Adaptive parametric BIST of high-speed parallel I/Os via standard boundary scan

Stephen Sunter, Aubin Roy
Mentor Graphics, Ottawa, Canada
{stephen_sunter, aubin_roy}@mentor.com

Abstract—**An I/O BIST with calibrated 50 ps delay measurement resolution, presented at ITC'10, requires no changes to 1149.1 boundary scan cells nor the I/O cells they access, and avoids using delay cells or delay matching. In this paper, we describe how we improved the silicon-proven resolution by 10X (to 5 ps) without limiting the delay range. The finer resolution enables measurement of crosstalk between I/Os, which is a known source of jitter – we provide results measured on an FPGA. We also describe three new tests (for duty cycle, slew rate, and skew) that unobtrusively test I/O parameters while high-speed data from core logic is being transmitted on DDR pins (or USB2). Lastly, we describe how adaptive test limits are applied by setting limits on-chip relative to the average value measured for any selected group of pins on each device.**

Keywords- I/O BIST; delay measurement; DDR; high-speed I/O

I. INTRODUCTION

Integrated circuits (ICs) with flash memory, or large amounts of DRAM, SRAM, or mixed-signal circuitry, have fundamental limits on how much their test time can be reduced. Multi-site testing of these ICs is the only way to further reduce the effective test time per IC.

To facilitate multi-site testing without increasing the number of automatic test equipment (ATE) channels, many pins of devices-under-test (DUTs) are not connected to ATE channels. The on-chip circuitry of these pins is tested only via boundary scan for shorts and opens by making all uncontacted pins bidirectional, as shown in Figure 1. Some companies only contact the pins at package/final test; even then, to simplify test development, they only test DC parameters.

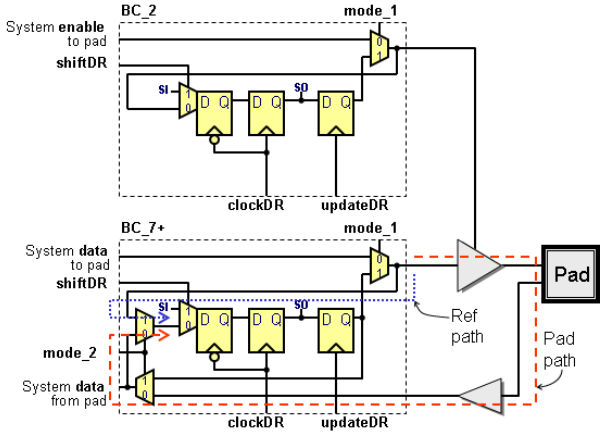

Figure 1. Bidirectional I/O pad and boundary scan cells

In an ITC'10 paper [1], an I/O BIST was described that facilitates AC testing of any input/output (I/O) pin via standard 1149.1 boundary scan [2]. No changes are needed in the boundary scan cells, as long as they are similar to the BC_2 or BC_7 cells (as in Fig. 1) specified in the standard. And no changes are needed in the I/O cells: preferably they are bidirectional, but output-only pins may be connected off-chip to input-only pins to permit BIST testing of their delays. The delay measurements are made without using delay lines, by using a PLL to generate a slightly asynchronous sampling frequency (inherently calibrated) and applying it to only the capture latches of the boundary scan cells.

Most companies regard testing delays of low-speed I/Os (<50 MHz) as unnecessary, but higher speed I/Os like those used for DDR and USB2 have important timing requirements and more circuitry susceptible to random defects and process variations. Adding delay measurement circuitry in these I/O cells usually requires altering the layout, and hence the tests cannot be applied by users of hard macro-cells.

However, the boundary scan cells in DDR macros may provide enough access for an I/O BIST that tests AC parameters via boundary scan. The sampling instant for DDR inputs is usually controlled by a delay-locked loop (DLL) that generates a clock edge mid-way between received clock edges. Jitter on DDR pins is sometimes tested by applying pseudo-random data while intentionally selecting increasingly un-centered DLL clock edges and detecting any resulting bit errors. But the timing of the DLL outputs is untested, uncalibrated, and relatively coarse.

For many I/O AC parameters, matching between I/Os is more important than whether each pin's performance meets absolute specifications. In DDR interfaces, each of the I/Os within a group of DQ I/Os is designed to have identical performance to the others. Nevertheless, process variations create inter-pin differences in duty cycle, clock-to-Q delay, and slew rate, which cause timing skew. All published BIST techniques that we are aware of, including [3][4][5][6], do not *measure* pin-to-pin performance matching, even BISTs that compare measured performance to test limits on-chip. Reference [1] provides a summary of these BISTs.

A variety of time-to-digital converters (TDCs) have been published, most of them based on Vernier delay line or oscillator techniques, or combining them ([7] is a recent example), aimed at measuring one-shot events, such as phase delay of each clock cycle in a PLL. The Vernier oscillator

978-1-4577-0152-8/11/$26.00 ©2011 IEEE

principle has also been used for digital-to-time conversion (DTC) for generating pulses with 2 ps resolution and a range up to one hour [8]. However, none of these techniques has been applied to measuring I/O delays via boundary scan; placing measurement circuits at each I/O location is impractical.

Section II of this paper summarizes the technique first described in [1]. Section III describes how we decreased the test time significantly, and improved measurement resolution by an order of magnitude while retaining the same (virtually unlimited) delay range. Section IV describes how the measured delays are compared to test limits, and how the test limits were extended beyond testing individual pin rise and fall times to measuring rise-fall mismatch, pin-to-pin mismatch, pin-to-reference mismatch, and pin-to-average mismatch. These latter tests facilitate adaptive testing, where pin delays are permitted to vary greatly between devices but much less between pins, or where outlier delays are within specification but may indicate a defect. Section V describes how the boundary-scan-based technique was adapted to measure timing parameters of data signals operating at rates much higher than the boundary scan clock rate. Section VI provides hardware results, and section VII discusses these results.

II. Measuring I/O Delays via Boundary Scan

The technique described in [1] is based on the "I/O wrap delay" measurement technique described in [3] which used a precision-delayed strobe pulse generated by ATE or an on-chip delay line; the strobe pulse was delivered to a capture latch with increasing delays until the captured result changed.

