

Variable Delay of Multi-Gigahertz Digital Signals for Deskew and Jitter-Injection Test Applications

D.C. Keezer¹, D. Minier², P. Ducharme²

1- Georgia Institute of Technology, Atlanta, Georgia USA

2 – IBM, Bromont, Canada

Abstract

The ability to precisely control the timing of digital signals is especially important for multi-GHz testing applications where errors are measured in picoseconds or even 100fs. While many solutions exist for continuous clock-type signals, delay of wide-bandwidth data signals is not so easy. In this paper we introduce a novel technique for adjusting the delay of ~7Gbps data signals on a picosecond scale without significant distortion. The approach is based on a timing/amplitude dependency effect observed in a variable-gain SiGe buffer. A prototype is demonstrated with a variable delay range of about 50ps. This circuit is enhanced by adding a “coarse” delay section, including four 33ps steps, to provide the desired total range of ~140ps. The end application requires several of these circuits for deskewing parallel buses of 6.4Gbps ATE signals. The circuit is also useful for injecting a variable amount of jitter, limited by the fine-delay adjustment range.

1. Introduction, Background, Motivation

When designing high-speed digital circuits there is often a need to adjust the phase or “delay” of one signal relative to another. For example, a clock signal may need to be aligned to the center of the data “eye” at a receiving register, as shown in **Fig.1**. Since it is generally easier to adjust a constant-frequency (narrow-bandwidth) clock signal, rather than the wide-bandwidth data signal, the solution usually involves adjusting the clock phase. Many VCO and PLL or DLL techniques are widely used for this purpose [1-8].

However, the more general (and more difficult) problem of aligning multiple *data* signals is not so easily solved for multi-Gbps signals. When the data bus contains multiple parallel signals which are approximately synchronized to a common clock, each signal must be “deskewed” before clocking the receiving register. This situation is closely related to the general problem of aligning multiple ATE signals at the DUT inputs, as shown in **Fig.2**. For this example, four high-speed ATE signals are produced with a small amount of timing skew between the channels (top part of the figure). In some ATE these small timing errors can be corrected by changing the programmed delays for each channel. The result is a realignment of the figure. The precision of

alignment is primarily limited by the ATE timing resolution and linearity. In our application, which uses “SB6G” 6.4Gbps signal sources on the Teradyne UltraFlex ATE, the resolution is on the order of 100ps. This resolution is adequate for some applications such as PCIexpress, where each “lane” operates as a separate communication channel, and can tolerate channel-to-channel skew. However for other applications, such as HyperTransport 3, the parallel data must be aligned more precisely to a common clock signal. In these situations, we need to have picosecond control of the relative delays.

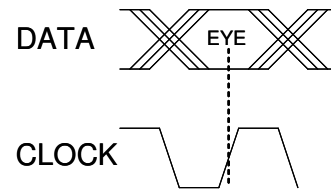


Fig.1 – Phase adjustment of the clock signal to the center of the data eye.

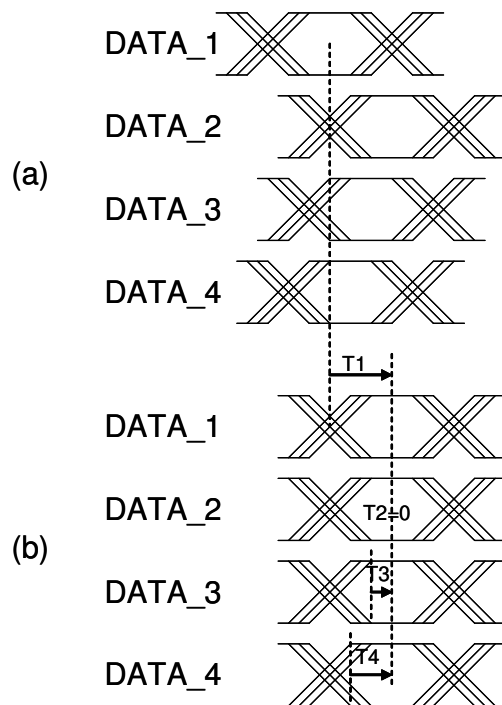


Fig.2 – Deskewing a parallel data bus, (a) input data bus with skew, (b) output data after adding small delays to each channel.

