

# A HIGH-THROUGHPUT 5 GBPS TIMING AND JITTER TEST MODULE FEATURING LOCALIZED PROCESSING

Mohamed M. Hafed, Antonio H. Chan, Geoffrey Duerden, Bardia Pishdad, Clarence Tam, Sebastien Laberge, Gordon W. Roberts

DFT MicroSystems Canada Inc.

1450 City Councillors Street, Suite 1000, Montreal, Quebec Canada H3A 2E6  
{firstname.lastname}@dftmicrosystems.ca, Tel: (514) 878-8271, Fax: (514) 878-9338

**Abstract** – A compact timing and jitter test system that leverages custom integrated circuit measurement methods and localized test result processing is presented. It consists of five timing measurement units (TMU) and five timing generation units (TGU) as well as hardware digital processing units for local test result processing or parameter extraction. The TMU channels rely on a component-invariant vernier-delay measurement circuit and the TGU channels rely on linear programmable delay circuitry. The system supports both LVDS and CML high-speed digital interface standards at rates of up to 5 Gbps. This solution occupies 3"x4" of board area, which makes it suitable for placement on the DUT-board. It has a relative delay generation resolution of 3 ps at 5 Gbps, and is capable of autonomous, platform-independent pass-fail testing.

## I. Introduction

Low voltage signaling techniques and the increasing reliance on analog methods within predominantly digital integrated circuits (IC's) pose particular challenges during the production test phase. The increasing reliance on analog techniques means that such IC's are increasingly sensitive to process and environmental conditions and that any screening methodology should accommodate this. One example of an analog metric is jitter, which is no longer confined to high-performance communications devices but is also starting to hinder the development of traditionally forgiving devices such as disk-drive ASICs, graphics card processors, and consumer processors. Recent attempts to address the test of such mixed-mode ASICs have focused on embedding some of the analog test capability within the device being tested. This certainly enhances observability and testability of critical deeply embedded cores but fundamentally falls short of covering the primary input/output interface ports. For such ports, some level of external test capability is required. For example, a divide-and-conquer approach involving the interaction between instruments internal to the DUT and ones external to it can be used [1].

In this paper, we focus on the test of primary input/output ports, and we propose a compromise between internal test (or DFT) and external rack-and-stack test. We do this by putting some test electronics on the load board as shown in Figure 1. This concept is not new. Most modern load boards have signal conditioning components on them. Additionally, when significantly more capability is required, much more complex components have been placed on the load boards in recent years [2]-[5]. We adopt the same approach as in these methods because of the merits of proximity to the device under test. However, we depart from previous approaches to DIB-based test solutions in the components we integrate. Specifically, our solution constitutes a complete DSP-based jitter test instrument that is designed to mount directly on ATE platforms. With its size, measurement capability, and localized processing engines, it is designed to achieve a radical improvement over existing production test solutions for jitter. This is elegantly achieved through integration of the key features into custom ICs and the reliance on powerful and flexible hardware computational engines for local test parameter extraction.

The rest of this paper is organized as follows. Section II describes the functional model of our proposed system and partitions it into three main components. Section III delves into some of the design considerations required using such an approach. Section IV summarizes our performance evaluation results, and Section V concludes the paper and describes future activity.

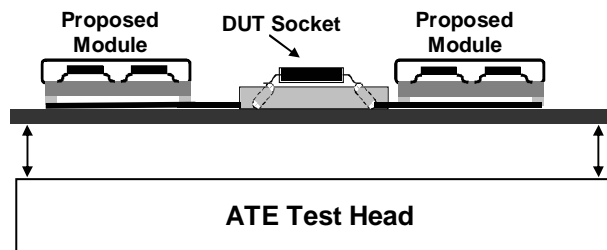


Figure 1 Conceptual diagram of proposed jitter test solution.

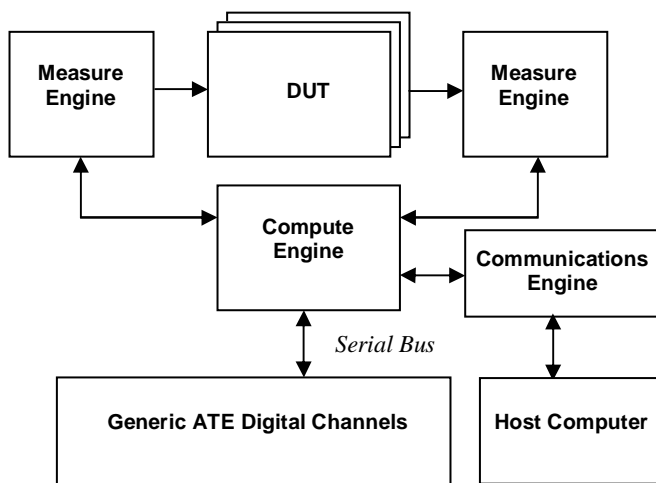
## II. Proposed System Configuration

The proposed system configuration is shown in Figure 2. As can be seen, it consists of three main components: a measurement engine, a compute engine, and a communications engine. It is thus intended to act as a complete stand-alone solution for timing and jitter measurement applications. This solution is envisioned to extend the capability of existing ATE, especially ones that target large DFT-enabled digital ICs and that do not support high-performance jitter measurement. It is also envisioned to extend the lifetime of legacy systems, as the ATE interface requirements of the proposed solution are designed to fall within the range of digital channel cards. As will be described shortly, the integration of the above three engines is achieved by leveraging a combination of custom integrated circuit technology and off-the-shelf semiconductor components.