In the newer technique [1], the clock for Capture latches is derived from an asynchronous but coherent frequency, relative to the frequency used for Update latches. The asynchronous clock is generated by a PLL normally used by the core logic of the IC. The ratio between the 'Async' clock and the 'Ref' clock used for all other boundary scan logic is made equal to (N-1)/N, where N is an integer, typically >10. The TAP clock, TCK, was not used because its frequency is typically not constant and may have considerable jitter.

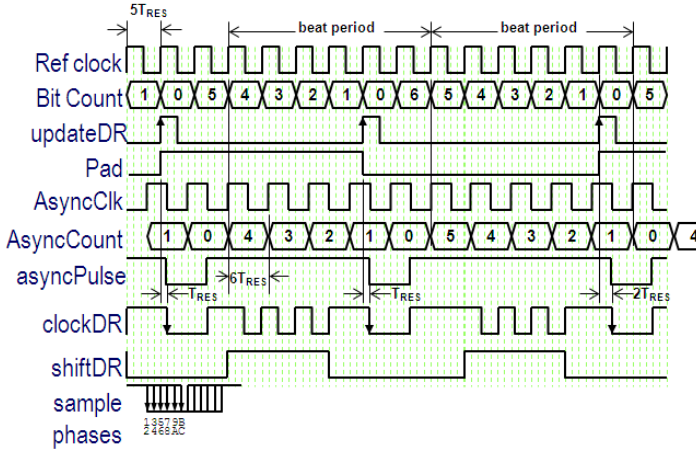

Figure 2. Waveforms for N=6, and L_{BSR}=3

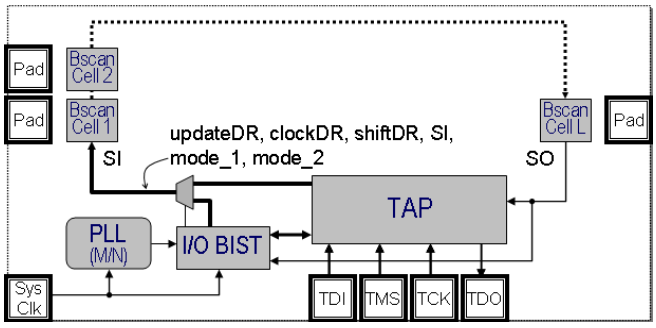

Figure 3. BIST connection to 1149.1 TAP, BSR, and system PLL

The technique's measurement resolution, T_{RES}, is equal to the Ref clock's period divided by N. Fig. 2 shows waveforms for an example 3-bit boundary scan register (L_{BSR} is its length, in bits), when N=6.

The BIST is initialized and loaded with test setup data via the 1149.1 TAP at TCK clock rates, and a reference path in each I/O cell (see Fig. 1) is selected via the mode_2 control. The BIST's Ref clock input is then connected synchronously to a reference clock, such as the IC's primary reference clock or a divided-down version of it. The BIST's Async clock input is similarly switched to a PLL-generated clock. In either or both cases, the clock can be generated by an off-chip PLL to obtain finer resolution.

After initialization, which occurs when the Ref and Async clock edges align, the boundary scan register (BSR) circuitry is clocked continuously, selecting the first Capture clock edge so that it occurs one half clock-period in advance of the Update edge – this ensures that up to one half clock-period of skew can be tolerated. The captured data is then shifted to the BIST circuitry. The BIST module has multiple "measurement slices" containing counters, each slice assigned to a different I/O address (bit position in the BSR). As boundary scan data is shifted out, a master counter is incremented and when its count equals that of one of the I/O address registers, the measurement counter associated with that address register is incremented if an edge has been detected earlier. Simultaneously, data shifted into the BSR is set so that the associated I/O will be updated to generate an alternating signal. Boundary scan data is recirculated for all uninvolved I/Os so that they are stable and known.

Other key features of the technique [1] are:

- The BIST comprises a common module plus a designer-selected number of measurement slices; prior to a measurement, each slice is loaded with the address for each I/O tested simultaneously (i.e., the address of the I/O's data or enable bit location in the BSR).
- The BIST block is inserted between the TAP controller and BSR as shown in Fig. 3; during a test, the BIST controls BSR clocks, mode, and data.
- Time intervals or delay differences are measured, e.g., from one signal edge to another, from a common reference edge to an edge of one path signal versus another (e.g., Ref path *vs*. Pad path in Fig. 1), or from a reference signal edge to an edge of a path signal under two conditions (e.g., high *vs*. low drive).

- N parallel captures are performed, spanning one clock period for one path or condition, and again for a second path or condition; measurement slices count from when their pin's captured data rises/falls for one path/ condition to the time it rises/falls for a second path/ condition.

For most measurements, the time to test each group of I/Os is calculated as follows:

$$T_{TEST} \approx 4 \times \text{Roundup}(L_{BSR}/N) \times N^2 T_{REF} \quad (1)$$

where T_{REF} is the Ref clock period (N and L_{BSR} were defined earlier). For a 40 ns period, 100 ps resolution (N=400), and L_{BSR}<400, test time is 26 ms per group of I/Os; for L_{BSR} = 400~800, it is 52 ms.

The total test time depends on the total number of pins whose AC performance is to be tested. For the case above where T_{TEST} = 52 ms, if the number of measurement slices was 16 and the total number of pin delays to be measured was 10X that number, then total test time would be 10 × 52 = 520 ms.

The BIST gate count is approximately 700 per measurement slice, plus 2000. The number of slices can be optimized for test time and silicon area, and hence, test cost.

III. Improving Range, Resolution, and Test Time

One way to reduce test time is to increase the reference frequency, since that reduces both the range and the time to shift out the results, hence the N^2 term in (1). However, if the I/O path delay (measured from the TAP controller updateDR edge to the transition-capturing clockDR edge) is greater than the range, its delay cannot be measured – it will be "out of range". As stated earlier, the range is about a half clock-period, so increasing the frequency reduces the range.