In this paper, we specifically target an ATE requirement for deskewing multiple 6.4Gbps data signals, which otherwise have a limited deskew resolution on the order of 100ps. Internally, the ATE can be programmed to step each of the data signals by these ~100ps increments in order to achieve approximate timing alignment at the DUT. However, with a bit-period of only 156ps at 6.4Gbps, the ~100ps resolution is not small enough for parallel-synchronous applications. Instead, we need ~1ps (or better) *resolution*, <5ps channel-to-channel skew *accuracy*, and minimal added jitter (<25ps). The problem is further complicated by a requirement to handle a wide range of frequencies (from <1Gbps to 6.4Gbps).

A method for adjusting multi-Gbps data delays is presented in the **Section 2**. This method achieves the required fine delays with <1ps resolution, but has limited range. The range limitations are solved by adding a coarse-delay circuit, described in **Section 3**. The combined prototype circuit performance is described in **Section 4**. The additional application of Jitter-injection is demonstrated in **Section 5**.

2 Variable Delay Adjustment Method

To make fine adjustments in the delay of multi-Gbps data channels, we need a circuit that can transmit up to 7Gbps without significant distortions (such as increased jitter), yet provide a voltage-adjustability of delay on a picosecond scale. To obtain this fine timing adjustment, we use a variable-amplitude SiGe differential buffer as shown in **Fig.3**. This commercially-available chip can transmit signals up to 12Gbps, and provides amplitude adjustment between 100mV and 750mV.

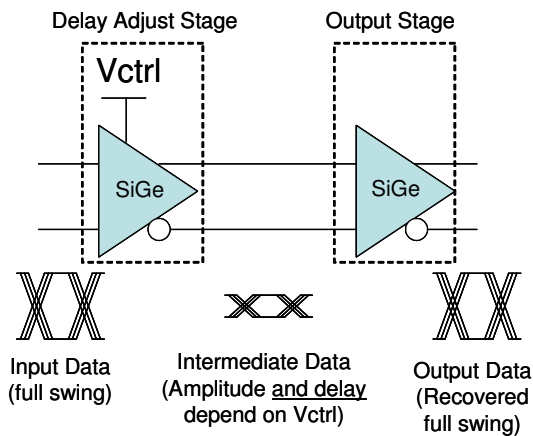


Fig.3 – Variable delay circuit.

Actually two such buffers are shown in the figure. The first one provides the variable-delay feature and the second one recovers the full logic amplitude. The Vctrl control voltage determines the amplitude of the first stage buffer. In an ideal buffer, adjustment of the amplitude would not change the signal timing. Experimentally we noticed a small (~10ps) skew range that depends on the adjusted signal amplitude. Furthermore, we found that the relationship between signal amplitude and timing

skew was approximately linear. Therefore this adjustable-amplitude buffer can also serve to adjust timing skew in the data signal that it passes.

To understand why there might be a small change in delay as the buffer amplitude is adjusted, consider **Fig.4** and **Fig.5**. Here the input to the gate is shown on the top. A small amplitude output signal may have a propagation delay of T_1 , as shown in the figure. If the amplitude is increased, the signal will tend to take a longer time to reach the 50% threshold (due to the limited slew rate of the buffer). Therefore $T_2 > T_1$ and we should expect an adjustable skew range of $T_2 - T_1$, which we have seen to be about 10ps for this buffer. The skew range is propagated through the output stage so that $T_4 - T_3$ is also about 10ps.

Other factors may influence the skew range, especially when the buffers are cascaded in multiple stages (as we do in our prototype). In this case, the response of the input of the buffer to changing amplitudes will also play a role in the effective skew range.

