Adaptive FPGA-based Test Module

April 30, 2013 (Initial DRAFT)

Hardware Design Task – Prof. Keezer, et al
Algorithms- Prof. Chatterjee, et al

Executive Summary (from the Interim Report, December, 2012)

This project began in June, 2012 and has proceeded for the past 6 months until the present. It is scheduled to be completed in May, 2013. The objective of the project is to develop methods and electronics for testing multi-GHz digital components (such as DDR memories), using low-cost methods based on state-of-the-art field programmable gate arrays (FPGAs). A further objective is to incorporate the means for the test electronics to “self-monitor” its own performance, and to “adapt” its behavior (performance) in order to optimize the quality of the test signals. Therefore this project seeks to realize two major benefits as compared to traditional testing methods: (1) lower test equipment cost and (2) improved test signal quality (especially for high-speed signals). This report describes the activities and preliminary results obtained during the first six months of this project.

Objectives

1. Develop an approach for testing multi-GHz digital components, using low-cost FPGAs and minimal additional electronics components.
2. Incorporate self-adaptive capabilities that allow the test electronics to monitor its own performance, and to adapt its characteristics so as to optimize the test signal quality.

Approach

Our approach to solving this problem is based on the realization that FPGA technology has progressed dramatically in the recent several years, to the point where in many ways it has become a leader of the technology. Today FPGAs are available that utilize 28nm CMOS technology and support I/O rates of ~30Gbps (at the high end, using dedicated serial I/O pins). Even “standard” I/O speeds are up to ~1Gbps for these new FPGAs. Furthermore, the cost of FPGAs is very low as compared with custom electronics, assuming low-volume applications such as ATE. Of course FPGAs have always had the significant advantage of re-programmability, so that design changes and improvements can be realized without incurring significant re-fabrication costs. Therefore our main focus is to leverage these features of state-of-the-art FPGAs in order to create a low-cost, but high-performance test system.

However advanced and capable the FPGAs may be, we recognize that (1) they too have limitations and that (2) we may need additional circuitry to achieve the performance targets for testing. Furthermore, we realize (3) that some modifications to traditional test strategies might be needed in order to enable full utilization of FPGA technology for testing, and (4) that significant performance gains can be obtained by innovative techniques for self-monitoring and self-correcting test signals.
With these issues in mind, we adopt the following in our approach:

1. Select an appropriate FPGA, based upon leading (yet available) technology.
2. Determine the capabilities and limitations of the FPGA with respect to the testing application.
3. Complement the FPGA features with additional (but minimal) circuitry in order to realize the desired test capabilities. This results in added “pin electronics” (PE).
4. Selectively enhance the PE to provide off-line and on-line self-monitoring and correction/optimization capabilities for improved test signal performance.

Our project, therefore, includes the following tasks:

1. FPGA evaluation and selection
2. Pin electronics design, prototype, evaluation
3. Test module electronics system-level design and integration
4. Development of self-monitoring/optimizing enhancements
5. Incorporation of control and analysis algorithms (working with Prof. Chatterjee et al)
6. Characterization and demonstration of the test system’s capabilities and performance

Summary of Activities and Results

(1) FPGA evaluation and selection (see also Appendix A)

In our proposal for this project we identified the Xilinx Virtex-7 FPGA as a good candidate for use in this application. Therefore, early in the project we more carefully reviewed this possibility. We quickly found that, while the technology was very advanced, it appeared to be difficult to obtain. We found very limited (if any) availability of these FPGAs from standard component suppliers. Mostly it appeared that there would be very long “lead times” in some cases several months, and in other cases not even specified. While the website indicated an entire “family” of products, with many possible configurations, it appeared that only a very small subset was actually in production and/or otherwise available. In all cases the prices for Virtex-7 were very high (several thousand dollars per part). Furthermore, we found that the available design documentation from Xilinx was lacking, so that it was difficult to predict some performance aspects.

Due to the difficulties in obtaining Virtex-7 at the time, we looked for alternatives with the understanding that our initial prototype efforts did not necessarily need all the features of virtex-7. So initially we utilized an available Virtex-6 evaluation board that we had obtained during the previous project with Samsung. Virtex-6 has many similar functional features as Virtex-7, except that it is built using the older 40nm process (a bit slower, and less dense). So, we were able to experiment with the virtex-6 and gain an understanding of how it or virtex-7 could be used for our application.

About the same time that we decided to try virtex-6, we also found that Xilinx offers a product similar to Virtex-7 called Kintex-7, which is based on the same 28nm process. The main difference that we saw between Virtex-7 and Kintex-7 was that the Kintex product line was more readily available in smaller packages with fewer I/O and somewhat lower gate-count, at lower cost. From the initial specs, it looked
like Kintex-7 had a good combination of features suitable for our project. The fact that it was closely related to Virtex-7 meant that we would have a convenient development path (to Virtex-7) if we chose Kintex-7 for our initial prototype efforts. At the time (July, 2012) we did find Kintex-7 components were available through suppliers, and also an evaluation board was available. Therefore, we decided to choose the kintex-7 FPGA for our initial prototype development, while leveraging some work with the Virtex-6 and leaving open the possibility of migrating to Virtex-7 in the future.

After first working with the Virtex-6 evaluation board, we obtained the kintex-7 evaluation board and performed several experiments to determine and/or verify its performance. In almost all respects we found that the part performed extremely well and was well suited to our application. There were some minor anomalies noted, but none that would significantly impact our progress. Similar to the Virtex-7, some documentation was lacking. We have attached examples of measurements made on the kintex-7 in Appendix A. These measurements gave us the understanding of the part’s capabilities that was needed to confirm its selection for this project.

The evaluation of the Kintex-7 FPGA is summarized as follows (see Appendix A for more details):

- The GTX transmitters worked exceptionally well at 3.2Gbps, and very well up to 10-12 Gbps.
- Overall signal quality was remarkably good up to 10-12 Gbps
- Min/Max single-ended swing was measured as ~140mV-620mV with ~50mV resolution
- Jitter was measured at 14-21ps peak-to-peak at BER=10⁻⁴
- Quad-PLL works best at higher frequencies (>6Gbps)
- Rise/Fall times were ~42ps (20-80%)

Figure 1 – Kintex-7 Evaluation (photo) – Shown connected to PE Proto#1. Lab equipment used for evaluation includes Agilent 86100c Oscilloscope and Agilent 81133a signal generator.
(2) Pin electronics design and prototype #1 (see also Appendix B)

In parallel with the evaluation of Virtex-6 and Kintex-7, we began developing initial designs for the PE that would be needed to support these FPGAs for signal generation and capture and for the added self-monitoring functions. The first prototype design was built using standard FR4 PCB material with standard layout approaches, and without fully optimizing all design features. The purpose of this prototype was mainly to validate our choice of components and to check for unforeseen problems in the basic design. In fact, we ended up spending a good amount of time using this prototype to characterize the individual components (especially the Driver IC). We did this beyond the level that was planned, and to the point where we realized that these measurements were limited by the materials and layout details of the PCB hardware, as well as some minor design details. So, while the initial measurements looked very promising, we decided that a second (improved) prototype was needed in order to make sure that we fully understood the PE capabilities (see the next section).

The first PE prototype included the basic PE functions of Driver and Compare, but also some enhanced PE features such as Driver Pre-emphasis control and a “shadow” sampler (2nd parallel comparator for under-sampling, eye-monitoring, calibration, and diagnostics). At the time its design, we assumed that the general test channel would be (a) bidirectional, and (b) single-ended (as opposed to differential).

