seconds at 50 Mbps, or 2147 seconds at 1 Mbps.

IV. Hardware Design

OS BERT also includes a fully designed PCB, with all required components for operation. Three fully independent data channels are accommodated for the version 1.0 PCB, with separate transmit and receive ports for each. Each transmit channel utilises a 600 MHz bandwidth TTL buffered driver, a PO49FCT32805, to ensure that data bits from the FPGA are reliably matched to a $50\,\Omega$ transmission line impedance. The buffer driver also boasts a less than 2.6 ns propagation delay, resulting in a negligible effect on the sampling timing of the FPGA receiver [14]. These factors combined permits the BERT to achieve data rates in excess of 75 Mbps whilst maintaining reasonable bit shapes, demonstrated in section V.

Since the design and manufacture of OS BERT version 1.0, a commercially available Anritsu MP8302A BERT was reverse engineered and was found to have a near identical analogue front-end topology, albeit with a different comparator and support components for additional functionality. The termination impedance may be set at this analogue front-end by the user, but as of version 1.0 of the OS BERT PCB, this can only be done through hardware modification. However, a number of termination options are incorporated onto the PCB allowing users to chose from a variety of terminations such as single-ended or split terminations. Figure 4 shows a simplified representation of both the transmitter and receiver front ends. Note that the receiver circuit shows a threshold voltage set for half of the supply voltage through the use of two $10\,\mathrm{k}\Omega$ resistors, which can be changed to suit a user's own particular requirements.

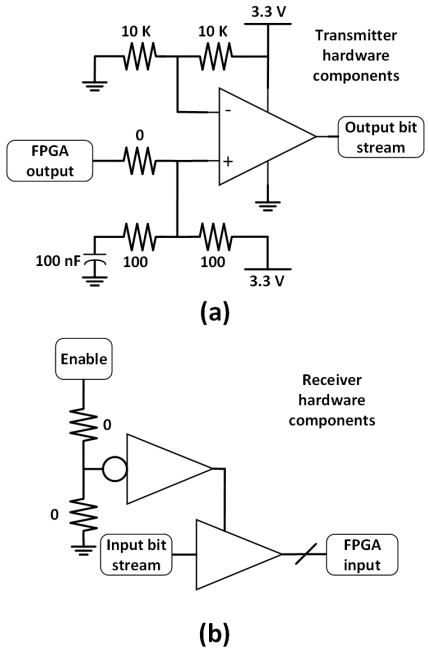

Figure 4. OS BERT hardware for data receiver (a) and transmitter (b). Note that the $0\,\Omega$ resistors on (b) are exclusive - only one placed.

An RS232-compatible serial communications port is provided for the communication with a dumb terminal program hosted on a PC. UART from the soft-CPU core is converted to the voltages required by RS232 by a MAX3232 integrated circuit.

Communication with a sixteen pin General Purpose Input/Output (GPIO) is also provided to allow users to easily incorporate their own, low-speed modifications with ease. Ground, 3.3 V and 5 V connections are provided to facilitate a wide range of devices. In the case of the authors, this port was useful to add an additional communications port to compliment the RS232 serial communication port for PC control. Other alternative uses may include the addition of LCD screens for quick reference of measurements, or to interface with other instrumentation such as amplifiers or arbitrary function generators.

V. Typical Usage

A typical setup using a version 1.0 PCB consists of the OS BERT PCB connected to all required inputs and outputs is shown in Figure 5. This PCB includes the input and output data connections through SMA connectors, an RS232 link for serial communication to a PC, JTAG for programming the device, power, and GPIO. In this particular setup, only one data channel is completely populated with components, however as described in section IV the PCB for this BERT facilitates measurements with three independent channels. GPIO is also provided for users' own needs, but in this case was not required for this measurement procedure.

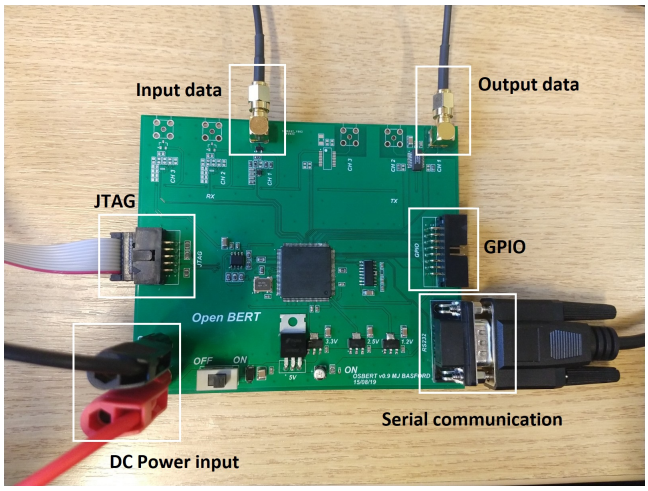

Figure 5. The BERT PCB, ready for a single channel measurement. Annotations highlight the primary hardware interfaces.