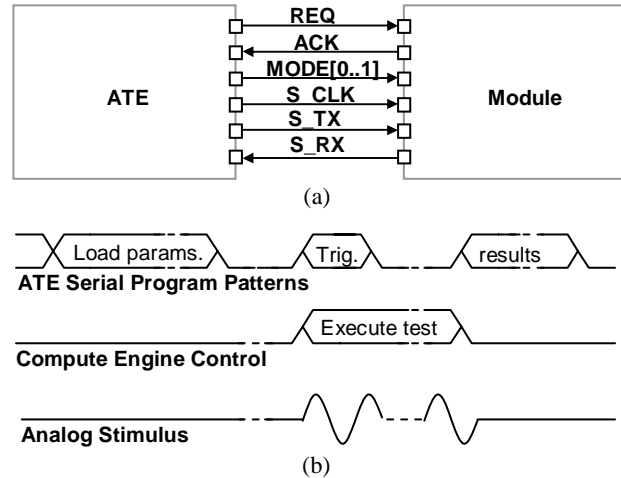
### A. Localized Compute Engine

A key feature of the proposed solution is local test data processing. Incorporating such capability greatly reduces ATE bandwidth and storage requirements while still providing high-speed multi-site test capability. This is an important differentiator from augmenting ATE with specialized and/or high-performance equipment, such as oscilloscopes, because the latter usually suffer from excessive test times [6]. For example, it is not unreasonable for an oscilloscope-based jitter measurement to take >30 sec for 4096 samples. As will be seen in Section III, customized jitter statistics extraction logic could operate at much faster rates.

It is important to note that the compute engine in this context is also responsible for controlling the overall



**Figure 2** Functional diagram of proposed module. Onboard computational power is used to eliminate data transfer bottlenecks and to extend the capability of the ATE platform.



**Figure 3** (a) Hardware representation of serial interface between tester and module. (b) Sample sequence of operations. Parameter loading needs to be performed only once.

operation of the proposed module. In particular, it is responsible for controlling tests, calibrating the measurement engines, and initiating communications with the host ATE platform. This last function is done through the communications engine, which is described next.

### B. Communications Engine

The communications engine consists of two paths. The primary communications path is a direct serial connection to the host ATE. Specifically, a set of 9 dedicated pins connected to the channel cards of the ATE is required for the proposed solution. Thus, from a test program and pattern file point of view, the module looks like a DUT. However, unlike a DUT, the reserved pins for our solution are used to implement a simple asynchronous serial bus protocol for passing commands from the test program to our module. For example, this bus can be used to pass stimulus parameters to the on-board measurement engines, to set pass-fail test limits for the localized processor, or to globally interrupt the operation of the module. Figure 3 illustrates the proposed wiring for this serial communications interface as well as sample timing diagrams. Not shown in the figure are two shortcut wires, Trig1 and Trig2, that are used to initiate tests or stop tests. Further details are described in Section III.

As for the secondary communications path, it relies on a standard Ethernet transceiver chipset in order to interface the proposed solution to a host computer running a simple graphical interface. In our case, we used a Java-based interface because of its ease of development and portability. We recognize that it is unlikely that such a communications interface would be used in production

(because of robustness, speed, and security issues). However, our experience has been that such a secondary path is invaluable as a debug tool in a laboratory environment. In such a setting, the module is capable of operating as a stand-alone instrument, thus facilitating test program development, parameter extraction customization, and initial DUT experimentation without tying up ATE resources.

### C. Measurement Engine

Much of the front-end analog stimulus and measurement capability of the proposed module is performed within integrated circuits that are custom-designed for the application. Short of embedding the test electronics within the DUT, such an approach seems to provide the best opportunity for allowing factory test equipment performance to track the performance of new semiconductor product introductions. The measurement engines within the proposed module have been demonstrated to excite and measure devices at rates of up to 5 Gbps, for which (to our knowledge) there are no available commercial test solutions. More importantly, our experience also suggests that relying on custom IC components is also a great contributor to efforts in miniaturization as much of the signal conditioning, bias, and power circuitry could be embedded within fine-geometry devices.

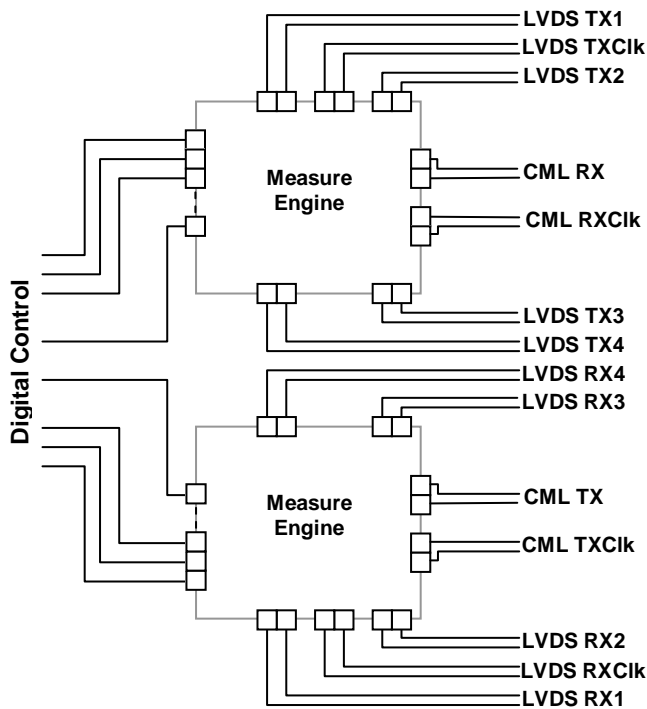


Figure 4 Connection diagram of custom measurement ASIC's.

### III. Design Details and Challenges

This section describes each of the components in more detail. We also focus on some of the design challenges associated with integrating such disparate technologies within a small area.