There were two competing scenarios regarding the transmission line connection to/from the DUT, namely (1) Single-transmission line, and (2) dual-transmission line. The dual transmission line has a significant disadvantage in that it greatly complicates the test/DUT interface. There is also the question of the “stub” effect where the two transmission lines connect to the DUT pin. The advantage to the dual transmission line approach is that it avoids the “dead zone” timing conflict at the PE comparator that is found using the single-transmission line approach. Later (at the Intl Test Conf in November) we found a variation of the single-transmission line approach called “simultaneous bidirectional signaling” that has the economic benefits of the single transmission line approach, and avoids the “dead zone” issue like the dual transmission line method. Therefore, we have been concentrating on the single transmission line approach since then, while keeping the dual-line approach as a back-up.

We also found that the Xilinx Kintex or Virtex FPGAs do not provide a convenient way to adjust skew of the transmitters over a moderately large range (several nanoseconds is needed). Therefore, a programmable delay IC was included in the PE design to allow us to control the Driver signal timing. The same IC was used to program the shadow-sampler clock phase, to a resolution of 5ps.

A summary of key features for the PE Proto#1 components is as follows:

**Driver** – The Driver must be able to handle I/O rates of at least 3.2Gbps without significant distortion. It must have a programmable output swing of ~100mV to ~1.5V and about 1.5V DC offset range. It must not add significant jitter (<1ps RMS, <10ps DDJ). In addition to these basic properties, the device must have a small footprint (<1cm²), and not dissipate too much power (<1Watt). These properties are needed in order to allow for high-density (high pin-count) interfaces. Later we added the requirement that the Driver provide differential output signals (required for implementing the simultaneous bidirectional signaling method). Other features that are desirable include: pre-emphasis control, jitter...
injection, pulse-width distortion adjustment, rise/fall time adjustment, and low-cost (although the Driver IC cost is not a major concern). Meeting all the requirements and many of the desired features is the Micrel SY58626 device, which we have tentatively selected for this project. Much of the measurement activity shown in Appendix B was devoted to checking the Driver performance while embedded in a typical PE configuration that included a power splitter, two comparators, and typical transmission lines to SMA connectors.

**Primary Comparator** – The “Primary” comparator is used in the PE to capture the DUT response, digitize it based upon a user-defined threshold voltage, and to produce full-swing differential output to the FPGA GTX receivers (RX). Therefore this comparator must be able to operate with single-ended input signals and produce fully-differential outputs for the FPGA. It must operate in “transparent” mode, i.e. without clocking. It further must be very sensitive to small (<50mV) single-ended input signals, while supporting ~10GHz bandwidth and low jitter. It must be small (~3x3mm), and dissipate minimal power (<100mW). Meeting these needs is the Hittite HMC674, which we have chosen for the project.

**Secondary (“Shadow”) Comparator** – This device is used for under-sampling the DUT response signal in the form of a “shadow” sampler. It shares all the requirements of the Primary Comparator, but additionally requires low-jitter clock input(s) for latching the sampled data bit. We found that the Hittite HMC874 device works well for this component.

**Delay IC** – This device must have a resolution of ~5ps or less, and support a range of at least a few nanoseconds. Like the other components, it must be small, low-cost, low-jitter, low-power. We found that the Micrel SY89297 device satisfied these needs.

The evaluation of the initial prototype is summarized as follows (see Appendix B for more details):

- When evaluated separately, the Driver IC performed well at 3.2Gbps and even up to 6.4Gbps (its spec). However, some signal degradation was noted due to the lossy FR4, the SMA launch, and especially the power splitter.
- When the power splitter was configured to connect to the Comparators, we noticed degradation of the signal.
- We had recurring problems with both the ‘674 and ‘874 comparators due to sensitivity of their input protection circuits to power-supply biasing sequence (NOT specified by Hittite). This actually cost a lot of time since we had to remove and replace these parts several times during the course of our development and evaluation. We also had numerous conversations with a Hittite applications engineer about this problem, in which we started to get a better understanding of the problem. However, it is still not completely resolved. We now suspect that the parts are being damaged during the power-up or power down sequence. We are now taking steps to further protect the inputs against these conditions.
Figure 2 – PE Proto – Circuit Diagram (see Appendix B for detailed diagram)
Figure 3 – PE Proto – Layout (see complete layouts in Appendix B)

FPGA-based Multi-Gbps Self-monitoring Pin-Electronics Evaluation Board

Figure 4 – PE Proto#1 (FR4) – Photo – The 3 red circles highlight the Drive and two Comparators

Note: This is a prototype, for initial evaluation only. (FR4 material, etc)
Figure 5 – PE Proto#1 – Driver performance – 1.6Gbps

Figure 6 – PE Proto#1 – Driver performance - 10 Gbps – No Pre-emphasis

(3) Pin electronics design and prototype #2 (see also Appendix C)
As noted above, during the evaluation of the initial PE prototype we found that the full capability of the Driver was not realized completely due to the limited design-optimization and choice of FR4 as the PCB material. Therefore, in a second PE prototype, we improved several aspects of the design as well as used low-loss dielectrics (Rogers 4350) in the PCB fab. In a second (improved) prototype#2 we made the following changes:

(a) Replaced FR4 with Rogers 4350 (low-loss dielectric) for the outer.
(b) Improved the grounding via arrangement and pad geometry of the edge-launch SMA connector.
(c) Changed the Driver and Delay IC DC bias to better match the DUT operating range.
(d) Improved/optimized the power-splitter geometry to reduce parasitic effects, and make a better impedance match to Z0=50 Ohms.
(e) Replaced the ~100 Ohm transmission lines to the comparators with 50 Ohms.
(f) Improved the power distribution and decoupling for the Driver and other ICs.
(g) Built a similar PE, except with the Dual Transmission line arrangement.
(h) Added biasing resistors for the ‘874 clock inputs.

As a result we obtained better performance and more fully realized the full capabilities of the Driver and Comparator components. These are summarized below (see also Appendix C for more details):

- The Driver was demonstrated at 3.2Gbps with excellent performance.
- The Driver was found to work very well at 6.4Gbps (its spec), but showed a tendency for pulse-width distortion.
- Surprisingly the Driver was able to handle signals up to 12 Gbps, with some degradation in quality (slightly higher jitter above 10 Gbps).
- The comparators also worked well up to 10Gbps (‘674) and 12Gbps (874) with some added jitter for the ‘674 device. We believe that much of the added jitter observed in the ‘674 was due to damage of the input protection circuitry, even though the device continued to function it was degraded.
- Overall signal quality was noticeably improved (sharper edges, less overshoot, less ringing/ reflections) as compared to PE Prot#1.
- The Driver biasing to match the DUT/Comparator range worked as expected.
- The Driver still showed a tendency for pulse-width distortion. We will investigate this issue and look for a way to improve it.
- The optimized signal splitter worked much better at matching to 50-Ohms, and accounts for at least some of the observed improvement in the Driver output signal quality.
- Unfortunately we continued to have trouble with the ‘674 and ‘874 input protection circuits. We strongly suspect that this is a result of sensitivity to the power-up sequence. We will continue to explore this issue. One possible solution may be to use additional protection elements that keep the inputs safely-biased during power-up. We’ve already done this on the ‘874 clock inputs, and it appears to have worked for these inputs.
Figure 7 – PE Proto#2 – Driver performance – 3.2Gbps (with 10% 100ps pre-emphasis)

Figure 8 – PE Proto#2 – Driver performance - 10 Gbps
Figure 9 – PE Proto#2 – ‘874 Comparator output – 3.2 Gbps

Figure 10 – Driver Output at 3.2Gbps with no pre-emphasis.
Notice that p-p jitter is 20.8ps, signal amplitude is 400mV.
Notice that the rising edge does not reach the upper rail.
Figure 11 – Driver Output at 3.2Gbps with **minimal pre-emphasis** (60ps, 10%).
Jitter and signal-amplitude are the same as in Fig.10.
Notice that both rising and falling edges swing between the rails.