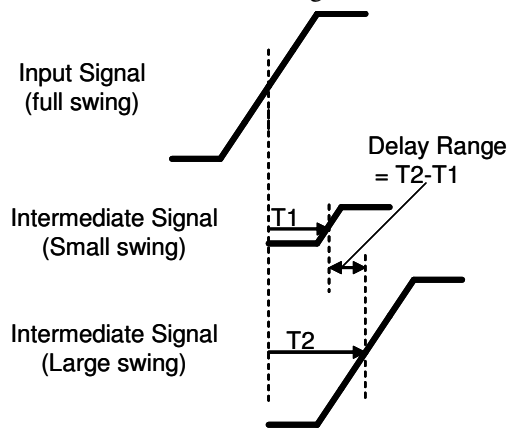


Fig.4 – Adjustment of delay using a variable-gain buffer with amplitude-dependant propagation delay.

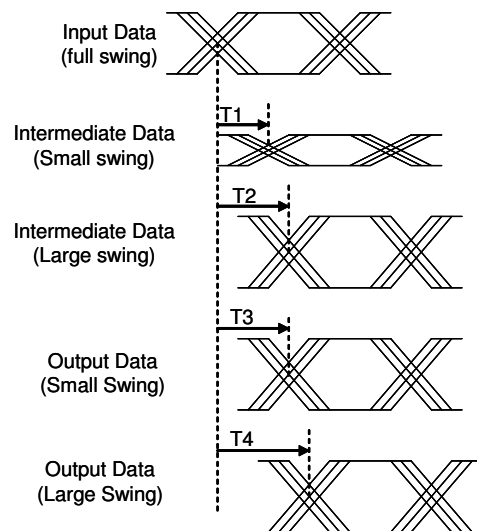


Fig.5 – The data delays at the intermediate stage (T_1 and T_2) depends on the programmed amplitude. The final output delays (T_3 and T_4) also depend on the amplitude control, but the output amplitude itself is fixed.

For our ATE deskew application, the 10ps range is not sufficient. Therefore we cascade four adjustable buffers as shown in **Fig.6** (along with a fifth “output” buffer). The combined effect of the four stages increases the expected range to about 40ps. For simplicity we use a common Vctrl for all four buffers.

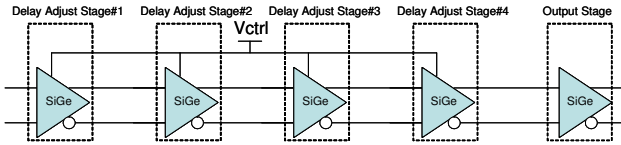


Fig.6 – 4-stage fine-adjustment delay circuit with output stage for amplitude recovery.

Experimentally we found that the range was slightly larger than this, as shown in **Fig.7**. Here the change in output delay is plotted against the programmed control voltage (Vctrl) across a 1.5V range. The function is approximately linear throughout much of the mid-range, with changes in slope near the extremes. Given these measurements, we can determine an appropriate control voltage for any desired delay within this ~56ps range. In our target application, Vctrl will be provided using a 12-bit DAC, so sub-picosecond resolution will be achievable.

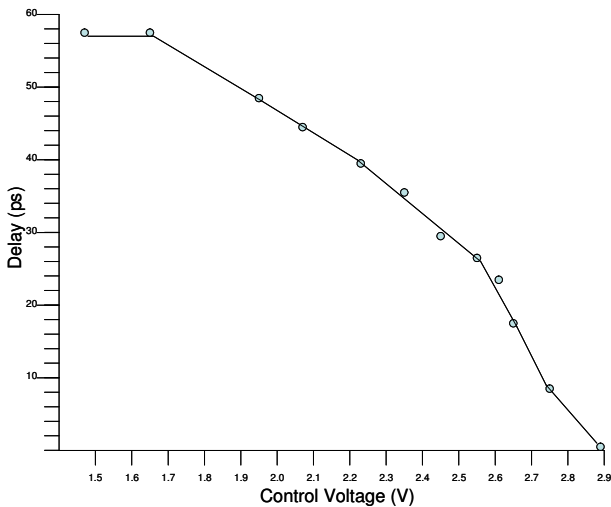


Fig.7 – 4-stage fine-adjustment circuit, typical measured Delay vs. Control Voltage (Vctrl).