To circumvent this range limitation, we abandoned trying to mimic normal TAP controller operation, in which shifting immediately follows capture, and instead doubled the time between the capture and shift cycles, and doubled the number of captures per path/condition to span two clock periods. This actually doubles the measurement time, since twice as many samples are analyzed. But it permits the clock frequency to be doubled (assuming the BSR can operate twice as fast, which, as processes shrink, is an increasingly safe assumption), while increasing the usable range.

For example, at 25 MHz (40 ns period), the previous usable range was only 20 ns because captures begin one half clock period in advance of the update edge to permit clock skew. Doubling the frequency to 50 MHz (assuming layout constrains the clock skew to one half clock-period) and doubling the span to two clock periods, means the new range following the update edge is 1.5 clock periods or 30 ns, as shown in Fig. 4, which is a 50% increase in range despite the shorter test time.

Furthermore, since N is reduced by 2X for the same resolution, it reduces test time even more when L_{BSR} is relatively short, and reduces the PLL requirements. For example, the new test time for the 100 ps resolution case earlier is 6.4 ms for L_{BSR} <200, 13 ms for L_{BSR} <400, and 26 ms for L_{BSR} <800 pins, which is ≥2X improvement for the 52 ms example provided earlier.

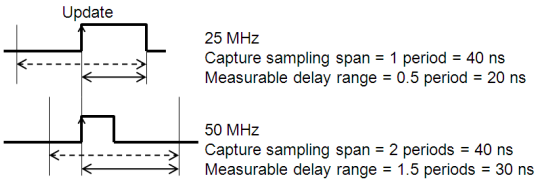

Figure 4. Capture range around Update edge, for two clock frequencies

The lower value of N permitted has the additional benefit of reducing the divider values needed in any PLL that generates the Async clock, which also reduces clock jitter.

In our RTL-generation software, we permit the range to be multiplied by any integer (i.e., not just 2), and hence it is unlimited, though we expect that it will rarely need to be increased to more than two or three clock periods.

Some parameters are tested by measuring the delay change caused by altering a path's characteristics, such as an output drive or input threshold voltage, but we wanted to avoid prescribing how to change these characteristics. For this reason, we redesigned the delay measurement so that after spanning the delay range for the first time (and setting a bit that indicates "phase 1 of 2"), all counter contents are retained while any subsequent TAP operation is performed, even a TAP reset. This permits a test pattern to invoke any alteration, via the TAP or other pins, or any non-BIST registers, and to then perform the second half of the delay measurement ("phase 2 of 2").

IV. Comparing Measurements to Test Limits

The BIST's delay measurements described thus far include rise delay and fall delay for one path or condition versus another. To facilitate adaptive test (described later), we added two counters so that while each measurement slice's rise and fall counters are being incremented, all the increment signals are also ORed together to increment an "average rise" counter and an "average fall" counter.

BIST operation always includes two steps: a measurement followed by shifting out all measurement counter contents (for characterization), or a measurement followed by comparison to shifted-in limits and shifting out pass/fail bits (for production test). The rest of this section describes various comparisons to test limits – each comparison is optional and is followed by an additional sub-step of shifting in upper or lower test limits for all measurement slices, or the two average counters, and shifting out all the resulting pass/fail bits – the test time added by any of these sub-steps is less than a millisecond.

After measuring a group of pin delays simultaneously, if these delay counts are to be compared to test limits, the rise and fall delay counters of all the measurement slices are concatenated into a single circular shift register; the two average counters are concatenated into a separate circular shift register. Then, as lower (or upper) test limits are shifted in via the TDI pin of the TAP and the concatenated counters are shifted, each limit is compared to each count to produce a pass/fail bit stored within each measurement slice.

Optionally, as the counts are shifted serially past a single

subtract/compare unit, different adjacent pairs of counts are subtracted from each other and the result compared to test limits. For example, each I/O's fall count can be subtracted from its rise count to produce a rise-fall mismatch. Each rise (and fall) count can be subtracted from the neighboring slice's rise (and fall) count to produce a pin-to-pin rise (and fall) mismatch count – note that each slice can be programmed with any I/O's address so delays of any two pins can be compared. Also, each rise (and fall) count can be subtracted from the rise (and fall) count of any selected reference I/O (assigned to slice #1). Lastly, each rise (and fall) count can be subtracted from the average rise (and fall) count.

A. Adaptive BIST

As stated in the ITRS 2009 roadmap [9], adaptive test refers to methods that change test conditions, flow, and limits "potentially at the die/unit or sub-die level". Test limits may be adjusted based on average measurements for the present wafer, or based on neighboring dies on the wafer, or even based on nearby structures within the same die.

For ICs in automotive applications, part average testing (PAT) [10] is often required to detect performance parameter outliers that are within device specification limits but indicate potential reliability defects. Test limits are set equal to the mean ± 6 sigma for multiple wafers and may be adjusted dynamically ("DPAT") as each additional wafer is tested.

To implement adaptive testing or DPAT, the test limits are conventionally computed on-line in the ATE during the test, or offline after wafer sort and prior to packaging ("post-processing"). Both approaches require test programs and ATE that perform computations based on measurements or captured data, and hence prevent use of simple WGL or STIL test patterns that run on any low-cost ATE. The latter approach also requires tracking each die after wafer sort.

On-chip comparison of individual I/O delay measurements to the average delay for a group of I/Os, as we describe here, is analogous to industry standard DPAT but allows WGL/STIL patterns to be used since no computations are needed in the ATE. The applied patterns contain binary-coded test limits for the difference between the individual measurements and each I/O group average. Many groups can be defined – for example, a group might comprise multiple I/Os using the same I/O cell.

The advantage of adaptive testing at the sub-die level is that every measurement is treated equally, including the first I/O group of the first die on the first wafer in the first lot. Defects that affect a single path are more detectable with a sub-die approach because each path can be compared to paths having more similar process, voltage, and temperature than other dies. Note, however, that our approach compares each measurement to a dynamic mean assuming a constant sigma, whereas DPAT may use a dynamic sigma value.