#### A. Stimulus/Measurement Instrument Design

As was mentioned earlier, the measurement components of our modules were designed within custom integrated circuits. We elected to use two small IC's in order to achieve the best compromise in size, cost, and performance. Figure 4 illustrates the connection diagram for the two IC's. This diagram is best described in the context of testing 4:1 SERDES devices running at 5 Gbps. Specifically, to test the serializer portion of such a DUT, our solution provides four "parallel" LVDS transmit channels (labeled LVDS TX<sub>i</sub>) to excite the device and one CML receive channel (CML RX) to measure the serialized DUT output. Similarly, within the second measurement IC, one CML transmit channel (CML TX) is used to excite the de-serializer portion of the DUT, and four LVDS receive channels (LVDS RX<sub>i</sub>) are used to measure the deserialized DUT outputs. Additional LVDS and CML clock channels are provided to allow for more flexible measurement time-base and trigger control, or to lock the DUT operating frequencies.

It should be noted that the above example is only an illustration of a typical configuration. In general, each of the channels in Figure 4 could be exercised independently, thus allowing the proposed solution to be applied to any jitter test problem at the given rates and interface options. The maximum rate for the CML channels in our IC's is 5 Gbps and that of the LVDS channels is 1.4 Gbps.

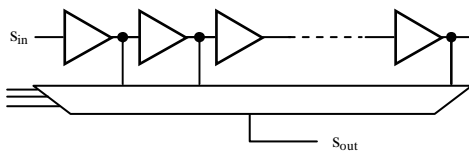
Behind each of the receive channels in Figure 4 lies a compact timing measurement unit (TMU) that is based on a component-invariant vernier delay line measurement method. The circuit for such a TMU has been described in [7] and is illustrated here in Figure 5 for the sake of completeness. The circuit measures the phase relationship between two input signals:  $s_1$  and  $s_2$ . In order to initiate a measurement, a trigger or arm command is applied to the input flip-flops of the circuit. This enables each flip-flop to detect the first incoming edge. Once an edge arrives, each flip-flop starts the triggered oscillator at its output. Because of a predesigned offset-frequency relationship between the two oscillators, their respective phases eventually align. A simple evaluation of the time it takes for the two oscillators to phase align yields an accurate representation of the original phase difference detected at the inputs of the  $s_1$  and  $s_2$  flip-flops. The primary benefit of



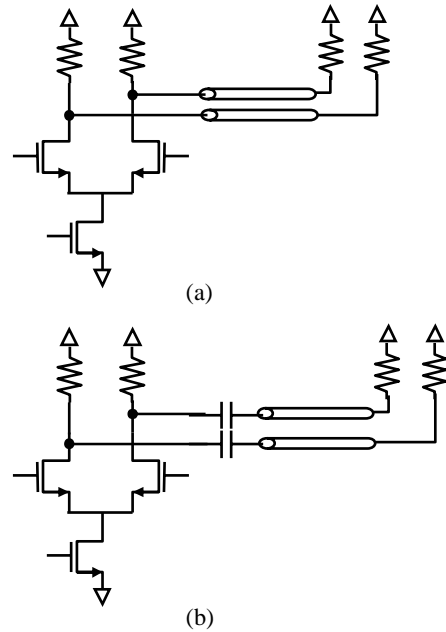
**Figure 5** TMU implementation using a component-invariant vernier delay line. The circuit measures the time delay between  $s_1$  and  $s_2$ , by counting the output of the AND gate.

this method is its compactness and inherent linearity over a large measurement range. For example, in our implementation, a range of 2.4 nsec is adopted without compromising a linearity target of 0.5 LSB (Section IV describes other performance numbers such as resolution and noise floor). Finally, the way the circuit of Figure 5 is integrated into our measurement engine chips connects  $s_1$  to the respective LVDS or CML RX pin and allows  $s_2$  to be either an internal time-base signal or the external RXClk signal.

Similar to the TMU channels, an independent TGU core circuit is integrated behind each of the LVDS and CML transmit channels. The TGU circuit consists of a cascade of programmable delay elements multiplexed into a single output pin. Each delay line consists of 16 delay elements, so the multiplexing logic consists of a simple 16-to-1 decoder circuit. The circuit allows a 4-bit control vector to select any of the 16 output delay settings. Due to the reliance on integration, the decoding circuit as well as the programmable delay elements can be made extremely small and logic operation in the 100's of MHz is easily achieved. The input signal,  $s_{in}$ , to the delay line is derived from an on-chip frequency synthesizer capable of generating a 2.5 GHz, 1.25 GHz, or a 625 MHz frequency. Thus, the TGU channels in this paper are only capable of generating 1,0,1,0 patterns at these frequencies. One known disadvantage of delay-line based circuitry is the



**Figure 6** TGU implementation using delay lines and programmable delay stages.



**Figure 7** Simplified CML driver schematics illustrating (a) DC and (b) AC coupling.

difficulty to achieve high linearity. However, by relying on proper layout averaging techniques and robustly tunable delay elements, good linearity can be achieved. Our delay generation circuitry also contains a vernier mode in which very fine delay steps (as low as 3 psec) can be achieved.

Since our proposed TMU and TGU circuits are no longer intended for embedded applications, a significant effort has gone into designing driver and receiver circuits that satisfy the configuration in Figure 4. The main challenge lies in the ability of our integrated drivers to impress the 5 Gbps signals on to the outside world knowing that there is a finite printed circuit board distance to be traveled between our measurement engine chips and the DUT. However, in keeping with the philosophy of proximity to the DUT, our drivers need not be designed for more than a few inches of distance. The other challenge we faced was the design of termination circuitry to enable both AC and DC coupling to the DUT (Figure 7). Table 1 summarizes the measured electrical specifications of our driver and receiver circuits.

### B. Serial Pattern and Control Considerations

This section describes some of the considerations in the design of the serial programming interface through ATE digital channel pins. This is not an insignificant portion of the system as it ultimately determines usability of the module and it affects the test engineer's program development task. It was mentioned earlier that the serial bus is used to select tests within a sequence, their parameters, test triggers, and test flags. Test parameters

**Table 1** Electrical specifications for drivers

	LVDS	CML
Rise Time (psec)	105	90
Fall Time (psec)	115	95
Differential Swing (mV)	1000	750
DC Bias (V)	1.30	1.5
Termination Resistance ( $\Omega$ )	100	100

could vary from pass-fail jitter limits for the TMU engines or phase selection waveforms for the TGU ones.