3 Coarse Delay Selection

Even though we achieved a fine adjustment delay range of over 50ps, this is still not sufficient for our application (which requires 120ps). In theory we could cascade two or more of these circuits to obtain the desired range. However, in practice we must be concerned with the undesirable noise and jitter added by each stage. For this and other practical reasons, we decided to use a coarse delay section that requires only two levels of logic. The circuit, shown in **Fig.8**, uses a SiGe 1:4 fanout buffer to create 4 parallel copies of the high-speed data signal. Each one passes through a differential pair transmission line with a controlled length. The lengths are designed to add increments of 33ps to the four signals before reaching a 4:1 multiplexer. Two digital select lines determine which of the four is passed through to the output of the coarse delay circuit.

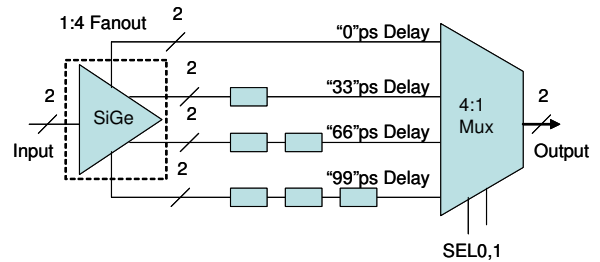


Fig.8 – 4-tap coarse delay circuit with 33ps steps.

The measured performance of this circuit is shown in **Fig.9**. This figure shows an expanded time-scale portion of the data “eye” near the 50% threshold for all four of the taps. The signals are overlaid for comparison, and the measured values of 0ps, 33ps, 70ps, and 95ps are noted in the figure. The deviations from the ideal 33ps increments are only a few picoseconds.

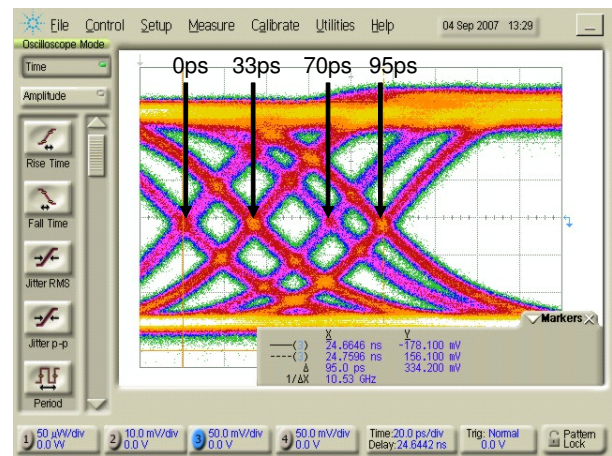


Fig.9 – Measured coarse delays (0ps, 33ps, 70ps, 95ps).

The combined prototype delay circuit is shown in **Fig.10**. This is obtained by cascading the coarse and fine delay sections. Desired delays between the coarse increments are obtained by adjusting the fine-delay using Vctrl. Also, delays beyond 95ps are obtained by increasing the fine delay section up to about 45ps. Therefore the total range of the combined circuit is about 140ps, and satisfies the application requirement of 120ps. The performance of the complete prototype is shown in Section 4.