Figure 12 – Driver Output at 3.2Gbps with **minimal pre-emphasis** (60ps, 10%).
Expanded time-scale (39ps/div).
Notice that p-p Jitter is 18.2ps at 2-khits.
Figure 13 – Driver Output at 3.2Gbps with **minimal pre-emphasis** (60ps, 10%).
Expanded time-scale (39ps/div), and longer data-acquisition (60-khits), jitter is 24ps.

**Effect of increasing pre-emphasis duration (60ps to 100ps)**

Figure 14 – Driver Output at 3.2Gbps with **minimal pre-emphasis** (60ps, 10%).
Time-scale (79ps/div), data-acquisition (2-khits), jitter is 24.6ps.
Compare to Fig.15, to see the effects of longer pre-emphasis duration.
Figure 15 – Driver Output at 3.2Gbps with longer pre-emphasis (100ps, 10%).
Time-scale (79ps/div), data-acquisition (2-khits), jitter is 21.1ps.
Compare to Fig.14, to see the effects of longer pre-emphasis duration.
(4) Test module system-level design and integration - ”Main” Board (see Appendix D)

As a result of our successful evaluation of the Kintex-7 FPGA, we were confident it would work well in our application. We therefore proceeded to develop a “Main board” that would support the Kintex-7 and up to 16 PE cards within an ATE system (specified by SEC). The Main board receives power and control signals through several high-pin-count connectors near its bottom edge (see the floor-plan diagram in Fig.16). The board is divided into 4 sections, each of which supports 4 connectors for PE modules/cards. Initially we plan that each PE card will have 4 channels, so the total would be 64 channels. In this prototype design, we actually connect only 4 of these to a single Kintex-7 FPGA, which is located to the right of the board center. The other areas show that expansion to 64 I/O channels is straightforward. The 16 PE card connectors are arranged across the top of the Main Board so that each PE card will be perpendicular to the Main Board. Connections to the DUT interface board will be through 4 SMP connectors pointing upwards from each PE card. In a fully-configured arrangement there will be 64 SMP blind-mate connectors pointing upwards from the PE cards to the bottom side of the DUT interface board. During the prototype evaluation we will limit this to 16 channels, maximum.

![Figure 16 – “Main board” floor-plan and mechanical properties.](image)

Schematics for the Main Board have been developed and are shown in Appendix D. The layout of the Main Board has been subcontracted, and is currently underway. Preliminary estimates are that it will be fabricated using ~20 layers, 3mm thick, using either blind vias or back-drilled to avoid stubs, and using Nelco low-loss dielectric material for the critical signal layers. We expect the board to be fabricated in January 2013.
(5) Design of 4-channel PE card (preliminary)

Largely as a result of our experiments with PE proto#1 and #2, we are now confident that we can build a successful PE card with multiple channels. We will be completing the design/layout in early January, and fabricating the PCBs around the end of January. Most of the critical logic will be similar to the earlier designs. However, we will be arranging the circuitry for 4 channels onto a 3”x2” PCB. An approximate floor-plan for this card is shown in Fig.17. On the bottom edge of the card is a Samtec 104-pin edge-mounted connector. This will be used to connection to the Main Board for signals, grounds and power. On the right side edge are four blind-mate SMP connectors that carry the main signals to/from the DUT. The logic design of each channel will be similar to PE Proto#2, with the following changes:

(a) Addition of a mechanical RF relay to allow switching between high-speed and DC test.
(b) Substitute serial-programmable DACs for previous potentiometers (for Driver and Comparator control voltages).
(c) Provide a high-current DAC for control of the Driver DC-offset (VTerm), replacing an external power supply input.
(d) Provide an option for implementing the “simultaneous bidirectional signaling” approach using the primary comparator (‘674). This will allow the FPGA to receive the DUT output without interference from the Driver signal, thereby eliminating the “Dead zone” timing problem.
(e) Replace SMA connection to the DUT with SMP connectors.

Figure 17 – 4-Channel PE Card floor plan (preliminary).

(6) Self-monitoring/optimization enhancements
The main enabler for self-monitoring in our initial design is due to the presence of the “shadow” sampler. Because this measurement element operates independently from the primary sampler, it can be used to gather channel-characterization information even while the normal functional testing is performed (by the primary sampler). Therefore it does not impede the normal functional test process. On the other hand, while the primary sampler will normally sample in the middle of the data eye, the shadow sampler can be programmed to observe the signal characteristics near the data-eye boundaries. This is where distortions in the signal will first occur, and is where information about non-ideal operating characteristics is available. In an extreme situation, a failing channel can be quickly identified. However, even slightly-degraded channels may be detected with the shadow sampler. Furthermore, since we are not planning to implement the simultaneous bidirectional signaling approach in the shadow sampler, we will have access to other information that is hidden from the primary sampler (namely the Driver signal). So we can use the shadow sampler to observe the Driver signal quality and the DUT signal quality, as well as distortions from defective channel transmission lines. In an off-line mode, the shadow sampler will facilitate TDR measurements, Driver/Comparator de-skew and voltage calibration, and channel length calibration. Therefore, while the addition of a shadow sampler, at first appears redundant, in fact it is very effective at enabling many channel characterization techniques.

The choice of the Micrel driver, with amplitude/offset-voltage/phase/pre-emphasis adjustments also means that we can use the shadow sampler information to correct or enhance the Driver signal quality. We have already demonstrated (in PE proto#1 and #2) that a small amount of pre-emphasis can improve the signal. We are looking forward to further developing these techniques when we have the 4-channel PE cards connected to the Main board in 2013.

Figure 18 – Data Eye Monitoring.

(7) Introduction to new material (January – April 2013)
**Review** - During the first 6 months of the project (July-Dec 2012) we concentrated on:

(a) Establishing the system-level approach.

(b) Initial design of the “Main board” (Kintex7 FPGA controller, etc)

(c) Design of prototype “Pin Electronics” (PE) channel, including Driver, dual receivers, delay, and clock distribution.

(d) Fab, assembly, and test of two PE prototype evaluation circuits.

(e) Evaluation of Kintex7 capabilities.

(f) Preliminary development of TDR and eye-monitor algorithms.

(g) Initial floor plan and top-level circuit design for a 4-channel PE card.

(h) Interim Report preparation.

**Progress Summary** – During the latest 4 months (Jan-April) we completed the following tasks:

(a) Completion and update of the main board design (January - COMPLETE).

(b) Completion of the first 4-Channel PE card design (January- COMPLETE).

(c) Fab and Assembly of the main board (February- COMPLETE).

(d) Fab and Assembly of the first 4-channel PE card (February- COMPLETE).

(e) Debug and evaluation of the main board (Feb/March- COMPLETE).

(f) Test of the first PE4 card (Feb-April- COMPLETE).

(g) Develop and apply TDR and eye-monitor algorithms using the Main Board and PE4 card 
   (March/April- COMPLETE)

(h) Redesign the PE4 card to correct DC-offset error and other improvements (March/April- COMPLETE).

(i) Fab the second PE4 card (April- COMPLETE)

(j) Assemble/Test the second PE4 card (late April, in-progress)

(k) Test and evaluate the TDR and eye-monitor performance (April, in-progress)


(8) **Main Board design**
The Main Board design was completed in January. It was designed using a 20-layer PCB stackup configuration that supports 8 signal layers, 8 ground planes, and 4 split-plane power supply layers. The dielectric was NELCO. The total designed board thickness was 3mm (about 120 mils), which is twice as thick as standard (0.062") PCBs. A sketch of the main features of the PCB is shown in Fig.19. These include:

- Xilinx Kintex-7 FPGA used as a central controller
- Digilent JTAG interface module
- Cypress USB 2.0 interface
- SMAs for clock and evaluation signals
- Utility LEDs and user-definable DIP switches
- Positions for 16 Samtec 104-pin high-density connectors (10Gbps capable)
- 4 populated Samtec connectors (each for supporting one of the PE4 channel cards)
- Two multi-pin connectors for providing multiple power supplies from the host tester
- Two multi-pin connectors for providing logic signals to/from the host tester
- Auxiliary power connectors (for bench-top operation during prototype development)

A photograph of the Main Board is shown in Fig.20. Updated schematics are included in Appendix E and the final layout drawings (“Gerbers”) are shown in Appendix F.