When programming OS BERT using the JTAG connection, OS BERT must be switched on first. OS BERT is rated for an input voltage of 7 V - 25 V DC, with a typical operating current of 400 mA with all three channels in operation. The BERT permits the direct upload of SRAM Object Files (.sof) through the JTAG, but also includes 4 Mb of Flash memory for JTAG Indirect Configuration (.jic) files. The use of the Flash memory module allows OS BERT to be used without the need for reprogramming after each power cycle.

Communication between a PC and OS BERT is carried out through serial communications interface, intended for use with a dumb terminal application. In the author's case, the freely available software called Termite [16] was found to be most reliable and easy to use. Alternatively, users may wish to interface OS BERT with their software programs to allow for easy automated data collection.

OS BERT utilises a set of simple, easy to understand instructions and replies. Table I shows a list of commands available under the version 1.0 release of OS BERT. The commands range from simple measurement start commands, to altering data rate and enabling specific outputs.

Experimental figures are returned through the same console, or can be processed by a user's program in the same way as any other serial input. An example of communication with OS BERT using the Termite terminal can be seen in Figure 6.

Table I
Open Bert Commands

Command	*Description*
start	Begins the bit counting and returns BER
freq	Changes the output frequency of the BERT data rate
duration	Sets the duration for the BERT measurement
outon	Turns on the output signal for the BERT
outoff	Turns off the output signal for the BERT
h	Displays the help menu
version	Returns the current version of the BERT software

```
UoN BERT v1_1
Startup...
Ready. For help send 'h'.

h
Valid command(s) v1_0:
'start' - begins counting and returns BER after set duration.
'freq' - sets frequency of the clock signal.
'duration' - sets duration of measurement in whole seconds.
'outoff' - turns off the output.
'outon' - turns on the output.
'version' - prints the current version of the BERT.
Ready. For help send 'h'.

freq
Enter frequency:
1000000
Ready. For help send 'h'.

outon
Output On.
Ready. For help send 'h'.

start
Resetting values - done!
Counting... Please wait....

Finished count.
Total bits:        13858035
Incorrect bits: 0
BER:
Please wait... DO NOT SEND COMMANDS

Ready. For help send 'h'.
```

Figure 6. OS BERT operation using the Termite terminal. Blue indicates commands from the PC, green indicates replies from OS BERT.

Fig. 7 shows the performance of OS BERT at two different data rates of 1 Mbps and 0.5 Mbps, using only the on-board hardware to emulate a constant stream of data bits at a 3.3 V TTL logic levels. Also shown is the output of the frequency divider module from Figure 1, clearly showing the data bits being generated on each rising edge of the divider output. The data plots show a very well constructed bit shape, with fast rising edges with a minimum of overshoot and ringing into a 50 Ω load.

It is difficult to perform a direct, quantitative comparison with commercial BERTs as each BERT is in effect measuring

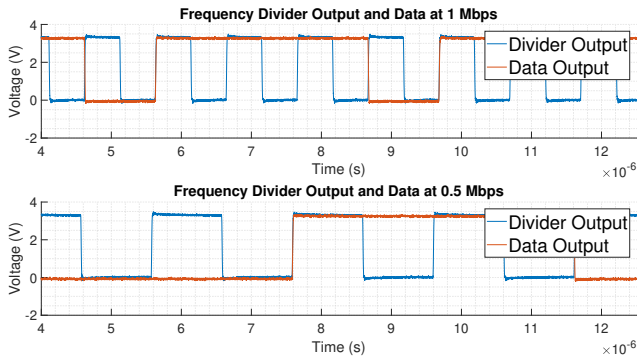

Figure 7. Frequency divider outputs (blue) show scaling depending on the selected frequency, whilst the randomised output data stream (orange) changes on each rising clock edge.

the response of its own analogue front-end to interference. As such, it is to be expected that different BERTs with different front-end components may produce a different BER. It is for this reason that this paper performs a qualitative comparison between OS BERT and a comparable Anritsu MP8302A BERT, shown in Table II.