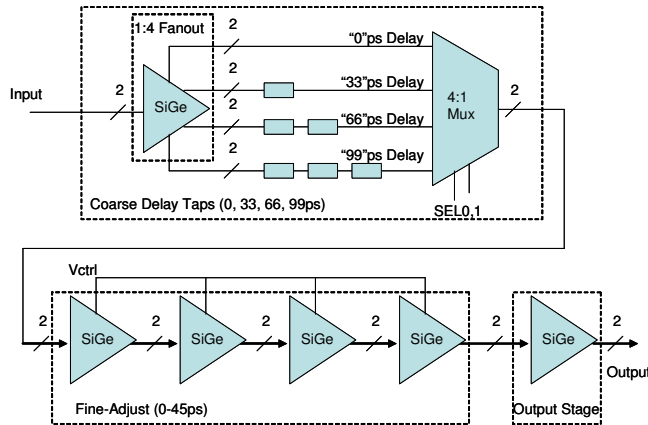


Fig.10 – Combined coarse/fine--adjustment circuit.

A photograph of a 2-channel prototype PCB is shown in **Fig.11**. The board size is limited by the SMA connectors which are included for evaluating the circuit performance. Some additional features (such as buffered test points) are also included for the experimental evaluations, so the final application circuit size will be much smaller than this prototype. Size is important since we need to deskew buses with 8 differential channels, and must fit the electronics in a very limited space under the Device Interface Board (DIB).

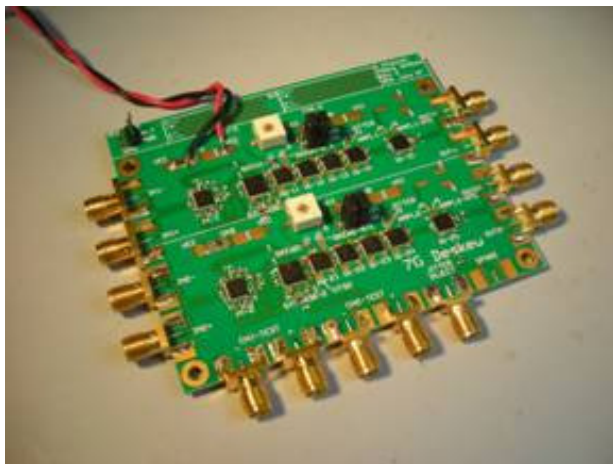


Fig.11 – Photograph of the 2-channel prototype.

4 Prototype Performance

Aside from the delay circuit’s ability to span the required delay range, we also require that it be able to pass the high data-rate signals (up to 6.4Gbps) with minimal distortion. In particular, since the delay circuit contains 7 active components, it might be susceptible to jitter (and other noise effects) which accumulate as the signal passes through all the 7 stages. Therefore, we measured the peak-to-peak jitter of the output signal under a variety of test conditions. We also measured the delay range of the fine adjustment section, as a function of data-rate.

An example measurement is shown in **Fig.12**, where two 4.8Gbps data eyes are overlaid. The left-most eye crossing corresponds to the output signal with *minimum* fine delay, while the next crossing is the same output except with Vctrl set to produce the *maximum* fine delay. At this data rate (4.8Gbps) we see a fine delay range of 49.5ps. Also noted in the figure is the peak-to-peak jitter value of 18.5ps which is about 7ps larger than the input reference signal. We found this small (~7ps) increase in total jitter to be typical for the circuit for data rates below 6Gbps. Slightly more jitter was observed above 6Gbps.

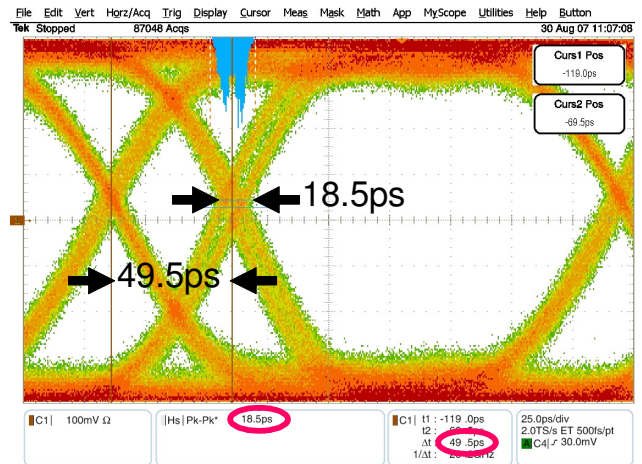


Fig.12 – Example delay measurement at 4.8Gbps showing a fine-delay range of 49.5ps and TJ=18.5ps.