The main design consideration for the serial interface is a programming mechanism that does not contradict our bandwidth improvement goals. Clearly, the serial bus of Figure 3 is slow, and it could potentially be slower than (i) connecting DUT outputs directly to the ATE channel card (assuming the ATE is capable of doing the measurement!), or (ii) connecting bench equipment through a secondary bus or similar interface. Instead our serial interface is intended to be used minimally: to set up a sequence of tests and to pass test parameters to the on-board module only once. Similarly it is not intended to read test results unless the test engineer would like to do so. Instead, result comparison is performed locally within our proposed solution. Thus, a typical sequence of events on the serial interface should be as follows:

```

Initialize Test Sequence (ATE)
  Trigger Test (ATE)
  End Test (Module)
  Trigger Test (ATE)
  End Test (Module)
  ...
  Fail Flag (Module)
End Test Sequence (ATE)

```

The last command in the above sequence could be due to a failed test in the sequence. In such a scenario, the rest of the tests in the sequence are not continued. As can be seen, some level of bit comparison within the test engineer’s program is required. However, the important point is that not much data is required to go back and forth across the digital channels. For example, after the *End Test Sequence* command in the above example, the test program is likely to initiate a handler command. Once the new DUT is placed in the socket, the (lengthy) *Initialize Test Sequence* command need not be re-executed.

In order to allow for the flexibility implied by the above example, a simple micro-code central processing unit is

implemented within our compute engine. Within such a framework, it is conceivable to think of the above example as an assembly-language “program” that is executed by this processor. The format we adopt for implementing such program is a N-long 16-bit file. The 16-bit words are used to implement operation codes (OPCODE) for the processor, to pad the input data vectors, and to select the different measurement engine instruments (e.g. LVDS TX1) within the module. Figure 8 shows an example program. In the interest of brevity, the OPCODE mappings are not described here, but some of the typical commands include WRITE, READ, and STOP.

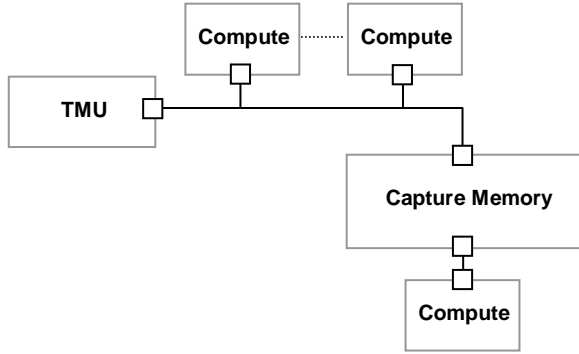
Our initial experience with the above programming sequence suggests that it is intuitive for application by test engineers. Moreover, it lends itself nicely to automation as much of the patterns can be generated automatically either within the ATE software or within third party software. We are currently investigating the automatic generation of such patterns. Additionally, we are investigating various trigger polling and flag generation mechanisms. Currently, two pins (Trig1 and Trig2) are required for test synchronization with the ATE software pattern file. The test engineer is responsible for appropriate coding in order to generate the trigger and monitor the flag. Finally, it should be noted that all test results can be passed to the ATE, completely bypassing any processing on the board.

**C. Algorithm Selection for Local Processing**

In addition to implementing the above micro-code architecture, the compute engine’s main feature is its ability to perform test parameter extraction locally. This is an area of continuing activity as we believe it is the most important feature of the proposed solution. It has the potential of significantly saving test time. For example, in our initial implementation, we implemented algorithms for the computation of typical jitter performance parameters such as RMS, peak-to-peak, mean, and median computation. Since our proposed solution is “focused” to a certain application, much of the general-purpose capability of ATE could be traded for custom computations for the purposes of parameter extraction. The concept of a focused

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	OPCODE															
2	OPCODE															
3	OPCODE															
4	OPCODE															
5	OPCODE															
6	WRI-68															
7	XXXXXXXXXXXXXXXXXXXX															
8	...															
9	XXXXXXXXXXXXXXXXXXXX															
10	STOP															

**Figure 8** Sample Program to load a stimulus file to the proposed module. The initial OPCODE commands include initialization of instruments, memory contents, and state.



**Figure 9** Illustration of dedicated compute engine connections.

instrument has been described in [8], and it suggests that an instrument that is customized to a certain application (e.g. in an embedded scenario) is necessarily simpler than a general-purpose one. We use this observation here for the parameter extraction step.

Consider for example the measurement engine configuration in Figure 4. It is known that any samples coming out of the TMU channels (LVDS RXi or CML RX) always represent phase relationships at the inputs. Thus, custom computation logic could be directly attached to the digital TMU outputs. Contrast this to today's practice in which data is first collected into a digital capture memory, then potentially to a DSP buffer memory, and finally to a general-purpose DSP that lies in the ATE mainframe. Figure 9 illustrates our approach graphically, and it also hints at the possibility of real-time processing of information. Relying on memory is clearly indispensable for complex DSP operations such as the FFT, but many DSP functions can be performed using recursive algorithms. A peak-to-peak measurement, for which the required computation is

$$j_x = \max(x[n]) - \min(x[n]) \quad n = 1..N \quad (1)$$

is computed on the fly within our proposed module using a pair of registers and a comparator. Similarly, a moving average filter can be expressed using the following expression:

$$\mu_x = \frac{\sum_{n=0}^{N-1} x[n]}{N} \quad (2)$$

However, through a simple manipulation, the numerator in (2) can be rewritten recursively as

$$\mu_{nx}[n] = \mu_{nx}[n-1] + x[n] \quad (3)$$

which lends itself to a recursive implementation using an accumulator. It is important to note that the average filter shown here is just one example of a filter that could be implemented recursively, and many others exist. Finally, in order to compute RMS jitter, we note that this quantity is related to the autocorrelation function of the input sequence at a lag of zero. Thus, it can be expressed as

$$r_{xx}[0] = \sum_{n=0}^{N-1} x[n]x[-n] \quad (4)$$

which is just a convolution summation of the incoming samples. The incoming samples in our case are the TMU digital numbers representing the delay relationships being measured.

