Utilizing FPGA as Synthetic Instruments for Test Reuse

T. M. Mak
A.T.E. Solutions, Inc.
San Jose, CA, USA
tmmak@BestTest.com

Neil G. Jacobson
A.T.E. Solutions, Inc.
San Jose, CA, USA
NeilJ@BestTest.com

Louis Y. Ungar
A.T.E. Solutions, Inc.
El Segundo, CA, USA
LouisUngar@ieee.org

I. Abstract

In this paper, we describe a new paradigm in automatic test equipment (ATE) utilizing Field Programmable Gate Array (FPGA) configurations, called Synthetic Instruments (SIs) to serve as the collection of test instruments that would have been typically housed in an ATE. Unlike ATE instruments that are general-purpose test equipment, such as voltmeters, signal generators and frequency counters, SIs can also perform protocol-related functions that are used in I/O operations. For example, a SI can be a USB tester, which can test the physical layers as general-purpose instruments do but can also help test and diagnose link and protocol layer faults as well. Tests developed in this manner will be more flexible and will enable test reuse, saving on test development time. Additionally, SIs will be flexible to evolving specifications, requiring only reprogramming of the FPGA rather than purchasing a new test instrument.

II. Problem statement

Automatic test equipment (ATE) is a collection of test instruments that apply stimuli and collect/compare responses under the direction of a controller. The controller provides switching and parametric selection services dictated by a test program developed by a test engineer specifically for a unit under test (UUT). Traditionally, ATE instruments comprise of general purpose test equipment, such as voltmeters, signal generators and frequency counters.

Such instruments become obsolete with time, creating a test ceiling. Acquiring a new ATE to meet the ever-changing characteristics of electronics is a monumental undertaking and is quite expensive. It is only feasible for high-volume products that can amortize this cost over the millions of units sold. Furthermore, the test program set (TPS) consisting of the test program, documentation and the device interface board is tailored to the ATE and likely not work on a new ATE or even on an ATE with substituted instruments. The TPS will have to be modified for the new ATE, which may require a complete redevelopment.

Even if cost is not a major issue for certain classes of products, the development of new ATE for these new electronics can take as long (or arguably even longer) a development timeline for the product itself. Sometimes, this leads to volume production waiting for new ATE, a totally unacceptable situation.

There is a need to make ATE flexible enough to changes in technology, so that it can service the testing needs of emerging technologies as well as backfilling to run legacy TPSs.

III. Proposed solution

A. Approach

We are suggesting utilizing field programmable gate arrays (FPGAs) as a development platform for test instruments. We believe it can serve the role of both the individual instrument as well as the controller which runs the test program.

The role of FPGA as a system development/debug platform has been well established for the last 20+ years. The same development language, HDL, is used for chip design and development. For many chip level product designs, compiling and downloading their design into a FPGA or clustered FPGA system allows them to validate and test their designs 100 to 1000 times faster than software simulation. If the design is a processor, that provides the ability to boot up an operating system and its diagnostic software in real time.

FPGAs have also been adopted into some commercial ATEs. Though their role has been limited to certain aspect of logic implementation or some customization of various logic functions that may be needed for the UUT (from one type of products to another). [1]

FPGAs have also been tried out as ATE test platforms by various academic researchers. [3], [2], [4]

As Moore's law progresses, it is now possible to integrate the entire ATE into a single FPGA unit, with ~25M gates to spare (and 100M gates with various SiP technologies). This gate count does not include any embedded ARM processor cores or fixed IP fabricated onto the FPGA itself, so valuable FPGA resources need not be wasted to synthesize a processor core, which is inevitably needed for executing a test program.

B. *Highlights of technical challenges and solution*

High-speed input/output (HSIO) applications are good examples of test instrument obsolescence. Years ago, as data rates (on a wire as part of a bus) exceeded 1 gigabit per second (Gbps), signaling morphed from parallel to serial. As part of the digital communication system, Serializer/Deserializer (SerDes) was used to handle transmission between the parallel digital system and the serial nature of the communicating medium). Unique transmitter (Tx)/receiver (Rx) pairs were used to provide various signal forming and reconstruction techniques to send data over inexpensive cables. SerDes not only converts data between serial data and parallel interfaces in each direction, it must also embed the clock into the transmitted data. On the receiving end, a clock recovery circuit, such as a phase-locked loop (PLL) is used to recover the frequency/phases of the transmitted clock, which is then used to strobe the data stream to recover the data. Traditional instruments used on ATE are not able to make measurements such as jitter and noise at these frequencies. Stand-alone bit error rate testers (BERT) are used to characterize the differential transceiver responses in what is called an eye diagram. The eye diagram is subject to a fair amount of interpretation and requires human analysis. Due to these factors, BERT is not traditionally found on an ATE.