Figure 19 – Main Board key features.
After developing the schematics for the Main Board, a 3rd party contractor was used to perform the layout of the 20-layer board. This occurred in November-January. Then the layouts were sent to another PCB fabrication company, which built the boards in January. Two copies were delivered to Georgia Tech in early February. One of these had “back-drilled” vias for the most critical signals. The other did not. Since we were concerned that the back-drilling might have damaged the vias, we chose to try the non-back-drilled version first. We assembled the Main Board using our in-house prototyping facilities, including the mounting of the 900-ball FPGA, using stencil-printed solder-paste. Some photos of this hand-assembly process are shown in Fig.21. The in-house hand assembly process allowed us to incrementally build/test portions of the Main Board, so that each section could be checked before proceeding. In most cases we were able to proceed from one step to the next without significant rework. Some of the finer-pitch devices required some rework. We also found a problem with the power pins of the Samtec socket, which were intended for 0.062” thick PCBs. Since ours was about twice that thickness, these pins did not extend far enough through the board to allow hand-soldering. We eventually got good solder connections using a very fine tipped iron and a lot of heat. Otherwise, the main board assembly/test proceeded smoothly.
Figure 21 – Main Board Assembly process (Feb.15, 2013).
Figure 22 illustrates one (of four) channels in the first (Rev1) PE4 circuit. It consists of a programmable Delay chip (U1) that takes high-speed differential data from the Xilinx Kintex-7 via a SamTec multi-pin connector (J1), and provides a delayed version of that data to the Driver IC (U2). The Driver output passes through a resistive “splitter” junction that produces three outputs. Two of the splitter outputs are “probed” using 200 Ohm series resistors together with 50 Ohm terminations within the two comparator ICs (U3 and U4). These 200/50 Ohm resistors form a 5:1 resistive divider with an effective impedance of 250 Ohms, so each comparator receives an attenuated version of the voltage at the splitter junction. Most of the signal energy passes through the two 10-Ohm series resistors, then through the relay (U5) and SMP to the device-under-test. For signals returning from the DUT, the active Driver forms part of the termination structure (along with the resistive dividers of the two comparators). The symmetric arrangement of the junction resistors provides a 50-Ohm impedance match looking into the junction either from the Driver or from the DUT. Therefore, the high-speed logic signals pass through the junction in either direction without significant distortions or reflections. In practice, the junction is made up of non-ideal resistor elements (with parasitic inductance and capacitance), so there is a small distortion.

Figure 22 – Block diagram of the PE4 (REV1) channel #1 circuit.
The Driver compliment output is shown with a terminating load resistor (R=60), so that the differential CML circuit remains balanced. In fact we found out experimentally that the CML outputs do not work so well with the termination set to Ground. In the final version of this circuit (see later section) the design was changed so that the termination was to Vref4, which is set slightly above the logic “high” level. For this reason, most of our measurements on the first PE4 circuit were done with an AC-coupling capacitor replacing the first 10-Ohm series resistor, and also AC-coupling the complimentary Driver output signal.

Also shown in the figure is an optional 210 Ohm resistor that connects some of the Driver Compliment signal to the leg of the junction that produces the signal for the “Primary” comparator (U3). This signal should cancel the attenuated Driver signal for the primary comparator. The only remaining signal component would be that of the DUT. This is an approach called “Simultaneous Bidirectional Signaling” that allows the primary receiver to “see” only the DUT signal, while ignoring the Driver output. Therefore the Driver can send data even at the same time that the primary comparator is receiving data from the DUT. In this way the PE and the DUT can share the single transmission line for bidirectional communication, while avoid “dead zone” timing conflicts. So far (as of April) we have NOT implemented this optional arrangement in the PE4 prototype. Instead, we have left off the 210 Ohm resistor, and tested the channel in one direction at a time. Later experiments will determine how well the simultaneous bidirectional signaling will work in this arrangement.

The Primary receiver (U3) is a very high-sensitivity comparator, capable of detecting logic signals less than 10mV. It is used to detect and amplify the high-speed signal coming from the DUT, and to create the differential logic signals needed for the Xilinx Kintex-7. This comparator (form Hittite) is intended for applications up to 10Gbps.

The Secondary receiver (U4) is also a very high-sensitivity comparator, capable of detecting logic signals less than 10mV. However, unlike the primary receiver, the secondary receiver has an edge-triggered “clock” input that defines an extremely short sampling window (~5ps). The clock signal is produced in the Kintex-7 FPGA, and is optionally delayed using the Delay IC (U1) with 5ps resolution and 5ns range. The secondary receiver is intended to “sample” the signal at the junction and thereby monitor either the Driver output or the DUT output signals. This sampler may gather test information without interrupting the normal functional test sequence. Since it works in the background, we call it a “shadow” sampler.

Some minor modifications to this PE4 design were later made in the Rev2 version that is described later, and is currently under evaluation. The main change (form Rev1 to Rev2) was to provide more flexibility for changing the termination voltage for signals coming from the Driver. This was done to better match the Driver CML output loading conditions. See the later section for some information on Rev2.
The Main Board serves as a central controller for the high-speed signals provided to, and obtained from the DUT. Based on the Xilinx Kintex-7 FPGA, it can produce up to 16 high-speed differential inputs and 16 high speed differential output, along with several hundred “low” speed signals (up to 1Gbps). In this section we demonstrate the ability of the Main Board to produce high-quality signals at 3.2Gbps (the target speed for our project) and even as high as 12.5Gbps (well above target). The signals shown below were obtained by programming the Kintex-7 GTX transmitters to output a pseudo-random bit sequence at the desired frequency. These signals were transmitted from the FPGA to the SamTec connectors using 100-Ohm differential striplines within the Main Board. After passing through the SamTec and its mating connector, the signals traveled through about 4 or 5 inches of microstrip transmission line on a Rogers4350 PCB that we had designed for making convenient connection to the high speed signals. This “Breakout” board connects the signals to high-quality SMA launches for convenient connection to test instruments (we used GORE SMA cables and an Agilent 86100D 50GHz sampling Oscilloscope). See the photograph of this arrangement in Fig.30. The resulting signals at 3.2Gbps, 6.4Gbps, 10.0Gbps, and 12.0Gbps are shown in Figures 31, 32, 33, and 34, respectively.

Fig.30 – Photograph of the high-bandwidth “Breakout” board mounted on the Main Board.
Fig. 31 – Data eye diagram at 3.2 Gbps from the Main Board (156ps/div).

Fig. 32 – Data eye diagram at 6.4 Gbps from the Main Board (78ps/div).
Fig. 33 – Data eye diagram at 10.0 Gbps from the Main Board (50ps/div).

Fig. 34 – Data eye diagram at 12.0 Gbps from the Main Board (42ps/div).
(11) **PE4 (Rev1) Performance** (1.6 Gbps to 5.0 Gbps)

(a) Driver Optimization using Pre-emphasis adjustments

In the following figures we show how adjustment of the Pre-emphasis parameters can be used to optimize the Driver signal wave-shape and to “open up” the data eye. In general, pre-emphasis is added to boost the high-speed signal amplitude shortly following logic transitions in order to partially compensate for losses in the transmission line. The Micrel Driver has 5 digital programming bits for adjusting the voltage-amplitude and time-duration of the pre-emphasis added to the nominal signal wave-shape. Three bits select from 0%, 10%, 15%, 25%, or 33% amplitudes, and two bits are used to set the time-duration as 60ps, 100ps, 200ps, or 400ps. For our PE card, we found that only 10% or 15% was needed to obtain optimal performance, and that 100ps or 200ps was the optimal time duration for the frequencies of interest (1 to 5 Gbps). Using too large of amplitude (25% or 33%) results in unwanted “overshoot”. Less than 100ps duration results in undesirable “ringing”. **Figures 35-41** show some examples of 3.2 Gbps with different combinations of pre-emphasis.