One common feature of the above examples is that computation of the quantities could complete in zero time (additional to the sample acquisition time) or almost zero time if one considers overhead steps such as division and square-root operations (performed only once per sample set). As will be seen shortly, our system is capable of acquiring samples at 2 MSamples/sec/channel. So collecting thousands of samples and computing their jitter statistics could be performed in a few milliseconds.

Finally, a 4096-point, 20-bit FFT computer is included in the proposed module for extending the parameter extraction capability. For example, the utility of the representation in (4) is that it can be most efficiently calculating using a FFT processor. Our implementation of the FFT processor allows us to perform a 4096-point transform in about 500 $\mu$ s. This is limited by our on-board clock speed, and we are currently working on optimizing the speed further. Also, we are working on implementing longer FFT computations (up to 32768) without significantly compromising throughput.

#### D. Board Design Issues

The proposed module is implemented on a multi-layer board with various plane layers for power and ground. There are three different power domains within the board: digital 3.3 V, 1.8 V, and analog 1.8 V. Two power inputs are delivered to the board through a pair of differential high-speed connectors from Fujitsu. The same connector is used both for power delivery and for signal delivery. Moreover, the digital power supplies are not regulated. Instead, we extensively used power planes and PI filters at the various components (e.g. buffers, memory, digital I/O). The analog power supply is internally regulated.

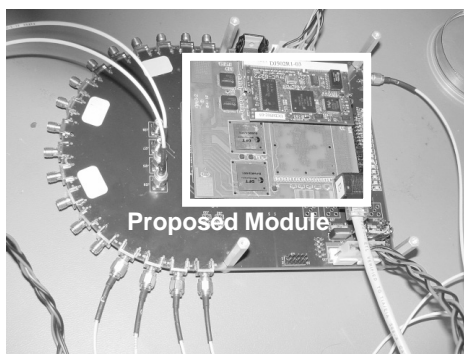
Of particular interest is the delivery of power to the reference clock input of each of our ASIC's. The ASIC's use this reference to synthesize the TGU frequencies and to synchronize with the various components on the board.

In order to avoid any noise accumulation as the reference enters the ASIC, a dedicated power pin was allocated for the pad receiver, and the power pin was isolated from the rest of the board through a series ferrite bead and a parallel capacitor. While it is expected that a single capacitor may not be able to supply enough charge for the switching pad receiver itself, the effect is a deterministic one occurring at the single frequency of the reference. Thus it does not adversely affect the jitter performance.

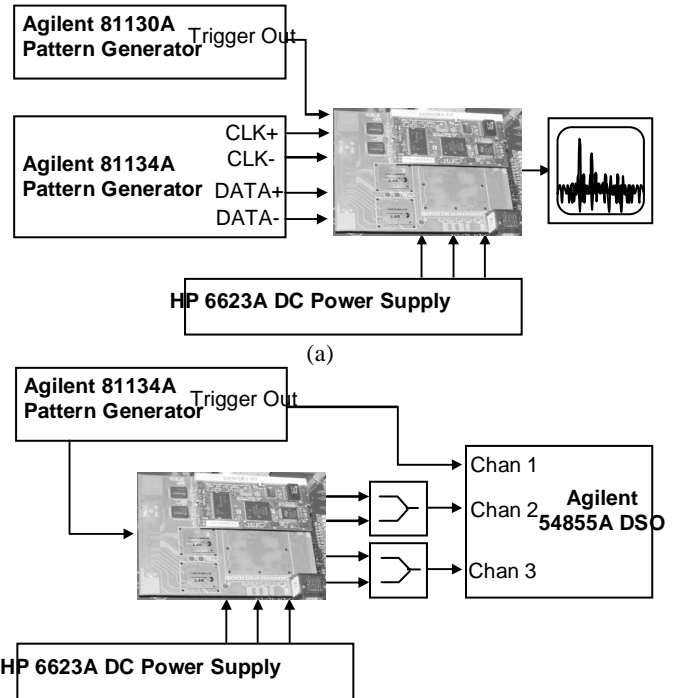
#### IV. Performance Evaluation

Figure 10 shows a picture of our proposed module when mounted on an evaluation board. The evaluation board is not part of the solution but is intended to house the module outside of an ATE environment. In what follows, we present performance results for each class of measurement engine in the module. We start with the TMU channels and then proceed to the TGU engines. Figure 11 illustrates the equipment setup in all subsequent results. As always, we evaluate the performance of our proposed solution using known good instruments. Thus, no sample DUT's are presented in this manuscript.

In the first experiment, the TMU is characterized with a 300 MHz input signal. This rate is chosen in order to allow an input stimulus period that is larger than our measurement range. Figure 12 shows a typical extracted histogram for such signal, and Figure 13 shows the transfer characteristic of our TMU. Several remarks should be made about these results. First, the hardware resolution of the device is about 20 psec. This can be inferred from the code transition points in Figure 13, and it explains the coarseness of the extracted histogram. In other words, Figure 12 is the measurement result of a 1 psec signal (Agilent 81134) using a time-quantizer at 20 psec. The extracted histogram (and further experimentation) lead us to believe that a lower random noise floor is present in our system, and most of our measurement uncertainty is due to