Table II
COMPARISON OF OS BERT AND ANRITSU MP8302A

Item	***OS BERT***	***Anritsu MP8302A***
Cost	£100	£700 - £3500 (Used)
Output Signals	TTL	TTL, ECL
Max Data Rate (Mbps)	50	20
Max Measurement Duration (bits)	4.29×10^9	10×10^{15}
Measurement Precision	Single Bit	2 Decimal Places
Data Channels	3 (Fully Re-configurable)	1
External Clock Inputs	1 (Configurable)	1
Sync Outputs	Up to 2 (Configurable)	None
Max BER (Sync Loss Condition)	No Limit	0.094
Polynomial Pattern Length	2^{32}-1	Variable Up to 2^{32}-1

Each of the devices in Table II has their respective strengths. The MP8302A is capable of multiple output signals, a longer measurement duration and has a variable polynomial pattern length in the LSFR. Of course, it also has the advantage for inexperienced users in that it is able to be used without construction and a technical understanding of the processes inside the device.

However the OS BERT shows a clear advantage in many other aspects. The cost is significantly lower than even the cheapest of pre-used units and it is capable of much higher data rates. Whilst the maximum length of a measurement is limited by the size of the 32 bit registers, OS BERT offers a vastly superior precision compared to the MP8302A. The former can report bit errors to a single bit, whilst the latter can only return figures to two decimal places, even when operated via a PC. The OS BERT also features three, re-configurable data channels. This would prove useful for the measurement of multiple BERs on multiple channels. One such example for measuring the BER on individual conductors of a differential communication link, as discussed in section VI.

An interesting difference to note between devices is based on the limitations imposed on the MP803A by the use of a MAXIM DS2174 EBERT IC. This component will always enter a sync loss state when a BER of 0.094 is detected (6 bits in 64) [17]. Whilst this may be useful in some applications, in experiments conducted by the authors this has proven to be an annoyance as no BER is returned if it is above this figure. OS BERT simply returns the measured number of incorrect bits, regardless of their quantity and/or distribution.

VI. OS BERT MODIFICATIONS

Whilst OS BERT is a flexible platform as-is, its well-documented, open-source nature allows for easy modification for specific experimental requirements. This section briefly examines how two such cases might be implemented on OS BERT. The first, a differential signal BERT module, allows for independent and collective BERs to be measured on differential communication links such as the well-known CAN bus, differential I^2C, and RS422, amongst others. The second example focuses on how an adjustable delay can be used to facilitate measurements on systems with long signal propagation periods.

A. Differential Signalling

Whilst it can be useful to examine the BER of a differential signalling system once the signal has been recombined at the receiver, there are cases where it would be more useful to know whether errors occurred on a particular conductor (for example, CAN-H or CAN-L for a CAN system). With a single channel BERT such as the MP8302A this is impossible, but through a small modification to the OS BERT HDL module, this becomes a trivial task thanks to the multiple re-configurable data channels.

Figure 8 shows a schematic view of a BERT module to perform such a measurement. The 31 bit LSFR PRBG is used to generate a random bit stream. This is then divided into two channels - one is inverted prior to both channels being routed out of the FPGA. A voltage level shift circuit, connected to channels A and B of OS BERT converts the 3.3 V TTL signals to the appropriate levels for CAN. The returning signals are then converted back into 3.3V for the FPGA. A pair of X-OR gates are used to compare the transmitted differential signals with the received bits, permitting a measure of the BER on each conductor, independently. The output from the X-OR gates are combined in an OR gate to provide a total BER for the entire differential system.

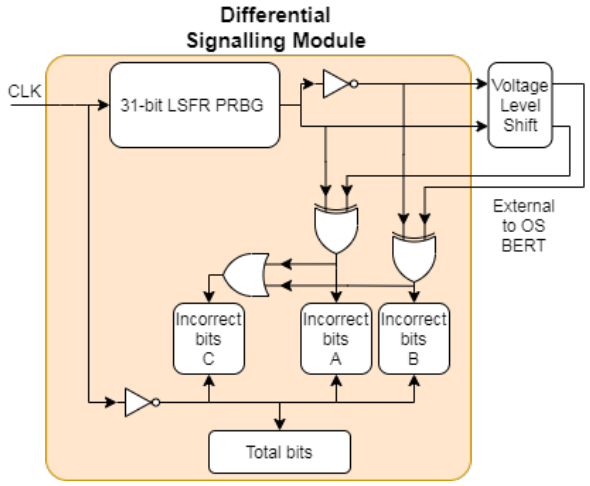

Figure 8. A modified version of the BERT HDL Module in Figure 3 for measuring differential communication links.