Data rates has also been rising fast. The initial introduction of a serial system level interconnect (e.g. the USB). ran at the relatively slow speed of 1.5/12 Mbps. It was then increased to 480 Mbps in 4 years. In another 8 years, it jumped to 5 Gbps, followed by an increase to 10/20 Gps in recent years. That is a 10,000x performance boost in just 20 years. In order to cope, the test industry's strategy must involve designing custom HSIO test circuits.

Fortunately, the FPGA industry has been leading the development/prototyping field, so they have been at the forefront of the signaling trend. Very often, they produce FPGAs with tens of high-speed transceivers that are capable of supporting multiple protocols. Their SerDes and PHY provide industry-leading data rates (e.g. one company recently claims 112 Gbps! [8]). So, raw performance is not a problem.

To assist their customers with leading edge FPGA-based designs, FPGA manufacturers also produce various development boards that highlight some of their product features. For example, one can easily find boards that include many high speed SerDes signals, all pinned out to the SMA cables [5], making it virtually a piece of equipment itself (minus the housing).

What about the protocol support? An FPGA is a large bag of gates, but without the proper hardware design, it is still a BIG bag of gates. Fortunately, the heavy lifting has been handled, as there are many independent design houses churning out Intellectual Property cores (IPs) or designs for various input/output (IO) so that their chip design clients can produce a complete design. We can expect to have a FPGA compatible IP by the time the USB4 specification is complete. The availability of such IP would enable the rapid development of USB4-capable ATE.

There is a lack of high precision analog instrumentation on FPGA chip itself. FPGAs are primarily logic gates, along with high speed transceivers and some hard IP.. Precision analog devices are not incorporated in any FPGA offering. Some FPGAs contain a simple ADC that can be made into a voltmeter for internal power buses or temperature sensors, But these ADCs lack the speed, precision and accuracy necessary for the exacting needs of ATE instrumentation.

The broader FPGA ecosystem provides assistance in the analog domain.. Reputable companies have developed good reference designs (i.e. actual boards with their analog chips on them) for many FPGA development systems [6]. They even provide sample code for using an FPGA system as a reasonable oscilloscope or a good voltmeter. So, these can be used to produce an inexpensive analog subsystem as part of the FPGA ATE.

Beyond low frequency analog, it will have to work (integrate) with some other RF subsystems, optical transceivers, etc. to handle other special test duties. Again, due to its flexibility in system prototyping, there are ample hardware and software designs and implementations so that one doesn't have to start from scratch.

Such an FPGA ATE can be programmed to work with a PC, an older ATE, and even be able to mimic an obsolete ATE should the need arise.

Reuse is addressed by looking at several angles. First, the FPGA chip or even the development system is totally re-programmable from one project to the next (or from one type of UUT to another). The downloaded FPGA bitstream and software will change the functionality of the whole system. Should a project be cancelled and a new one is started, you can reuse the same development board. Second, individual synthetic instruments, like a voltmeter or signal generator/checker can also be recycled from one type of UUT to another type of UUT. The external interconnect and test flow may differ, but the test function does not. Third, many high speed IOs have very similar physical layer (PHY) characteristics. So, this portion of the test configuration and test code can also be reused from one type of interface to the next. Lastly, should an upgrade (performance, gate count, and/or other function) become necessary, the HDL of the old FPGA system can be readily ported to the new FPGA. This is preferable to having the old schematic redesigned.

IV. EXPERIMENTS AND ANALYSIS

A. *Experiment Setup*

A Xilinx VCU1287 development board was purchased to add as the development platform for our FPGA ATE. Details of the FPGA and the board can be found at [5]. It is

hooked up to a commercial Intel Core i7 PC that has 64GB of memory. This larger memory footprint allows a large design to be complied more efficiently.