Our approach also simultaneously tests that the average of each group of I/Os is within limits to ensure that defects affecting a whole group of I/Os are detected.

B. Detecting small delay defects

Instead of adaptive testing, testing pin-to-pin variations can simply be used to detect delay faults that are within normal die-to-die process variations but excessive considering that the I/O cells and loading were designed identically.

Differences in output loading at the board level, or in multi-die modules (SiP, 3D, MCM, etc.), can cause detectable delay differences. This is clearly important for high-speed parallel data busses, such as DDR interfaces which have delay skew specifications (discussed in next section), but it is also important for diagnosing inter-die interconnect opens. As these systems become denser, diagnosing the location of opens is increasingly difficult, yet essential for improving yield. If an open is closer to one die than another, then the difference in interconnect capacitance may be detectable, but only relative to other I/Os of the same type and loading.

Another small delay of interest occurs when a pin's transition affects that of neighboring pins – crosstalk. Our BIST measures this by changing the BSR contents between the first and second phases of the I/O delay measurement, so that neighboring pins both toggle in the same direction during phase 1 of the measurement, and toggle opposite to each other in phase 2 of the measurement. Test limits are applied to the change in each pin's measured delay, for both the rising edge and the falling edge. Test limits can also be applied to the mismatch between I/Os for this delay change.

V. MEASURING HIGH-SPEED SIGNAL TIMING

The delay measurements described thus far use the Update latch in boundary scan cells to provide the stimulus. Each time the BSR is reloaded, the data for the Update latch is alternated (and the expected value is alternated). As described in [1], this allows many timing properties of the I/O cells to be measured, and with time resolution limited by only the frequency resolution of the PLL that provides the Async clock.

Note that the Ref clock could be provided by the PLL, and the Async clock provided by a crystal – any combination that delivers an (N-1)/N ratio between the clock frequencies is acceptable.

For I/Os that deliver signals at frequencies higher than the boundary scan rate, for example USB2 data at 480 Mb/s or DDR data at 1.6 Gb/s, it will usually be more accurate to measure the timing parameters while data is delivered to the I/O from the core function logic instead of from the boundary scan logic.

We added this measurement capability by requiring that the core logic generate alternating data (1010…) at a rate synchronous to the BIST's Ref clock. This may be accomplished by loading a memory with alternating data and reading consecutive addresses, or by inserting a toggle flip-flop that supplies the data instead of the memory, in test mode. All edges in the data signals from the logic core must have an expected constant phase relative to the I/O BIST's Ref clock.

For example, if the BSR has a 50 MHz (20 ns period) Ref clock and a 49.9375 MHz (20.025 ns) Async clock to obtain 25 ps resolution, then the DDR logic can be clocked at any integer multiple of 50 MHz and the same resolution will be obtained. For example, the DDR interface may be clocked at 800 MHz to deliver data at 1.6 Gb/s. In this case, while the

alternating data is being transmitted via the DQ pins, the boundary scan logic can sample it synchronously to the 49.9375 MHz clock and shift out each sample to the BIST at 50 MHz. No data needs to be shifted into the BSR since Update latch contents are not used in this mode.

The BSR mode_1 signal (see Fig. 1) selects whether the boundary scan cell's Update latch output or the high-speed signal from the core is sampled, and the mode_2 signal selects whether it is sampled before or after the I/O driver (via pad).

The measurement range can be reduced to span only two periods of the high-speed clock, as shown in Fig. 5, which ensures that it spans at least one rising edge followed by a falling edge. This reduces test time proportionately.

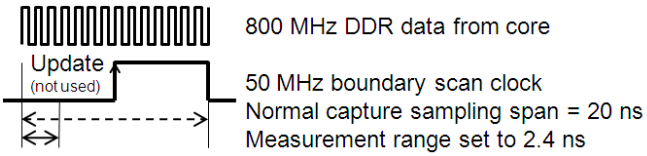

Figure 5. Capture range for high frequencies from logic core

A. Duty cycle distortion (DCD)

Total DCD for an I/O signal is equal to mismatch in output pad rise and fall times plus DCD in the data signal from the core logic. The data signal DCD is caused by mismatch in rise/fall delays of any logic between the last function register and the I/O cell.

To measure DCD in the core data signal, the BIST samples the signal with BSR Capture latches (mode_1=0 and mode_2=1) and counts the interval from each first detected rising edge to the next detected falling edge, in units of the measurement resolution (e.g., there should be 25 consecutive '1' samples for 1.6 Gb/s data, when sampled with 25 ps resolution). This count is compared directly to test limits.

The detected rising edge will be affected by jitter in the clocks and power rail noise, but the BIST computes median edge positions [1] so that it can tolerate peak-to-peak jitter up to six times the measurement resolution (with the default BIST logic; up to 30X can be tolerated with more logic gates).

B. Slew rate

Output slew rate can be measured by measuring the Update latch to Capture latch delay through the I/O pad at two different receiver threshold voltages. In [1], the change in threshold had to occur synchronously, whereas the new approach does not have this constraint between phases of a delay measurement. The input threshold for DDR inputs is controlled by a termination voltage (V_T) and it can be offset by, say, 50 mV between the first and second half of the delay measurement; the voltage change divided by the difference between the two delays is equal to the slew rate. During a production test, the division is not performed – the test limits are applied to only the measured delay difference.

Output slew rate can also be measured while core logic is the data source. While transmitting alternating data originating from the core logic, as done for DCD measurement, the BSR samples the I/O pad signal edges.

C. Clock-to-output delay

Measuring the delay from a clock input pin to a data output pin requires sampling with zero skew, which requires precision calibration and adjustment. If it is practical to sample signals of a small number of I/Os, using Capture latches clocked by a single wire output of a clock buffer, and hence with near-zero skew, then we can measure the delay from an edge of one of the signals to an edge of another of the signals. For example, DQS-to-DQ or clock-to-DQ delay may be measured, since DDR macros are constructed in groups of eight DQ pins per DQS pin.