![Data eye diagram at 3.2 Gbps from PE4-Rev1](image)

**Fig.35** – Data eye diagram at 3.2 Gbps from PE4-Rev1 (**No pre-emphasis**).
Fig. 36 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 60ps pre-emphasis).

Fig. 37 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 100ps pre-emphasis).
Fig. 38 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis).

Fig. 39 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (15% 60ps pre-emphasis).
Fig. 40 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (15% 100ps pre-emphasis).

Fig. 41 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (15% 200ps pre-emphasis).
(b) Driver Performance at 1.6 Gbps (0-900mV)

In this section we show the ability to program the Driver amplitude (single-ended “swing”). This is accomplished by programming a DAC voltage within about 1 Volt range. The resulting signal swing varies, nearly linearly, from near zero to about 700mV. Figures 42-56 show the change in signal swing from 50mV to 700mV in 50mV increments. For these measurements, the premphasis was set to 10% with 200ps duration, except for 650mV, 700mV and 900mV where no preemphasis is applied.

Fig.42 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 50mV swing.

Fig.43 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 100mV swing.
Fig.44 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 150mV swing.

Fig.45 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 200mV swing.
Fig. 46 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 250mV swing.

Fig. 47 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 300mV swing.
Fig. 48 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 350mV swing.

Fig. 49 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 400mV swing.
Fig. 50 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 450mV swing.

Fig. 51 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 500mV swing.
Fig.52 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 550mV swing.

Fig.53 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 600mV swing.
Fig.54 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (No pre-emphasis) 650mV swing.

Fig.55 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (No pre-emphasis) 700mV swing.
Fig. 56 – Data eye diagram at 1.6 Gbps from PE4-Rev1 (No pre-emphasis) 900mV swing.

For Fig. 56, we reduced the junction series resistance from 26 Ohms to 15 Ohms. This change reduced the resistive-divider effect for the Driver output through the splitter junction, and allowed more of the Driver signal energy to be transmitted through the junction to the oscilloscope. It also appears that the impedance mismatch is reduced by this change. It may be necessary to optimize the junction series resistance experimentally, since it appears that the parasitics of the junction components are not negligible.
(c) Driver Performance at 3.2 Gbps (0-700mV)

In Figures 57-71 the Driver output is shown at 3.2Gbps using 10% and 200ps pre-emphasis, starting with 50mV swing and incrementing in 50mV steps up to 700mV.

**Fig.57** – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 50mV swing.

**Fig.58** – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 100mV swing.
Fig.59 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 150mV swing.

Fig.60 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 200mV swing.
Fig. 61 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 250mV swing.

Fig. 62 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 300mV swing.
**Fig. 63** – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) **350mV swing**.

**Fig. 64** – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) **400mV swing**.
Fig. 65 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 450mV swing.

Fig. 66 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 500mV swing.
Fig.67 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 550mV swing.

Fig.68 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 600mV swing.
Fig. 69 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (No pre-emphasis) 650mV swing.

Fig. 70 – Data eye diagram at 3.2 Gbps from PE4-Rev1 (No pre-emphasis) 700mV swing.
For **Fig.71**, we reduced the junction series resistance from 26 Ohms to 15 Ohms. This change reduced the resistive-divider effect for the Driver output through the splitter junction, and allowed more of the Driver signal energy to be transmitted through the junction to the oscilloscope. There is however, some additional jitter observed using this arrangement, as shown in the Figure.
(d) Driver Performance at 5.0 Gbps (0-700mV)

While the project requirements were to develop the capability to test up to 3.2 Gbps, we designed the system to support even faster rates. The following figures (Fig.72-85) show the PE4-Rev1 Driver output at 5.0 Gbps, which is nearly at its maximum rate. This data rate is limited by the programmable Delay IC used in the PE4-Rev1 design. All the other components (Driver, Comparators, Relay, Connectors, Kintex-7) support data rates above 10.0 Gbps.

Fig.72 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 50mV swing.

Fig.73 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 100mV swing.
Fig. 74 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 150mV swing.

Fig. 75 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 200mV swing.
Fig. 76 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 250mV swing.

Fig. 77 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 300mV swing.
**Fig. 78** – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 350mV swing.

**Fig. 79** – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 400mV swing.
Fig. 80 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 450mV swing.

Fig. 81 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 500mV swing.
Fig. 82 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 550mV swing.

Fig. 83 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (10% 200ps pre-emphasis) 600mV swing.
Fig. 84 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (No pre-emphasis) 650mV swing.

Fig. 85 – Data eye diagram at 5.0 Gbps from PE4-Rev1 (No pre-emphasis) 700mV swing.

(e) Delay Circuit Performance (5ps resolution, 5ns range)
A critical element of the PE4-Rev1 design is the programmable Delay IC, which is used to adjust the skew of the Driver output signal as well as the delay of the Shadow Sampler Clock. For this function we chose the Micrel sy89297 device because it provides the needed dual-channel delay capability with 5ps resolution and 5ns range. It is also a relatively small, low-power device. The manufacturer provides a specified “typical” LSB (resolution) of 5ps, but does not guarantee a minimum or maximum (to account for part-to-part variation). Therefore, this device requires calibration to determine the actual LSB value, from which the 1024 unique delay codes can be estimated. Because of the internal logic structure within this IC, there are 10 independent parameters that are related to the LSB by integer powers of 2. In principle, any of these can be measured and then used to estimate the LSB. We used the largest parameter, representing a delay of $2^9 \times 5\text{ps} = 2560\text{ps}$ approximately. The measured delay is then divided by 512 to find the LSB value (near to 5ps). When we calibrated the Driver signal delay path using this approach, we estimated the LSB to be 4.42ps. Then, using this value, we can estimate the other 9 delay parameters, such as 2x4.42ps, 4x4.42ps, 8x4.42ps, etc. Knowing all 10 parameters, we can accurately predict the actual delay for each of the 1024 delay codes. Some examples are shown in the table below.

<table>
<thead>
<tr>
<th>Delay Code</th>
<th>“Typical” Delay (using LSB=5ps)</th>
<th>Predicted Delay (Calibrated)</th>
<th>Measured Delay</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000001</td>
<td>5</td>
<td>4.42</td>
<td>4.00</td>
<td>+0.42</td>
</tr>
<tr>
<td>0000000010</td>
<td>10</td>
<td>8.84</td>
<td>9.10</td>
<td>-0.26</td>
</tr>
<tr>
<td>0000000100</td>
<td>20</td>
<td>17.68</td>
<td>17.40</td>
<td>+0.28</td>
</tr>
<tr>
<td>0000001000</td>
<td>40</td>
<td>35.36</td>
<td>36.80</td>
<td>-1.44</td>
</tr>
<tr>
<td>0000010000</td>
<td>80</td>
<td>70.72</td>
<td>73.70</td>
<td>-2.98</td>
</tr>
<tr>
<td>0001000000</td>
<td>160</td>
<td>141.44</td>
<td>142.10</td>
<td>-0.66</td>
</tr>
<tr>
<td>0010000000</td>
<td>320</td>
<td>282.88</td>
<td>286.30</td>
<td>-3.43</td>
</tr>
<tr>
<td>0010000000</td>
<td>640</td>
<td>565.75</td>
<td>572.00</td>
<td>-6.25</td>
</tr>
<tr>
<td>0100000000</td>
<td>1280</td>
<td>1131.50</td>
<td>1132.00</td>
<td>-0.50</td>
</tr>
<tr>
<td>1000000000 (calib code)</td>
<td>2560</td>
<td>2263.00</td>
<td>2263.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

The residual error (after calibration) ranges from 0ps (for the calibration code) to -6.25ps, which is well within the manufacturer’s specification of ±15ps for INL.
A plot of the entire range of delay codes, showing some sample codes is provided in Figure 86. Here the decimal value of the binary delay code is shown on the X axis, and the delay value (in picoseconds) is shown on the Y axis. The Red curve plots the actual measured delay, and the Blue curve plots the expected delay if the LSB is set at the "typical" value of 5ns.