To demonstrate the performance of the delay circuit at the target data rate of 6.4Gbps, we show a DUT output signal in the top part of **Fig.13**. This signal had approximately 26ps of peak-to-peak jitter, as shown in the figure. The bottom plot shows the signal after passing through the delay circuit, with only about 13ps of added jitter. The amplitude attenuation is due to series resistors added for measurement convenience and is not a concern for our applications.

In order to check the circuit performance above 7Gbps (the limit of our signal generator for NRZ data), we used RZ clock signals at rates up to 6.8GHz. One example measurement is shown in **Fig.14** for a 6.4GHz clock pattern. This signal is essential twice as fast as the 6.4Gbps maximum NRZ rates of our application, in some ways comparable to a 12.8Gbps NRZ rate. Even so, the prototype worked well, with a 32.5ps fine-delay range.

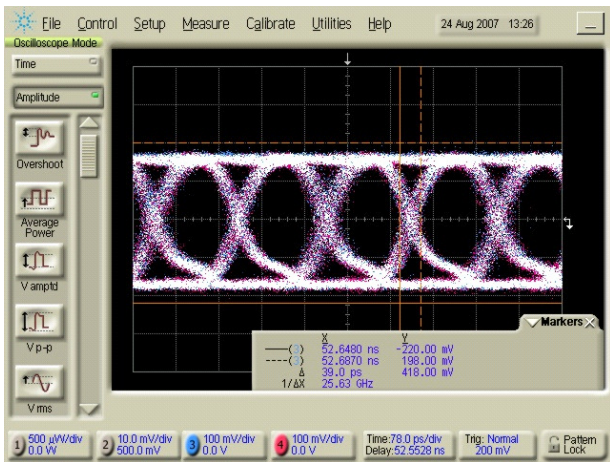
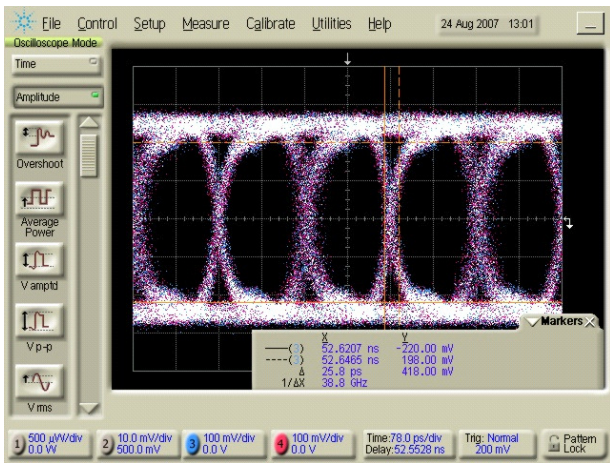


Fig.13 – Example 6.4Gbps Data Eyes. The top plot is the Input reference ($TJ=26ps$), and the bottom is the delayed output with only slightly higher jitter ($TJ=39ps$).

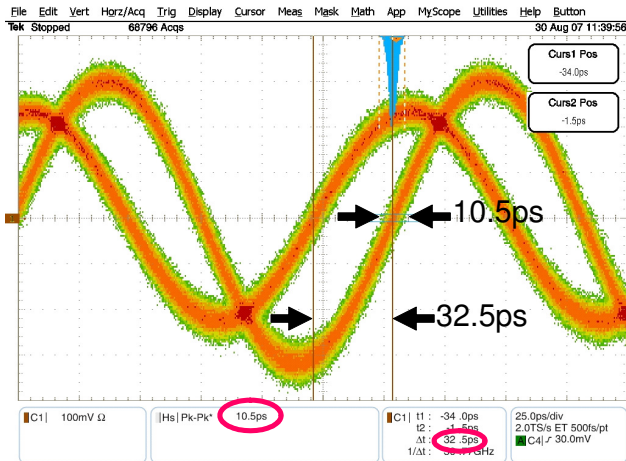


Fig.14 – Example delay measurement at 6.4GHz showing a fine delay range of 32.5ps and $TJ=10.5ps$.