To measure delay between edges of different signals occurring nearly simultaneously on different I/Os, delay is measured from an arbitrary phase reference – in our case it is the initialization instant, which is when selected edge types of the Ref and Async clock align. The delay measured for one I/O is then subtracted from the delay measured for a reference I/O, by assigning the reference I/O to measurement slice #1, as mentioned in section IV. As long as the reference I/O is sampled similarly to the other I/Os (i.e., has a similar boundary scan cell), regardless of whether it is an input or output clock, the difference can be measured.

D. Skew

Timing skew is the maximum difference in clock-to-output delay measured between any two output pins within a defined group of pins.

In [4], timing skew is measured by continually incrementing the delay of the sampling strobe pulse and counting the number of increments between the strobe time when the first output fails and the strobe time when the last output fails.

In our approach, each pin's clock-to-output delay can be compared to the average clock-to-output delay for the group. Through characterization, the pin with the longest (or shortest) clock-to-output delay could be identified, and then the clock-to-output delay of each pin in a group of simultaneously measured pins could be compared to that reference pin delay, and the difference compared to test limits. However, this would not actually measure skew.

To measure skew, we slightly modified the two counters that measure average count (for rise, and fall) so that, selectively, they start counting at initialization and stop counting when the last delay count increment pulse (for rise, and fall) is detected in any measurement slice – this produces a maximum count (for rise, and fall) instead of an average count. Then the difference between each I/O's delay count and the maximum count is compared to test limits as described earlier for the average count.

VI. Hardware Results

We used an Altera Stratix II FPGA (EP2SGX90EF1152) to implement the I/O BIST, and synthesized from RTL with no layout guidance other than typical clock timing skew constraints. The boundary scan cells were synthesized in the logic fabric of the FPGA (the FPGA's hard-wired boundary scan could not be used because its clocks, data, and control

signals are not accessible). We used four I/O pins that were adjacent to one another on the die and in the package, but not connected to the board, as shown in Table 1. We chose these pins to emulate contactless testing yet maximize crosstalk. We configured the FPGA with the four outputs programmed for 8 mA drive and then for 4 mA drive.

TABLE I. LOCATIONS OF MEASURED PINS

Label	FPGA die pin	1152-pin BGA package pin
Pin 1	X=0, Y=40	U28
Pin 2	X=0, Y=40	U29
Pin 3	X=0, Y=41	U30
Pin 4	X=0, Y=41	U31

The TAP interface was implemented with general-purpose FPGA I/O pins, and connected to an off-the-shelf "Signalyzer" USB-to-JTAG module [12] and cable connected to a desktop PC using a Linux operating system. All measurements were performed via a graphical user interface (GUI) on the PC. The TCK clock rate was limited by the speed of the PC software, and was approximately 2 MHz. Two differential reference clocks were generated by two LMK03000 PLLs [13]: one provided 200 MHz to the FPGA and 25 MHz to the other PLL, which in turn provided 24.9875 MHz to the FPGA. These PLLs allowed us to deliver clocks with less than 0.5 ps rms jitter, which is less than a typical on-chip PLL but they allowed us to explore the limits of the BIST technique's timing resolution and could be added to an ATE interface board. Within the FPGA, the 200 MHz was supplied as a high-speed core "data" signal to the boundary scan cells, and it was divided by 8 to provide a 25 MHz Ref clock to the BIST. Hence during measurements, the Ref clock period was 40.00 ns, the Async clock period was 40.02 ns, and the sampling resolution was the difference, 20 ps.

Each measurement was repeated 10 times, and delays for all four pins were measured simultaneously by four measurement slices. While pins 1 and 3 toggled, each delay was measured while adjacent pins 2 and 4 were either steady-state, toggling in the same direction as pins 1 and 2, or toggling in the opposite direction. Tables II~V show the mean of each set of ten measurements, and the standard deviations. Our objective was to verify only the sub-steps and phases of each measurement described in the preceding sections, since verifying every possible combination would provide minimal additional validation and would be too time-consuming.

A. I/O wrap delay

Mode_1 of the boundary scan cells (see Fig. 1) was set to logic 1 to select the update latch output signal, while mode_2 was logic 1 for the first half of the measurement to select the direct path from the update latch, and logic 0 for the second half of the measurement to select the path via the pad.

The output transitions are driven by the updateDR signal which was routed via a global clock in the FPGA logic fabric. The Update latches were also in the fabric but layout optimization naturally selects logic blocks as close as possible to the I/O pads.

Table II shows measured results for N=2000, which provides 20 ps resolution. Each delay in column 2 shows the difference between the direct path from the update latch, and the path via a pad, i.e. the I/O wrap delay. Columns 4 and 6 show these delays when there are signal transitions on adjacent pins (pins 2 and 4), relative to the delays in column 2.

These results show that I/O wrap propagation delays through 8 mA output drivers, pad, and input receiver, *decreased* by an average of 719 ps, about 10%, when the drive was decreased to 4 mA (recall that the I/Os are not connected to board wiring). The FPGA's datasheet [14] indicates about 300 ps *increase* for these two drives, with 0 pF load. If an external capacitance of, say, 50 pF were connected to the pins for package test, then delay would undoubtedly increase when drive decreases (by about 4 ns).

The 8 mA driver delay also decreased by an average of 24 ps when adjacent pins had the same transitions, and by 12 ps when adjacent pins had opposite transitions, whereas the 4 mA driver delay was unaffected (1~6 ps difference). The delay dependence on adjacent pin transitions will appear as crosstalk-induced jitter in function mode. In experiments with other 8 mA I/O pins connected to approximately 10 cm of adjacent board wiring, we measured almost 100 ps delay decrease for same transitions on adjacent pins, and 100 ps delay increase for opposite transitions.

Table II also shows mismatch in rise versus fall times: an average fall minus rise of 472 and 176 ps for pins 1 and 3, respectively, in the 8 mA case, and 238 and 258 ps in the 4 mA case.