**Fig.86** – Measured delay (Red) compared to “Typical” delay (Blue).

**Fig.87** – Demonstration of ~5ps delay step resolution (5ps/div).
Due to the limitations of the Micrel CML output Driver, we decided in mid-April to make a second revision to the PE4 design. The main change (from Rev1 to Rev2) was to provide more flexibility for changing the termination voltage for signals coming from the Driver. This was done to better match the Driver CML output loading conditions. Modifications are shown in Figure 88.

Figure 88 – Block diagram of the REV#2 PE4 circuit (only Channel #1 is shown).

Figure 89 – Photograph of the REV#2 PE4 circuit (Channel #1 is assembled).
Channel Monitoring – Signal Reconstruction Algorithm

1. Multi-Channel Monobit Receiver Architecture

Figure 1 shows the proposed multi-channel monobit receiver architecture. For each device under test (DUT), a single user I/O port receives a sequence of data logic from a clocked-comparator. The clocked-comparator converts the input test signal into two logic-levels (‘High’ or ‘Low’). A time-variant threshold signal is applied to the clocked-comparator and compared with the input test signal. If the input test signal is greater than the threshold signal, the output of the clocked-comparator is ‘High’. Otherwise, the output is ‘Low’. The threshold signal comes from an oscillator which generates a sin wave or from a DAC which provides any arbitrary waveform. The threshold signal can be either synchronized or asynchronous with the system. Figure 2(a) shows the asynchronous case that the DUTs and the threshold signal are not synchronized with the main system. In this case, the back-end algorithm performs extra processes: estimation of the fundamental frequency of the input test signal (due to asynchronous DUTs) and digital phase adjustment of the threshold signal (due to asynchronous threshold signal).
In this report, we focus on the signal reconstruction and its performance on a single channel. The input test signal is assumed to be AC-coupled. The amplitude and the frequency of the threshold signal are supposed to be very precise. For example of the asynchronous case, the oscillator is assumed to generate an ultra-clean sinusoidal wave. Therefore the amplitude and the frequency of the threshold signal are known in precise but the phase is unknown. The amplitude of the threshold signal is set to be large enough so that it covers the peak-to-peak of the input test signal. In addition, the input test signal is wide-sense-stationary (WSS) which guarantees the spectrum of the signal does not vary with respect to time. This condition allows frequency estimation of the input test signal.

Before illustrating the proposed algorithm, the notation of this report is denoted as followings:

\[ f_s = \frac{1}{T_s} : \text{sampling clock frequency (Hz)} \]

\[ f_{th} : \text{threshold signal frequency (Hz)} \]

\[ x(t) : \text{input test signal} \]

\[ x[n] = x(nT_s) : n\text{-th sample of input test signal} \]

\[ v(t) : \text{threshold signal} \]

\[ v[n] = v(nT_s) : n\text{-th sample of threshold signal} \]

\[ d[n] : n\text{-th sample of clocked comparator output} \]

\[ \hat{f}_d : \text{discrete fundamental frequency of input test signal} \]
Figure 2. Conceptual diagram of a clocked-comparator.

Figure 2 shows the conceptual diagram of a clocked-comparator. The input test signal \( x(t) \) and the threshold signal \( v(t) \) are sampled at the rising edge of the sampling clock. The two sampled points are compared and the output logic level is determined as following:

\[
d[n] = \begin{cases} 
1, & \text{if } x[n] \geq v[n] \\
0, & \text{if } x[n] < v[n]
\end{cases}
\]

Note that ‘1’ and ‘0’ represent a voltage level, not the actual voltage value.

Figure 3. Flow of the signal reconstruction algorithm. The dotted boxes indicate the extra processes for asynchronous case.

The signal reconstruction involves several steps as shown in Figure 3. The dashed boxed denote the extra processes for the asynchronous case.

\textit{Step 1: Estimate } \tilde{f}_d \text{ (Asynchronous case)}
Accurate discrete frequency estimation using the frequency shifting technique is presented. When a sampling clock and an input signal is not synchronized and the sampling is incoherent, the discrete frequency estimation is not obvious due to spectral leakage. We use the frequency shifting method to improve the resolution of the discrete fundamental frequency ($\hat{f}_d$) of the input test signal. To compute $\hat{f}_d$, the ground reference is applied to obtain a two-bit resolution samples of the input test signal.

When the input test signal is synchronized with the system, this step is redundant. Since the synchronization guarantees a fixed relationship between the input signal and the sampling clock (no drifting), the nominal frequency is accurate enough.

*Step2: Locate samples*

![Diagram](image)

Figure 4. (a) A square wave and a sinusoidal threshold signal are plotted with the output logic of the clocked-comparator over the time axis. (b) The samples are remapped over the estimated $\hat{f}_d$. The circles indicate the flipping points. (c) More samples are plotted.

In this step, the amplitude-resolution of the input test signal is improved by comparing with the threshold signal. Though the phase of the threshold signal is adjusted in the next process (“digital phase adjustment”), we assume the phase is known for the following example. Figure 4(a) shows an example...
of a square wave input test signal and a sinusoidal threshold signal. During the input test signal is greater than the threshold signal, the clocked-comparator generates the logic ‘1’ samples. The grey downward-pointing triangles represent the samples of logic ‘1’. Similarly, the black upward-pointing triangles represent the samples of logic ‘0’. Note the logic of the samples is flipped when the threshold signal crosses over the input test signal. The dotted-circles emphasize the boundary points of the flipping logic. After the $\hat{f}_d$ is estimated by the previous step in asynchronous case or determined by the nominal frequency relationship between the input test signal and the threshold signal in synchronous case, the discrete time of $v[n]$ is found over the fundamental period as shown in Figure 4(b). The time location of the k-th sample is remapped within the fundamental period as

$$t_d[k] = \text{mod}\left(k, N/\hat{f}_d\right)$$

where $N$ is the total number of samples. The discrete amplitude of $x[n]$ is evaluated by the boundary points of logic ‘1’ and ‘0’. In the example, the nine dotted-circles which indicate the boundary points imply the shape of the input test signal. As the number of samples increases the time and amplitude resolution of the boundary points increases as shown in Figure 4(c).

**Step3: Digital phase adjustment**

![Figure 5. Incorrect digital phase of the threshold signal produces errors.](image)

The digital phase adjustment is achieved by iteratively changing the phase of the virtual threshold signal. In asynchronous case, the phase of the true threshold signal is unknown. If the phase of the virtual threshold signal does not match the phase of the true threshold signal, the mismatch error occurs at the boundary points. Figure 5 shows the previous example with an incorrect digital phase offset. Comparing
with Figure 4(b), the incorrect digital phase of the virtual threshold signal introduces the error from the boundary points to the input test signal as shown in Figure 5(a). The best phase of the virtual threshold signal is selected when the following cost function is minimized.

\[
\text{cost} (\phi) = \int_0^{\hat{f}_d} \text{env}_0(u, \phi) - \text{env}_1(u, \phi) du
\]

where \(\phi\) is the digital phase of the virtual threshold signal and \(u\) is a dummy variable in \([0, \hat{f}_d]\). The function \(\text{env}_0(u, \phi)\) is the envelope of the logic ‘0’ samples and \(\text{env}_1(u, \phi)\) is the envelope of the logic ‘1’ samples. The shape of the function \(\text{env}_0(u, \phi)\) and \(\text{env}_1(u, \phi)\) depends on \(\phi\).