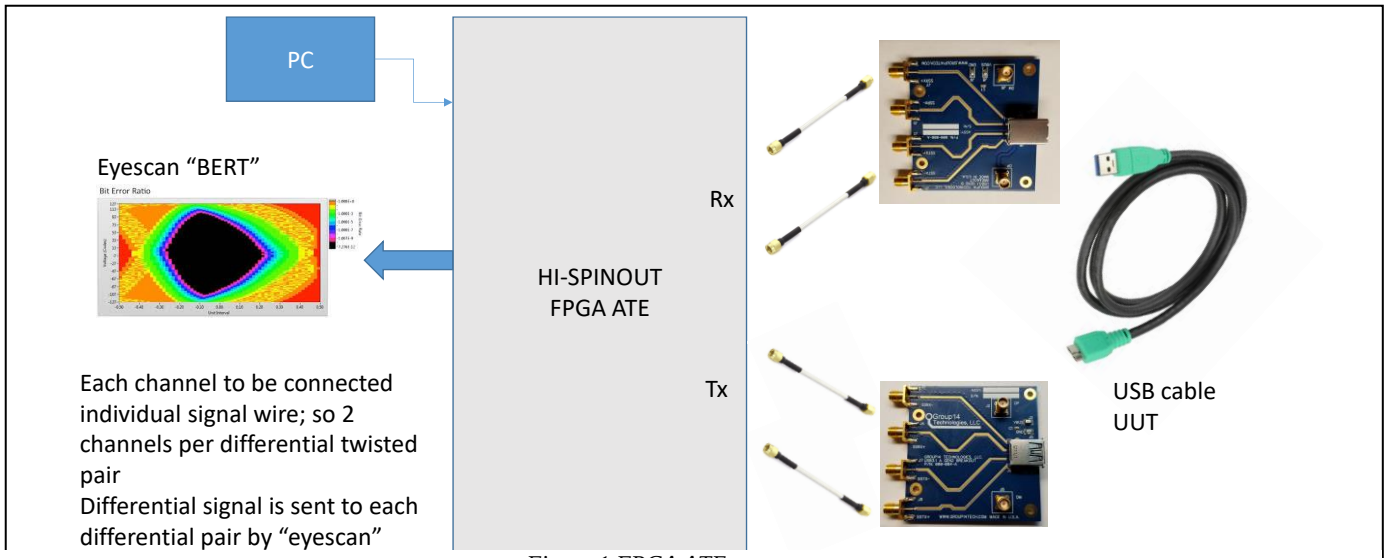

Figure 1 FPGA ATE current setup

B. *Results*

We have successfully booted the machine. We installed the Eye Scan BERT app [9] [10] from Xilinx and it is already useful in characterizing faulty cables [11].

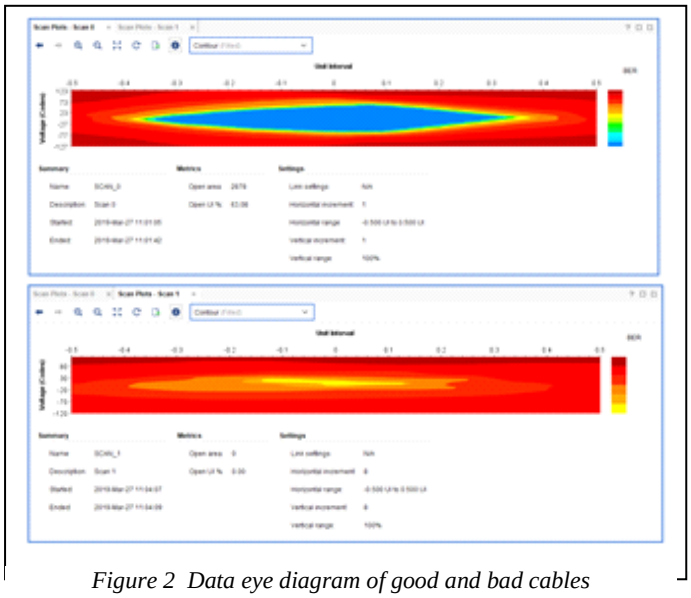

Figure 2 Data eye diagram of good and bad cables

Figure 1 illustrates the setup and connections for this experiment

In the top graph of figure 2, we can observe a wide-open eye (running a 12.5 Gbps random signal) with a known good cable. We then replaced the good cable with a cable where we cut one signal out of the twisted pair, creating an open. This has severely diminished the eye as shown in the bottom graph of figure 2. However, we observed that the