TABLE II. I/O DELAY MEASUREMENT SUMMARY (IN PICOSECONDS)

Edge	Reference: Silent adjacent	Std. dev.	Transition same as adjacent	Std. dev.	Transition opposite to adjacent	Std. dev.
Pin 1 fall	**7538**	33	-24	25	-22	26
Pin 1 rise	**7066**	19	-12	19	-6	25
Pin 3 fall	**6854**	28	-28	28	-16	27
Pin 3 rise	**6678**	20	-32	19	-4	23
average	**7034**	**25**	**-24**	**23**	**-12**	**25**
Nominal output drive = 8 mA			w.r.t. silent		w.r.t. silent	

Edge	Reference: Silent adjacent	Std. dev.	Transition same as adjacent	Std. dev.	Transition opposite to adjacent	Std. dev.
Pin 1 fall	**6580**	27	12	29	+8	23
Pin 1 rise	**6342**	18	2	21	-8	33
Pin 3 fall	**6298**	33	0	22	+14	32
Pin 3 rise	**6040**	21	-10	22	+8	21
average	**6315**	**25**	**+1**	**23**	**+6**	**27**
Nominal output drive = 4 mA			w.r.t. silent		w.r.t. silent	

The results also show that standard deviation is slightly greater than the sampling resolution and much greater than the near-picosecond clock jitter, perhaps caused by other significant sources of random timing variation, such as input threshold noise and clock path jitter. To explore this further, we also performed the measurements with N=4445 and 25 MHz to obtain 9 ps resolution, and then with N=4000 and 50 MHz clocking to achieve 5 ps resolution. For the 9 ps case,

delays were similar to those in Table II, but standard deviations improved to 15 ps and 13 ps for the 8 mA and 4 mA drives, respectively, with transitions same as adjacent pins. This suggests that the variance is linearly proportional to the sampling resolution in picoseconds, i.e., $\sigma^2 \approx 25 \times T_{RES}$.

B. Slew rate

Measuring slew rate requires changing the input threshold of pad input receivers, or changing the output slew rate and measuring the change at a constant input threshold.

We were not able to adjust the threshold voltage of the FPGA inputs, since all on-chip comparators across pin-pairs used a reference voltage pin that was connected to ground on the board. We were also not able to adjust the slew rate by adjusting the drive of the output drivers because this particular FPGA has no slew rate control, nor by adding a capacitor load, because the distance between package pins and places where a capacitor load could be connected was too long for the capacitor to affect the slew rate at the pad. (Since we could not affect the slew rate, we chose the pins in Table I because they are not connected to any board wires.)

To prove that we could split the measurement into two distinct phases, each of which would measure an I/O delay for a different input threshold, we compared the single-phase measurement of I/O wrap delay shown in Table II (that measured the change in I/O wrap delay when switching between two separate paths) to measurement comprising two separate phases: phase 1 measures the delay from an Async clock reference edge to the captured edge while mode_2 is set to logic 0 (selects update-to-capture via pad), and phase 2 measures the delay while mode_2 is 1 (selects update-to-capture bypassing pad). Table III shows the results.

TABLE III. 2 VS 1-PHASE DELAY MEASUREMENT SUMMARY (IN PS)

Edge	2-ph: Phase 1	Std. dev.	2-ph: Phase 2	Phase 1-2	Δ	Reference: 1-Phase measure	Std. dev.
Pin 1 fall	5674	21	-1858	7532	-6	7538	33
Pin 1 rise	5196	25	-1862	7058	-8	7066	19
Pin 3 fall	5022	29	-1824	6846	-8	6854	28
Pin 3 rise	4822	25	-1856	6678	0	6678	20
Average	**5179**	**25**	**-1850**	**7029**	**-5**	**7034**	**25**

Nominal output drive = 8 mA Silent adjacent

From Table III, we can see that subtracting the measurements of phase 2 from the measurements for phase 1 produces an average result that is identical (within 5 ps) to that obtained in single-phase measurements (shown in column 7, copied from column 1 of Table II). Every measurement has approximately the same standard deviation, so the standard deviation of the subtraction will be higher by a factor of root two (= 1.414). Each measurement involves two edges subject to random jitter, so the difference between measurements involves four edges and hence has twice the variance.

C. Skew

Because skew is the variation in output transition delays relative to other transitions in a group, each output's transition delay can be measured relative to any synchronous signal and then the delays can be compared to each other. We used the BIST's ability to measure in two separate phases, but did not run a second phase of the measurement. While mode_2 was 0, we measured the time from the start of a beat period between the Ref and Async clocks, to the time of a selected edge type at an output pin (detected by sampling with the Async clock).

The top left-hand number of Table IV shows the average delay measured for the falling edge of pin 1 – an arbitrary reference edge – and the other cells (except bottom two rows) show average delays relative to that delay. The table shows that skew stays constant (within 12 ps), regardless of whether adjacent pins have no transitions, same transitions, or opposite transitions, despite individual delays varying up to 32 ps.

Note that column 2 of Table IV is based on the same data as column 2 of Table III, since we can use the Ref clock edge as a reference edge for skew measurement.

TABLE IV. SKEW MEASUREMENT SUMMARY (IN PICOSECONDS)

Edge	Silent adjacent	Transition same as adjacent	Transition opposite to adjacent
Pin 1 fall	**5674**	-4	+8
Pin 1 rise	-478	-468	-446
Pin 2 fall	-	-824	-828
Pin 2 rise	-	-1016	-996
Pin 3 fall	-652	-666	-666
Pin 3 rise	-852	-858	+842
Pin 4 fall	-	-584	-562
Pin 4 rise	-	-776	-752
Skew (1,3)	**852**	**854**	**850**
Skew (1~4)	**-**	**1016**	**1004**
Nominal output drive = 8 mA		w.r.t. Pin 1 fall, silent	w.r.t. Pin 1 fall, silent

Note: column 2 is based on the same data as column 2 of Table III

D. Duty cycle

Mode_1 of the boundary scan cells was set to logic 0 to select the core signal (200 MHz clock supplied differentially to the FPGA from off-chip PLL), while mode_2 was logic 0 to select the pad signal. The positive and negative pulse widths were measured simultaneously (the test measures pulse width instead of duty cycle), starting from first detected target edge type to next opposite edge type. The results are shown in Table V (next page) for duty cycle distortion (DCD). The sum of the two pulse width measurements for each pin provides a checksum that should equal the period of the 200 MHz clock (and does, within 36 ps or 0.7%), since the measurements are independent. The adjacent pins' drivers were steady-state.