B. Re-configurable Sampling Delay

For systems with long signal propagation periods such as Software Defined Radios, or networks with long conductor lengths it may prove necessary to provide a delay in the sampling clock signal to the bit counting registers. This is easily achievable by placing a string of D flip-flops in series. The sampling clock is cascaded through the flip-flops by a delay-control clock, as shown in Figure 9

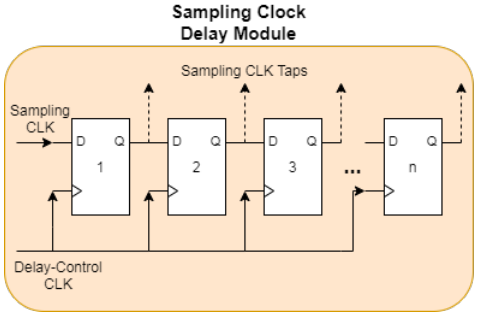

Figure 9. A sampling clock model for high delay propagation systems.

By taking a tap from the output of an n^{th} flip-flop, a delay of the product of n and the delay-control clock can be achieved.

VII. Conclusion and Further Work

This paper presented the work towards the design of a low-cost, Open Source Bit Error Ratio Tester with a focus on the flexibility required for research purposes, which are often overlooked compared to the requirements for industry.

A qualitative analysis was performed against an existing commercial BERT, with strengths and weaknesses of OS BERT discussed. Furthermore, two different modifications for specific experimental requirements were presented in this paper. More detailed explanations will be published within the user-guide for OS BERT, when released.

All associated project files for OS BERT, including: schematics, PCB gerbers, HDL, and software, may be found on the project's website at http://www.etopia-itn.org/openbertuon.co.uk.html [18].

References

[1] Basic Knowledge of EMC Standards, EMC Standards, TDK EMC Technology, International Standards ISO, IEC/CISPR.
[2] B. W. Jaekel, "EMC and Functional Safety – Challenges and Standards", 7th International Symposium on Electromagnetic Compatibility and Electromagnetic Ecology, 2007.
[3] International Telecommunication Union (1996). General requirements for instrumentation for peformance measurements on digital transmission equipment, ITU-T Recommendation G.286 (1993) and CCITT Recommendation O.151 (1992).
[4] IEEE, IEEE Standard Methods and Equipment for Measuring the Transmission Characteristics of Pulse-Code Modulation (PCM) Telecommunications Circuits and Systems, Jan 1992.
[5] N. Marda, G. Vaishnav, U.S.N. Rao, "Bit Error Rate Testing Scheme for Digital Communication Devices", 2014 International Conference on Control, Instrumentation, Energy and Communication.
[6] Tektronix sales contact, private communication, Feb. 2018.
[7] Keysight, "N4962A Serial BERT 12.5 Gb/s User Guide" [online] Available at:http://literature.cdn.keysight.com/litweb/pdf/N4962-91021.pdf.
[8] M. J. Basford, C. Smartt, D. W. P. Thomas and S. Greedy, "On the Disruption of Wired Serial Communication Links by Time Domain Interference" IEEE International Symposium on Electromagnetic Compatibility and 2018 IEEE Asia-Pacific Symposium on Electromagnetic Compatibility (EMC/APEMC)
[9] M. Jeruchim, "Techniques for estimating the Bit Error Rate in the Simulation of Digital Communication Systems" IEEE J. select areas Communication, Vol. SAC-2, pp. 153-170, Jan.1994.
[10] Keysight Technologies. How Do I Measure the Bit Error Rate (BER) to a Given Confidence Level on the J-BERT M8020A and the M8040A High-Performance BERT? [online]. https://www.keysight.com/main/editorial.jspx?ckey=1481106&id=1481106&nid=-11143.0.00&lc=eng&cc=US
[11] Intel. Download Centre for FPGAs [online]. https://fpgasoftware.intel.com/?edition=lite
[12] A. Alimohammad and S. Fouladi Fard, "FPGA-Based Bit Error Rate Performance Measurement of Wireless Systems" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 22, NO. 7, JULY 2014 1583.
[13] Xilinx, "Efficient Shift Registers, LFSR Counters and Long Pseudo-Random Sequence Generators" [online]. Available at: https://www.xilinx.com/support/documentation/application_notes/xapp052.pdf
[14] Potato Semiconductor, "PO49FCT32805G Datasheet" [online]. Available at: http://www.potatosemi.com/potatosemiweb/datasheet/PO49FCT32805G.pdf
[15] Analog Devices, "ADCMP601 Datasheet" [online]. Available at: https://www.analog.com/media/en/technical-documentation/data-sheets/ADCMP600_601_602.pdf
[16] Termite Terminal Software [online] Available at: https://www.compuphase.com/software_termite.htm
[17] Dallas Semiconductor, "DS2174 EBERT Datasheet" [online]. Available at: https://datasheets.maximintegrated.com/en/ds/DS2174.pdf
[18] OS BERT [online] Available at: http://www.etopia-itn.org/openbertuon.co.uk.html