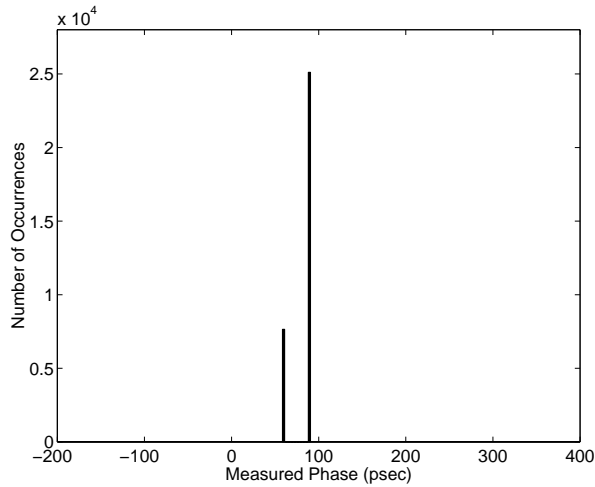
**Figure 10** Photograph of proposed module mounted on an evaluation board. Dimensions are 3.8" x 4".



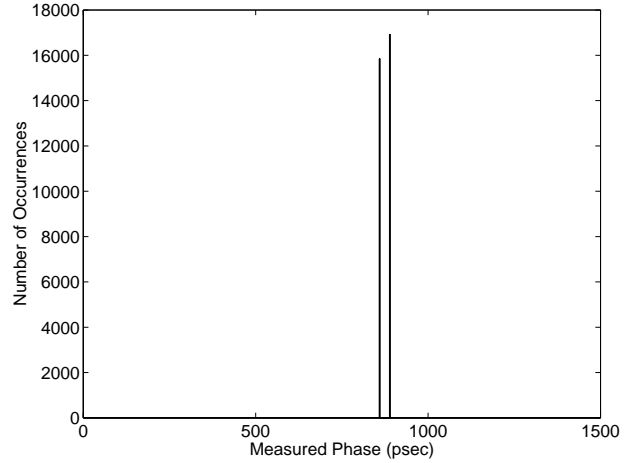
**Figure 11** Experimental setup for (a) TMU evaluation, (b) TGU evaluation.

quantization. We recognize that the coarse resolution limits the applicability of the device to the test of very fast data rates. However, it should be noted that the presence of the FFT processor allows us to discriminate smaller amplitude jitter phenomena by trading jitter signal bandwidth (i.e. using frequency domain filtering) [9]. Concerning measurement range, Figure 13 demonstrates very good linearity over the whole measurement range. This figure was obtained by measuring the two incoming edges while progressively varying their phase relationship. Each point on the graph is the average measured phase over 64K samples. The reason the code transition edges start to fade for larger phase differences is due to increased noise (hence uncertainty) for large phases. Specifically, the approach in [7] relies on two triggered oscillators to digitize time differences, so the longer these oscillators have to oscillate the larger the uncertainty in their edge locations (a phenomenon commonly referred to as jitter accumulation). Coming back to linearity, the extracted DNL error is about 0.5 LSB at the measured resolution.

The same set of experiments was repeated at an input rate of 5 Gbps and the measured results are shown in Figure 14 (histogram) and Figure 15 (linearity). As can be seen, the same remarks about resolution and noise floor apply for the extracted histogram. Also, the linearity, as expected, remains unaffected at the larger frequency, which suggests good signal integrity preservation in our design.

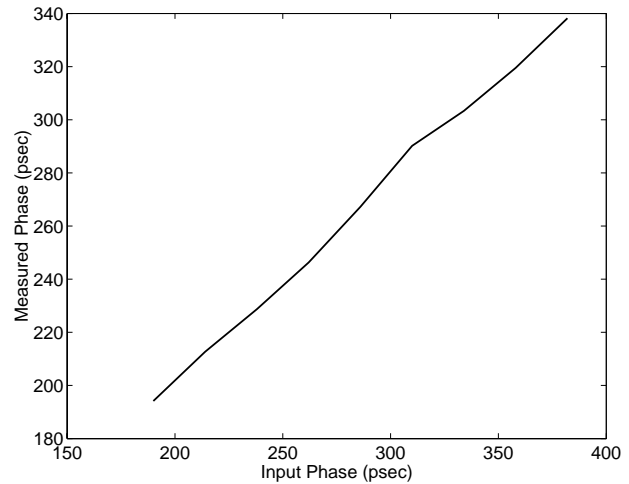


**Figure 12** Measured histogram of a jitter-free signal at 300 MHz.

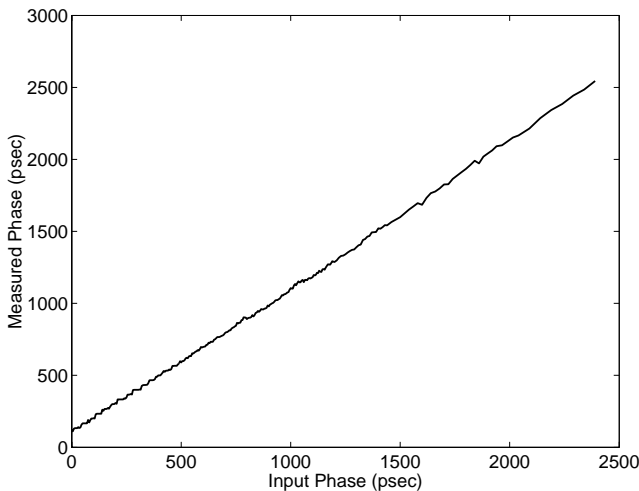


**Figure 14** Measured histogram of a jitter-free signal at 5 Gbps.

The next experiment was to exercise the TGU channels. First, the stability of the output signals was evaluated. This was performed by programming a single output phase and monitoring the total jitter on the output LVDS and CML signals. Ideally, zero jitter should exist. The actual amount of jitter is 2.7 psec on the LVDS channels and less than 4 psec in the CML channels, as illustrated in Figure 16. These are extremely good figures, and more importantly, the jitter remains relatively constant across all programmed output phases (the reported numbers are worst cases across channels and phase selections). As for the linearity of the phase increments, Figure 17 shows an experimental result in which a simple sweep of programmable output phases was performed and the output phase was measured on the oscilloscope. As can be seen, a delay step of about 60 psec is achieved, and extremely good linearity is maintained over all output phases (3 psec). When the vernier phase