The measurements in Table II for pin 1, with silent adjacent pins, showed 472 and 176 ps mismatch for pins 1 and 3, whereas the same case in Table V shows 2788-2248=540 ps and 2694-2336=358 ps mismatch, respectively. Thus the 200 MHz clock on-chip appears to have 540-472=68 ps or 358-176=182 ps positive-negative pulse width mismatch, corresponding to 0.7% or 1.8% DCD, respectively. The 200 MHz 50% duty-cycle clock was provided differentially to the FPGA from the off-chip PLL, but clock buffers and logic gates within the FPGA will alter the duty cycle of any clock, so this amount of DCD seems reasonable and unavoidable.

TABLE V. DCD MEASUREMENT SUMMARY (IN PICOSECONDS)

Parameter	Measured value	Std. Dev.
Pin1 pos. width	2788	25
Pin1 neg. width	2248	30
Sum	5036	
DCD	5.4%	
Pin3 pos. width	2694	21
Pin3 neg. width	2336	28
Sum	5030	
DCD	3.6%	

Nominal output drive = 8 mA. Silent adjacent.

E. Clock-to-output delay

Measuring clock-to-output delay is similar to measuring skew, except that one of the pad signals is a synchronous clock input or output, instead of an output from an Update latch. Since our duty cycle measurement demonstrated the BIST's ability to measure timing of signals that did not originate from an Update latch, and the skew measurement demonstrated the ability to measure delays for multiple pad signals relative to a common reference time, we did not perform a separate verification of measuring clock-to-output delay.

VII. DISCUSSION

The I/O wrap measurement results for a path including only a multiplexer, pad driver, pad receiver, and another multiplexer, show that the delay for uncontacted pins was approximately 7 ns and decreased by 10% when output drive decreased. For loaded pins, the delay could be expected to increase by many nanoseconds. The delay varied less than 25 ps when there was crosstalk from adjacent pins but by 200 ps when connected to 10 cm of wiring.

The results also show that a delay test of an uncontacted pin would not reliably distinguish 4 mA from 8 mA drive – a load capacitance is needed for final test.

The skew measurement results show about 1 ns skew, which would be excessive for DDR pins but we used core standard logic instead of DDR-optimized output registers.

The duty cycle measurement results show up to 5.4% DCD for the pad signal, but most of it is due to the rise-fall mismatch in the I/O circuitry that was measured as I/O wrap delay, since after accounting for that mismatch, the on-chip clock DCD is less than 2%. We measured DCD for only 200 MHz due to practical limitations of conveying higher frequencies into the FPGA (we did not use the on-chip PLL so that we could use our standard RTL flow), but nothing in the technique restricts the frequencies that can be measured – we used the same principle to measure DCD at 10 Gbps in SerDes signals [15]. Our 20 ps resolution equals 1% of a 500 MHz clock period, but resolution can be adjusted as we have shown.

A. General observations

The technique described in this paper allows arbitrary and almost unlimited timing resolution because the resolution is set by the ratio between the Ref and Async frequencies supplied to the BIST. On the FPGA, we verified resolutions ranging from 6.4 ns to 5 ps. Setting a coarser resolution usually requires a lower frequency which increases the time to shift out test results. Setting a finer resolution requires more shifts out, which increases test time. In practice, we found that BIST initialization effects require $N \geq 10$, most PLLs can only achieve $N < 512$, and test time becomes excessive for $N > 4000$ and for resolution coarser than 10 µs.

Table VI shows test times for a range of resolutions spanning 6 orders of magnitude, for different clock periods and lengths of boundary scan register, as calculated using equation (1). These test times are per group of simultaneously measured I/Os, and the number of I/Os in a group must be less than or equal to the number of measurement slices in the BIST implementation. The length of the BSR (L_{BSR}) has no effect on test time if it is less than N, and in that case, halving the resolution increases N by 2X and test time by 4X.

The last row of Table VI shows the dramatic effect of increasing the Ref clock frequency: doubling it reduces test time by 8X, if N stays greater than L_{BSR}. This cube-law test-time reduction is due to the shift out time halving, the N samples per period halving, and the sampling span halving.

So the resolution is mostly limited by test time, which is mostly limited by the Ref clock frequency.

TABLE VI. TEST TIME FOR VARIOUS RESOLUTIONS AND CLOCK PERIODS

$T_{RESOLUTION}$ (ns)	T_{REF} (ns)	N	L_{BSR} (bits)	T_{TEST} (ms)
5000	50000	10	100	200
1000	10000	10	100	40
1000	10000	10	500	200
100	1000	10	500	20
10	100	10	500	2
1	50	50	500	5
0.1	20	200	500	10
0.04	20	500	1000	40
0.01	20	2000	1000	320
0.005	20	4000	1000	1280
0.005	10	2000	1000	160

Most techniques for measuring time use simple delay lines, Vernier delay lines, DLLs, or Vernier ring oscillators. These techniques usually require analog design to achieve acceptable PVT sensitivity. Also, delay lines do not generate inherently calibrated delays, so each delay setting of interest must be calibrated. Their resolution is usually limited to one gate delay and is sensitive to VDD. Similarly, Vernier delay lines and ring oscillators are uncalibrated, unless they are in a DLL or PLL, where feedback compensates for changes in the power rail voltage (but not noise, which causes jitter).

For a DLL, delay resolution is always equal to the clock period (to which the delay loop is locked) divided by the number of inverters in the loop. If two DLLs are used, to implement Vernier delay lines, then finer resolution can be achieved, but only for a range equal to the resolution times the number of inverter stages – typically fewer than 32.