**Step 4: Evaluate logic boundary**

The final step of the algorithm is to evaluate the logic boundary. The logic boundary roughly indicates the input test signal. The envelope of the logic ‘0’ (logic ‘1’) samples is obtained by choosing the minimum (maximum) value in a given time window. Figure 6 shows an example to evaluate the logic boundary.

2. Frequency Selection

Given the input test signal, the frequencies of the sampling clock and the threshold signal determine the resolution of the reconstruction. The frequency relationship between the sampling clock and the input test signal impacts on the time resolution of the reconstruction. Suppose the frequency relationship is as following:

\[
f_s = \frac{c}{d} f_d
\]
where $c$ and $d$ denote two non-negative integers and $f_s$ is the frequency of the sampling clock. The fundamental frequency of the input test signal is denoted by $f_d$. If the integers $c$ and $d$ are co-prime, then the time-resolution of the reconstruction is

$$\text{res}_t = \frac{1}{f_d \cdot c}$$

This equation implies that the sampling frequency does not require being fast. To achieve a high time-resolution with a low sampling frequency, we only need to find a large integer $c$ and a large integer $d$ which satisfies a co-prime relationship and a small $c/d$.

Figure 7 shows an example when $c$ is 17. In this example, $d$ is chosen to be 39. Therefore, the sampling clock is $17/39$ of the input test signal frequency. The input test signal is a square wave without noise. Note that the number of the reconstruction grids is 17 and the grid equally divides the fundamental period of the input test signal.

Figure 1. Signal reconstruction of $c=17$.

3. Hardware Evaluation

The measurement setup is shown in Figure 8. HMC874LC3C from Hittite is a clocked-comparator supporting 20Gbps clock operation and 10GHz input bandwidth. The output waveform of the clocked-comparator is captured by an FPGA board. In the hardware measurement, only asynchronous case is included.
Two types of input test signals are measured: a digital square wave and a sin wave. The high-speed digital square wave of 1.6GHz frequency is generated. For the sampling clock,

\[
f_s = \frac{163 \cdot 223}{977 \cdot 983} \text{GHz} = 60.557 \text{MHz}
\]

is chosen as the sampling frequency. The frequency of the sinusoidal threshold signal is chosen as

\[
f_{th} = \frac{163 \cdot 223 \cdot 991^2}{977 \cdot 983 \cdot 997^2} \text{GHz} = 59.8303 \text{MHz}
\]

Similarly, the frequencies of the sampling clock and the threshold signal are chosen for the 1GHz sin wave with different prime numbers as followings:

\[
f_s = \frac{211 \cdot 223}{977 \cdot 983} \text{GHz} = 48.9937 \text{MHz}
\]

\[
f_{th} = \frac{211 \cdot 223 \cdot 991^2}{977 \cdot 983 \cdot 997^2} \text{GHz} = 48.4057 \text{MHz}
\]

To cover the input test signal, the peak-to-peak of the threshold signal is set to 600mV. The threshold signal is applied externally without synchronization. Figure 9 shows the cost function over the adjusted digital phase in the case of the square wave. While the phase is swept over the range [-180, 180] degree, the cost function is computed. The cost function has the minimum value at the phase of -124.4 degree. The threshold signal of phase -124.4 degree produces least mismatches on the logic boundary.

Figure 10 and Figure 11 compare the input test signal and the reconstructed signal. The envelopes of the logic ‘1’ and logic ‘0’ are plotted together. The result of the square wave input test signal shows the reconstructed signal has a minor distortion on the waveform. However, the reconstruction accurately
captures the edge of the square wave and the amount of jitter is very close to the signal measured by the high-speed oscilloscope. In sin wave measurement, the proposed algorithm reconstructs the waveform with better accuracy.

The proposed algorithm will be implemented in an FPGA. For the channel monitoring purpose, the incident waveform is generated by a high-speed I/O in the same FPGA. In addition, the threshold signal will be provided by a low-speed programmable DAC. Thus, the system is synchronous and therefore the estimation of the fundamental frequency of the input signal and the digital phase adjustment are not necessary in Figure 3. For simplicity, the threshold signal will be a step-waveform. In order to evaluate all the grids of the reconstruction, the level of the threshold signal holds for \( \frac{c}{f_s} \) seconds. For example of c=17 in Figure 7, the threshold signal will be look like Figure 12. Note that the threshold level stays even until it covers all 17 grids.

Figure 3. Cost function and remapped samples.
Figure 4. Square wave input test signal and reconstruction.
Figure 5. Sin wave input test signal and reconstruction.

Figure 6. Threshold level holds until it covers all number of girds.
For the characterization of interconnects, the test channels including transmission lines, connectors and cables are considered DUTs where electrical faults present. The transmission line will have continuously varying impedance discontinuities along its length, the presence of which will depend on the type of interconnect under consideration, the trace topology and geometry and the fabrication results. Depending on all these factors, there will be both minor impedance variations as well as large discontinuities.

For a continuously varying impedance profile as shown in the figure above, any step voltage launched will suffer reflections and transmissions at each impedance mismatch. The dotted lines signify the higher order reflections. At each sampling instant, the voltage measured is the superposition of the incident step voltage and the reflections reaching the incident port.

The detailed impedance reconstruction procedure is shown in Appendix – Enhanced Spatial Resolution of Time-domain Reflectometry.
The proposed time-domain reflectometry (TDR) algorithm consists of four functional blocks: (1) Discrete-time Sample Index Generator, (2) Threshold Voltage Generator, (3) Sample Time Mapping, and (4) Impedance Estimation/Fault Detection. These algorithmic blocks are implemented in an FPGA except (4) Impedance Estimation/Fault Detection (because of the complexity of the algorithm), which is implemented in an external numerical simulator (i.e. MATLAB).

The signal driver associated with the FPGA high-speed I/O continually generates a low-frequency clock signal, which is used as a TDR stimulus traveling on the test channel and being reflected when impedance mismatch presents on the channel.

Details of the implementation of the algorithm are shown in the figures below.
In this experiment for time-domain reflectometry, the Micrel driver chip of the pin electronics board is used as a test stimulus generator, whose driving performance is separately shown in the previous sections. Since the driver output is AC-coupled in the current version of the pin electronics board, the shadow sampler, which captures both the test stimulus generated by the driver and any reflected signals from impedance discontinuities on the test channel, is able to see the upper half of the signal above the ground level. However, we are still able to see reflected travelling signals on the reconstructed shadow sampler waveform. We plan to continue time-domain reflectometry experiments with the revised pin electronics board to overcome such a limited signal acquisition voltage range. The signal acquired by the shadow sampler is digitally transmitted to the FPGA and processed by the TDR algorithm that includes the signal reconstruction algorithm with offset frequency sampling.
We use a test channel board that emulates test channel faults. To emulate a termination resistance fault, we use a test channel that is terminated with a discrete resistor to ground. This termination resistor is considered an impedance discontinuity as compared to the channel characteristic impedance, which is 50-ohm (or slightly higher due to manufacturing imperfection). The frequency of the TDR test stimulus we use in this experiment is ???-MHz. The frequency of the test stimulus can be lowered so the voltage level after the rising edge of the test signal sustains longer when a test channel that needs to be monitored is longer. This is because it would take longer for an impedance discontinuity located farther away from the test stimulus generator or monitoring system (or shadow sampler) to generate a reflected signal and the shadow sampler observes the signal.