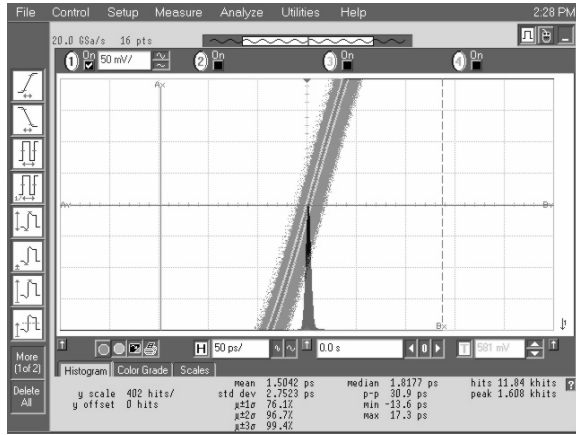
**Figure 15** Measured TMU transfer characteristic at 5 Gbps.



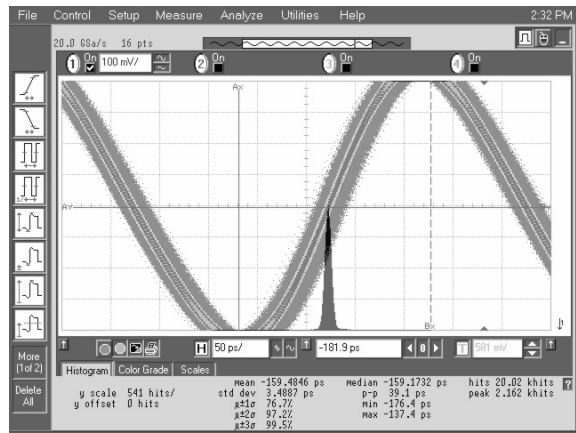
**Figure 13** Measured TMU transfer characteristic at 300 MHz.

generation is exercised, the resulting behavior is shown in Figure 18. In this figure, the *relative* delay between one of the TX channels and the output Clk channel is displayed. As can be seen, the relative difference between the two channels can be programmed to within 3 psec with reasonable linearity. This feature is useful for crosstalk measurements as well as jitter injection or setup and hold time testing.

As far as the actual waveforms are concerned, Figure 19 shows another oscilloscope capture illustrating complete LVDS waveforms. As can be seen, good waveform fidelity is maintained even as the signals traverse the package parasitics out of the measurement engine IC's, the module board connector, and the evaluation board traces. The figure also illustrates the actual phase generation capability. This figure was created by generating a repetitive "ramp" vector as a "program" to the TGU



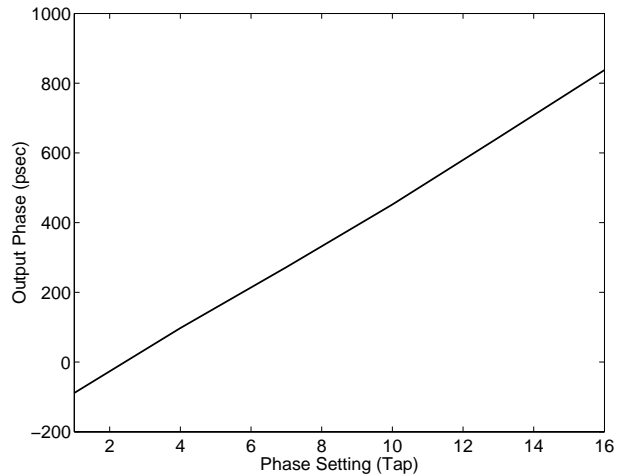
(a)



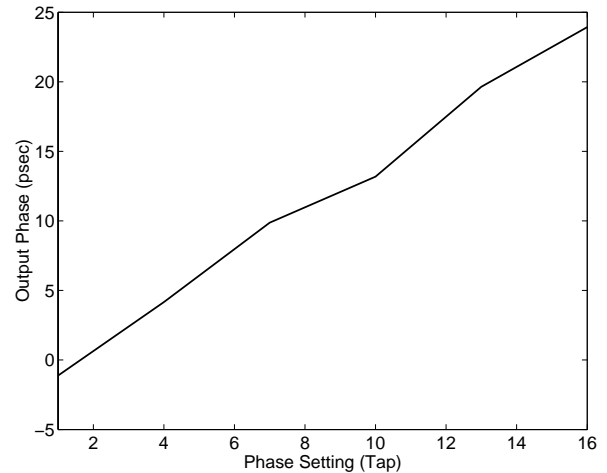
(b)

**Figure 16** Oscilloscope capture illustrating jitter performance of the (a) LVDS TGU output channels (2.75 psec) and the (b) CML outputs (3.44 psec).

channel. The vector was passed to the module's compute engine according to the format of Figure 8, and the compute engine cycled through the vector in order to periodically alter the output phase of the TGU and loop around all possible phase outputs. It is important to note that once the program was loaded into the module's compute engine, stepping through the vector was done automatically without user intervention. Figure 20 illustrates the output waveforms for the CML channels running at the maximum speed. Significant attenuation is observed in the signal both due to our connector performance and due to oscilloscope and probe bandwidth limitations (7 GHz). Fortunately, for 1,0,1,0 patterns, such effects are negligible thus allowing us to extract the raw performance of our module. Table 2 summarizes some other performance metrics for our module.