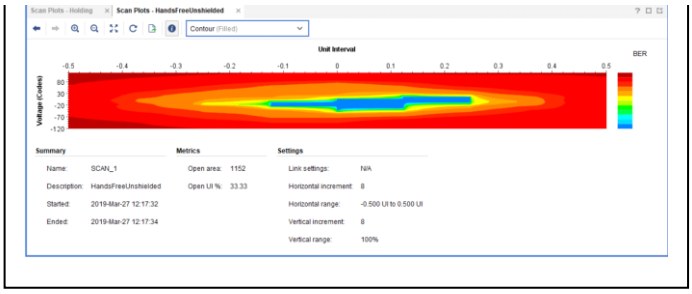

Figure 3 Data eye diagram with faulty cable (unshielded wires)

eye re-opened at lower data rate, indicating that a broken cable can still marginally function (with the inherent USB error correction protocol). This functionality, however, will give more errors, which impede performance, but may otherwise hide fault.

In figure 3, we stripped away portion of the protection shielding. This introduced more coupling among the pairs and caused the data eye to shrink.

We are porting the Xilinx application code so that we can modify it for our extended purposes.

We expect to select an external USB IP soon to speed up our testing of the protocol stack and a standalone UUT that does not have DFT support.

V. COMPARISON TO STATE-OF-THE-ART

Clearly, this is a work in progress, so there are lots of projections included here.

Comparing with a BERT bench setup, this may not be as accurate, but then a complete BERT setup would cost 100s

of thousands (maybe million) of dollars, as well as a long training curve for someone to learn it. Despite the high price tag, a discrete BERT instrument won't transform itself into an ATE that can test other types of products. Our FPGA ATE, however, should be able to do that.

Comparing with a high-end state-of the art ATE, this is much lower cost, and most flexible, as new instruments can be added with potentially little cost and effort. It has a much smaller physical footprint and can be brought to the field for testing, which the high-end test instrument would never be able to do.

ACKNOWLEDGMENT

This effort was partially funded by US Navy contract N6833518-C-0165, a small business innovative research (SBIR) Phase II contract entitled Advanced High Speed Bus Technologies for Units Under Test (UUT), Test and Evaluation.

REFERENCES

[1] Schreier, L., Improving Instrumentation With User-Programmable FPGAs, EE Evaluation Engineering July 1, 2008; https://www.evaluationengineering.com/home/article/13003905/improving-instrumentation-with-userprogrammable-fpgas

[2] Keezer, D.C., et. al. Low-Cost Multi-Gigahertz Test Systems Using CMOS FPGAs and PECL, ETS 2015

[3] Vanitha, K., C. A. Sathiya Moorthy, Implementation of an integrated FPGA based automatic test equipment and test generation for digital circuits, ICICES 2013

[4] Alp Arslan Bayrakci, ELATE: Embedded Low Cost Automatic Test Equipment for FPGA Based Testing of Digital Circuits, ELECO2017

[5] Xilinx VCU1287 development board ; https://www.xilinx.com/products/boards-and-kits/ck-u1-vcu1287g.html#hardware

[6] Analog Device Evluation Board; https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-ad-fmcdaq2-ebz.html#eb-overview

[7] Shawn Hu, FPGA development in ATE, AutoTestcon 2017

[8] Intel® Next Generation 10NM FPGAs

[9] In-System Eye Scan of a PCI Express Link with Vivado IP Integrator and AXI4, Xilinx application notes xapp1198

[10] Eye Scan with MicroBlaze Processor MCS, xapp743

[11] Ungar, L.Y., Jacobson, N.G., Mak, T.M., Creating Reusable Manufacturing Tests for High-Speed I/O with Synthetic Instruments, Proc. IPC APEX Conf. Feb 2019.