A summary of the fine-delay performance is shown in **Fig.15**. Here the delay range is plotted as a function of RZ clock frequency up to 6.8GHz. The top curve shows the measurements from our 4-stage prototype. For comparison, the bottom curve shows an early 2-stage circuit. The 2-stage circuit worked well up to 2.6GHz (5.2 Gbps effective NRZ rate), but had a much smaller delay range as the frequency increased, becoming ineffective beyond 6Gbps. The 4-stage circuit shows similar trends, except that its usable range extends beyond 6.4GHz (12.8Gbps effective NRZ rate). Recall that we need about 33ps of range to cover the coarse delay steps.

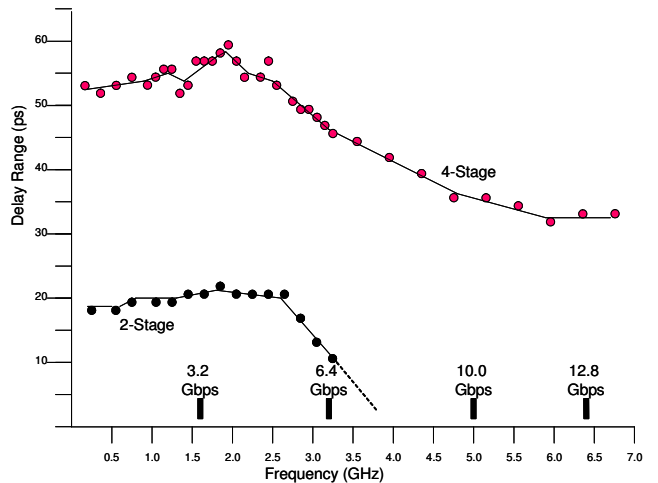


Fig.15 – Delay range vs. Clock Frequency for both a 2-stage and a 4-stage fine-adjustment circuit.

5 Jitter Injection

In the previous sections we saw how the fine delay circuit could be used to deskew multi-Gbps signals with minimal added jitter. However, in some testing applications we actually want to *add* a controlled amount of jitter (for example to test input jitter tolerance). With a very minor modification, the fine delay portion of our prototype can be used to “inject” jitter onto a test signal. This is accomplished by AC-coupling a voltage noise source to the Vctrl signal which determines the fine delay adjustment. If this voltage changes, then the delay also changes. We are then able to convert a voltage noise source to timing jitter injected onto a multi-GHz signal.

An example of injecting jitter is shown in **Fig.16**. The top part of the figure shows a reference signal at 3.2Gbps with total jitter of about 28ps. Using an external signal generator with 900mV (peak-to-peak) Gaussian voltage noise AC-coupled to our Vctrl control, we obtain an output signal jitter of 69ps (for an increase of 41ps). By adjusting the noise source amplitude, we can control the resulting amount of added jitter, as shown in **Fig.17**.