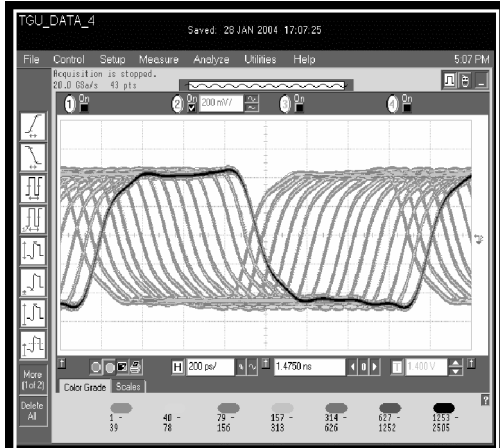
**Figure 17** Measured TGU transfer characteristic across all phase settings.



**Figure 18** Measured TGU transfer characteristic in vernier mode. The plot shows the relative phase difference between an output channel and an output clock.

## V. Conclusions and Comments

In this paper, a compact DIB-based jitter test solution suited for 5 Gbps SERDES or other timing-critical applications has been presented. The solution relies on custom integrated circuits for performing GHz-rate jitter stimulus and measurement functions and on local signal processing hardware for parameter extraction. The combination of these two components, in addition to a simple communications link to the host ATE, has been shown in this paper to provide a highly versatile solution that is well suited for production environments. Indeed, looking at the prior art, incorporating GHz-rate test capability in production on a digital tester or a legacy system has usually been achieved using bench equipment. The main drawback of such approach, apart from the size,

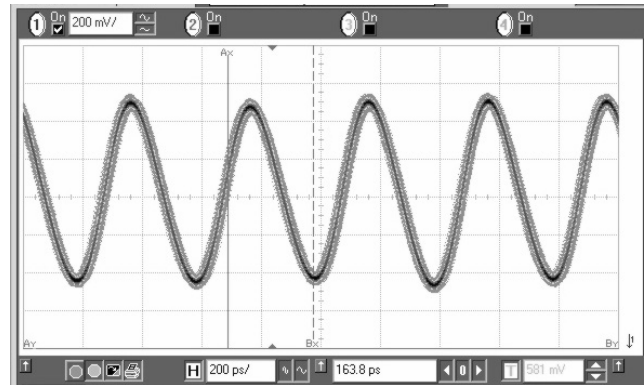


**Figure 19** Oscilloscope plot showing LVDS output characteristics and automatic phase generation capability.

is the throughput bottleneck associated with taking the measurement and with passing test data (in large volumes) to the host ATE. Our proposed solution is free of such limitations. At the other end of the spectrum are embedded BIST solutions. We continue to believe that on-chip embedded solutions are key to increasing test coverage and reducing test cost. The proposed solution is seen as a compromise between an embedded approach and the use of bench equipment. As such, it borrows from the benefits and disadvantages of each approach. It is more focused than bench equipment, yet it is not as flexible or general-purpose; it covers the test of I/O interfaces, yet it does not

**Table 2** Summary of Results

Technology	Custom ASIC (0.18 $\mu\text{m}$ CMOS), Standard Logic, Memory, & Interface Components
Volume	3.8" x 4" x 0.7"
Load-Board Landing Pad Area	2" x 0.4" per connector
Sample Acquisition Rate (All Channels)	10,000,000 Sample/Sec
Sample Acquisition Rate (single Channel)	2,000,000 Sample/Sec
Parameter Extraction Throughput (simultaneous on all channels)	~0.5 msec @ 4096 samples
Current Draw	3 A
TGU Output RMS Jitter (all channels, all phase settings)	2.7 psec LVDS 4 psec CML
TGU Range	1.2 nsec
TGU Linearity	3 psec
TMU Range	2.4 nsec
TMU Resolution	20 psec
Maximum Data Rate	5 Gbps



**Figure 20** Oscilloscope plot showing CML waveform characteristics.

provide maximum internal observability or controllability. Other limitations of our approach include the utilization of some ATE resources and the complication of continuity testing. These are the subjects of on-going research.

## VI. References

- [1] M. Hafeed, G. W. Roberts, "Techniques for High-Frequency Integrated Test and Measurement," *IEEE Transactions on Instrumentation and Measurement*, Vol. 52, No. 6, pp. 1780-1786, December 2003.
- [2] J. S. Davis, D. C. Keezer, "Multi-Purpose Digital Test Core Utilizing Programmable Logic," *Proc. IEEE International Test Conference*, pp. 438-445, 2002.
- [3] D. C. Keezer, Q. Zhou, C. Bair, J. Kuan, B. Poole, "Terabit-per-second Automated Digital Testing," *Proc. IEEE International Test Conference*, pp. 1143-115, 2001.
- [4] D. C. Keezer, R. J. Wenzel, "Low Cost ATE Pin Electronics for Multigigabit-per-Second At-Speed Test," *Proc. IEEE International Test Conference*, pp. 94-100, 1997.
- [5] J. Ferrario, R. Wolf, S. Moss, M. Slamani, "A Low-Cost Test Solution for Wireless Phone RFICs," *IEEE Communications Magazine*, Vol. 41, No. 9, pp. 82-88, September 2003.
- [6] T. J. Yamaguchi, M. Soma, D. Halter, R. Raina, J. Nissen, M. Ishida, "A Method for measuring Cycle-to-Cycle Period Jitter of High-Frequency Clock Signals," *Proc. IEEE VLSI Test Symposium*, pp. 102-110, 2001.
- [7] A. H. Chan, G. W. Roberts, "A Jitter Characterization System Using a Component-Invariant Vernier Delay Line," *IEEE Transactions on Very Large Scale Integration*, Vol. 12, No. 1, pp. 79-95, January 2004.
- [8] Mohamed Hafeed, "Analog and Mixed-Signal Test Methods Using On-Chip Embedded Test Cores," *Ph.D. Dissertation*, McGill University, 2002.
- [9] A. Oppenheim, R. Schaffer, J. Buck, *Discrete-Time Signal Processing*, 2<sup>nd</sup> Edition, Prentice Hall, 1999.