For a PLL, frequency resolution is independent of the number of inverters in its voltage-controlled oscillator. Instead, it depends on the feedback binary divider, which is typically 6~10 bits, providing up to 1024 steps of resolution. Many ICs contain more than one of this type of PLL; a pair of

these PLLs can usually be programmed as Vernier oscillators to achieve 4X finer resolution, which is how the I/O BIST described in this paper can be clocked. For example, from a common reference clock with a 25.6 ns period (39 MHz), PLL1 can use a feedback divider of 25 and output divider of 216 to generate a Ref clock period of 55.296 ns, and PLL2 can use a feedback divider of 19 and output divider of 164 to generate an Async clock period of 55.242 ns, to provide an I/O BIST resolution of 54 ps (for which, N=1026). Thus 10-bit resolution is obtained from two PLLs that have 8-bit dividers.

PLLs that use an LC tank oscillator are typically 10X larger [16] than ring-oscillator-based PLLs but can achieve resolution up to 28 bits to provide more than a million steps of resolution per clock period. These LC-based PLLs may be on-chip [11] or used as a clock conditioner on an ATE loadboard or in a test head. We used board-mounted clock conditioner PLLs [13] for all the results in this paper.

We did not repeat any of the leakage tests performed in [1], in which the discharge delay of tri-stated pins is measured, but it should be noted that a leakage test is an important aspect of testing pins since it verifies that transistors can turn off completely whereas AC tests primarily verify how fast transistors can turn on. To achieve highest coverage without connecting ATE channels to I/O pins, the results in this paper combined with the leakage tests in [1] show that I/O speed and crosstalk can be tested contactlessly but capacitive loading is required to verify output transistor drive.

Measuring these delays with ATE would require per-pin timing measurement units with similar timing resolution, manually-written test patterns, and ATE computations. Using ATE to measure jitter on many pins would not distinguish between clock jitter and crosstalk, and require an even higher performance tester.

VIII. Conclusion

This paper presents improvements to a BIST [1] that measures timing parameters of circuit paths related to integrated circuit I/Os that will be or are connected to package pins or other components. The parameters can be measured without contacting the I/O pad or changing the I/O cells.

The improvements include 10X finer timing resolution, to 5 ps, which also improves measurement repeatability, and an arbitrarily large delay range spanning multiple clock periods. This is the smallest, calibrated resolution ever reported for I/O delay testing, and even finer resolution is possible (without changing the circuitry). This resolution allows simpler, faster measurement of timing parameters for parallel high-speed I/Os, such as those in DDR interfaces.

The improvements also include a wide variety of matching tests, such as mismatch in pin rise-fall, pin-to-pin, pin-to-reference-pin, pin-to-average-pin, and pin to maximum-pin delays. All the mismatch measurements can be compared to upper and lower test limits on-chip, which enables use of simple test patterns that run on any ATE.

The pin-to-average tests permit adaptive testing that can detect delay outliers that may indicate reliability faults, similar to the methods defined for part average testing.

With this resolution capability, we report the first calibrated, silicon-verified BIST that can access I/O circuitry via only standard boundary scan, and measure duty cycle, slew rate, skew, and crosstalk for signals at frequencies much higher than the scan clock rate.

The hardware results in this paper have shown that a purely digital, synthesizable BIST, together with one or two clock sources and a desktop PC, can be used to characterize and test the performance of high-speed parallel I/Os with arbitrary timing precision. The test patterns are suitable for high-volume, multi-site production testing on low-cost ATE.

Acknowledgment

The authors would like to thank the anonymous reviewer who asked us to explain why our original results for 4 and 8 mA output drives were almost identical. Upon reviewing our set-up, we realized that we had tested the same drive twice.

References

[1] S. Sunter, M. Tilmann, "BIST of I/O circuit parameters via standard boundary scan", *Proc. of ITC*, pp. 529–551, November 2010.

[2] IEEE Std. 1149.1-2001, "IEEE standard test access port and boundary-scan architecture", The IEEE, Inc., 345 East 47th St., New York, NY

[3] P. Gillis, F. Woytowich, K. McCauley, U. Baur, "Delay test of chip I/Os using LSSD boundary scan", *Proc. of ITC,* pp. 83-90, October 1998.

[4] M. Tripp, T.M. Mak, A. Meixner, "Elimination of traditional functional testing of interface timings at Intel", *Proc. of ITC*, October 2003.

[5] S. Sunter, C. McDonald, G. Danialy, "Contactless digital testing of IC pin leakage currents", *Proc. of ITC*, pp. 204-10, October 2001.

[6] N. Vijayaraghavan, B. Singh; S. Singh; V. Srivastava, "Novel architecture for on-chip AC characterization of I/Os" *Proc. of ITC*, October 2006.

[7] J. Yu, F. Dai, R. Jaeger, "A 12-bit Vernier ring time-to-digital converter in 0.13 µm CMOS technology", *Jour. of Solid-State Circ.*, vol. 45, no. 4, April 2010.

[8] P. Chen, P-Y Chen, J-S Lai, Y-J Chen, "FPGA Vernier digital-to-time converter with 1.58 ps resolution and 59.3 minutes operation range", *IEEE Trans. on Circ. And Sys.-I*, vol. 57, no. 6, June 2010.

[9] ITRS 2009, "International technology roadmap for semiconductors, 2009 Edition, Test and test equipment" http://www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/2009_Test.pdf

[10] AEC - Q001, "Guidelines for part average testing" July 2003 http://www.aecouncil.com/AECDocuments.html

[11] R. Kinger, S. Narasimhawsamy, S. Sunter, "Experiences with Parametric BIST for Production Testing PLLs with Picosecond Precision", *Proc. of ITC*, November 2010.

[12] Signalyzer datasheet, http://www.xverve.com

[13] "LMK03000 Family," National Semiconductor, 2009 http://www.national.com/ds/LM/LMK03000.pdf

[14] Table 4-86, Stratix II GX Device Data Sheet, Altera Corp., June 2009

[15] S. Sunter, A. Roy, "Structural Tests for Jitter Tolerance in SerDes Receivers", *Proc. of ITC*, Oct. 2005.

[16] J. Galloway, R. Caplan, "SOC-PLL design requires trade-offs", *EDN*, pp. 40-43, December 2, 2010 http://www.edn.com/file/25736-SOC_PLL_design_requires_trade_offs_PDF.pdf