Figure a) shows a test result with a termination resistance value of 45-ohm (10 percent mismatch). The voltage level in the time range that corresponds to the test channel between the SMA connector and the termination resistor is slightly higher than the prior voltage level. This is because the channel characteristic impedance of the test board is designed slightly higher than 50 ohm. When the test signal
hits the termination resistor which is not 50 ohm, a signal reflection occurs and lowers the voltage level. By observing the voltage level and decomposing reflected signals, we detect the 10-percent impedance discontinuity on the test channel. Figure b) shows a test result with a termination resistance value of 40-ohm (20 percent impedance mismatch) which is located at the same position as in the previous experiment.

In this experiment setup, a 5-ohm series resistance located in the middle of the test channel is used. Such a lumped resistor contributes the increase of the channel impedance to 55 ohm when looked into the resistor from the signal monitoring system. The associated test result is shown in Figure a), where the impedance discontinuity is not clearly shown because the resistance is located too close to the input SMA connector of the channel test board. Since another impedance discontinuity presents at the input of the channel test board due to manufacturing imperfection (note: the test channel characteristic impedance measures above 52 ohm), the resistance fault located very close to the prior fault is hardly noticed when observing the measured waveform. Such a limitation can be overcome by using a better...
matched test channel board or locating the lumped series resistance farther away from the test channel board input connector.

Figure b) shows a test result obtained with a test channel that contains a 5-pF parallel capacitance located in the middle of the test channel. Since a 5 pF capacitance is considered a relatively large impedance discontinuity, a reflected waveform due to the capacitive mismatch is shown as a large voltage droop and can be easily detected as shown in the figure.

**Online Eye-opening Monitoring (with Shadow Sampler) – Hardware Evaluation**

The eye-monitoring algorithm described in this section uses the shadow sampler data without using the primary sampler data. In cases where the test digital pattern is balanced (3.2-Gbps PRBS is used in this evaluation setup), the shadow sampler output signal will be also balanced when the test signal is sampled near the middle of the data eye. However, the shadow sampler data will not be balanced when the signal is sampled outside the data eye vertically or horizontally. By observing the statistical data of
the shadow sampler output signal for various sampling time/amplitude points, the test signal eye opening can be monitored.

A 3.2-Gbps PRBS signal is generated by an external digital signal generator and fed to the pin electronics board. The signal is sampled by the shadow sampler and processed in the FPGA without using the primary sampler data. Since the proposed eye-diagram reconstruction method uses a statistical analysis assuming that the test signal being monitored is well balanced in terms of statistical distribution of ones and zeros in the digital bit pattern. The shadow sampler clock speed used in this experiment is ???-MHz, but the frequency can be adjusted for purpose of changing the time resolution of the reconstructed eye diagram.

In the figures above, the reconstructed time resolution is 30 ps, and the reconstructed amplitude resolution is 5mV. 60 samples are collected per slot in the reconstructed eye diagram and the distribution of ones and zeros is indicated with a color map. The slot highlighted with dark red indicates that all the 60 samples are ones, and the slot with dark blue shows that all the 60 samples are zeros. Such an extreme distribution indicates the data eye is closed in the associated region. However, the Final Report – Test Module Hardware
middle of the data eye shows the sampling slots highlighted with green, which indicates that the sampled data set is well balanced. Figures a) and b) show the reconstructed eye diagram for 31-bit and 1023-bit pseudo-random bit sequences respectively.

In the figure below shows the enhanced time resolution of the eye diagram, which can be obtained by changing the shadow sampler clock frequency to ??? MHz. The enhanced time resolution is round 8 ps, and more vertical samples are obtained per unit interval of the eye diagram.

Channel Monitoring BOST
- Online Eye-opening Monitoring - Measurement Data (shadow sampler only)

Fig. a) Test stimulus: 31-bit PRBS (320mV to 460mV swing)  
Fig. b) Test stimulus: 31-bit PRBS (320mV to 460mV swing)

Note:  
1) 64 samples per grid obtained from the shadow sampler with variable threshold voltage levels.  
2) Test data pattern (generated from the external signal generator) must be "balanced."  
3) Time Resolution: ~8 ps, Voltage Resolution: ~5 mV

Online Eye-opening Monitoring (with Primary & Shadow Sampler) – Algorithms

The online eye monitoring algorithm that does not use the primary sampler data requires an assumption of balanced test signals. To remove such a limitation, we use the primary sampler data as well as the shadow sampler data in this section. The primary sampler data are obtained by the FPGA GTX I/Os and internally deserialized for digital signal processing as shown in the figure below. The shadow sampler data are acquired by FPGA user I/Os and also internally deserialized. The key idea in this approach is that the data rates of these two signal path need to be the same after deserialization. If a particular shadow
sampler data channel (after deserialization) is associated with the mid-point of data eye, the data channel should not contain any bit errors and is perfectly matched with the primary sampler data channel (after deserialization). However, the shadow sampler data channel that is associated with the edge of the data eye, it should contain bit errors and cannot be matched with the primary sampler data. By using this correlation analysis between the shadow and primary sampler data, an eye diagram can be reconstructed without the assumption of a balanced test signal.

In the online eye monitoring experiments illustrated below, we use a test setup summarized below.

1. A 3.2-Gbps PRBS-31 test stimulus is generated by the internal driver (AC coupled output).
2. The output of the PE board is terminated with a 50-ohm resistor to ground by the oscilloscope.
3. The shadow sampler is used to capture the driver signal with a sampling speed of 480 MSPS.

The proposed data correlation analysis method that uses both the primary and shadow sampler data is implemented in a numerical simulator, not in an FPGA, because of the programming complexity. The driver signal is AC-coupled and the shadow sampler is able to see the upper half of the waveform.

Final Report – Test Module Hardware
Final Report – Test Module Hardware
Figure (a) above shows a reconstructed eye-diagram of the test signal, which is a PRBS-31 signal generated by the internal driver. The reconstructed eye-diagram consists of three sampling time points and each sampling grid contains 1000 samples (the number of sample points can be reduced if needed). When compared to the oscilloscope measurement shown in Figure (b), the reconstructed eye-diagram shows a good correlation.

**Channel Monitoring Bandwidth (5-GHz) – Hardware Evaluation Results**

In this hardware measurement, we evaluated the system bandwidth of the channel monitoring sub-system of the pin electronics board (with AC-coupled drivers). An external RF signal generator (Agilent 81300B) was used and generates an RF test stimulus with swept frequencies, which is fed to the input of the pin electronics board as shown in the figure above. The test stimulus is acquired by the channel monitoring sub-system especially the shadow sampler. The test signal travels over various components, which may help reduce the overall bandwidth of the system, such as SMP connectors, mechanical relays and power splitters. The measured system bandwidth is slightly above 5-GHz as shown in the next page. The test stimulus frequencies chosen to be used in this test setup are 454-MHz, 3021-MHz, 5135-MHz.
and 6041-MHz. The sampling frequency of the shadow sampler is set to ???-MHz and a time resolution of 20 ps is obtained.

We will repeat the measurement with a revised pin electronics board with DC-coupled drivers. Since the new board does not contain AC-coupling capacitors in the power splitter, which may function as a low-pass filter, we anticipate a slightly wider system bandwidth of the new pin electronics board. We plan to use 100-MHz, 1.5-GHz and 3.0-GHz sine stimuli to test the system bandwidth as discussed in the earlier phase of the project. However, the shadow sampler does not equipped with signal reconstruction capabilities for these particular frequencies. For this reason, we will also conduct the experiments with other frequencies located near those three selections to show the reconstructed waveforms.

As shown in the figure above, the RF test stimulus with a frequency of 5135 MHz is attenuated by -3dB as compared to the signal with a frequency of 454-MHz. According to the experiment results, the channel monitoring system bandwidth is slightly above 5 GHz. As previously indicated, the revised pin electronics board may show wider system bandwidth. The experiment with the revised board is currently being performed and the measurement results will be included in the final report.

Final Report – Test Module Hardware