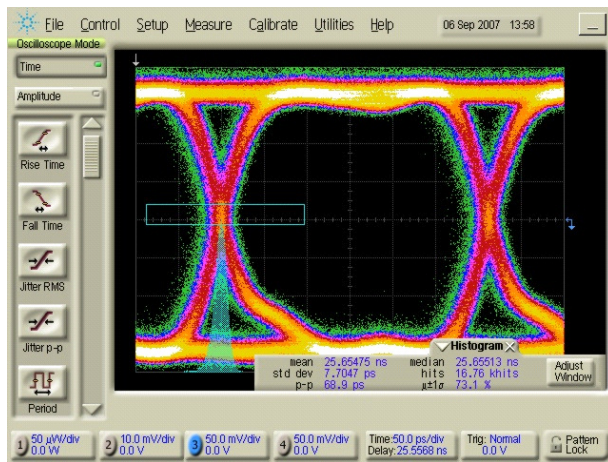
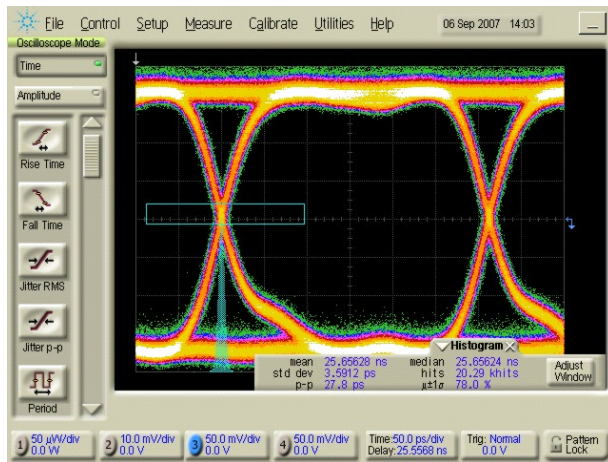


Fig.16 – Jitter injection at 3.2Gbps – Top is the input reference (TJ=28ps), Bottom is the output with 900mV of voltage noise causing increased jitter (TJ=69ps).

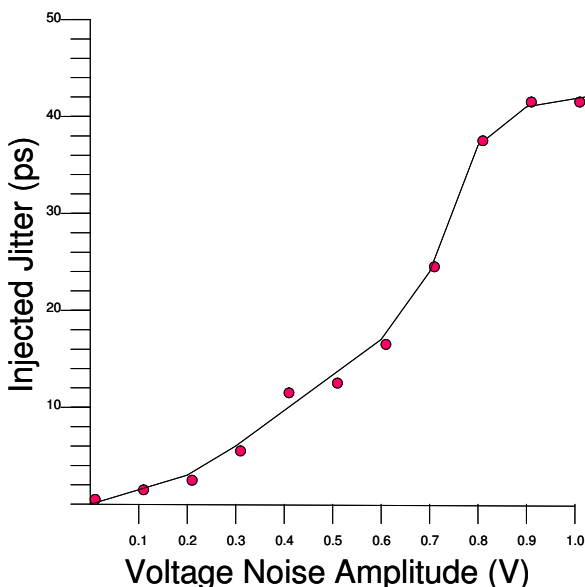


Fig.17 – Added jitter vs. applied voltage noise.

6 Conclusions and Future Directions

We have presented a technique to adjust the delay of multi-gigahertz digital signals with sub-picosecond resolution, through a ~50ps “fine adjustment” range. This was accomplished by taking advantage of a small (~10ps) timing dependency (vs. amplitude) effect in an otherwise high-performance SiGe variable-amplitude buffer. By cascading these buffers, the larger delay range (~50ps) was achieved. Further extension of the delay range was demonstrated using selectable delay lines that formed a “coarse delay” section with 33ps steps. The resulting 2-channel prototype was demonstrated up to, and beyond, its intended maximum rate of 6.4Gbps, with an effective range extending to 12.8Gbps.

We have recently built a 4-channel version of this circuit for deskewing parallel data buses from an ATE.

While the work presented here was targeted at ATE deskewing applications, the authors envision that broader applications may be found in many situations where existing signals need small timing adjustments.

7 Acknowledgements

This work was conducted as a joint R&D project between Georgia Tech and IBM, Canada. The system-level experiments were conducted using equipment at the IBM, Canada facility in Bromont.

8 References

- [1] M. Shimanouchi, “Periodic Jitter Injection with Direct Time Synthesis by SPP ATE for SerDes Jitter Tolerance Test in Production,” Proc. of the Intl. Test Conf. (ITC’03), pp. 48-57, 2003.
- [2] H.W. Johnson, M. Graham, High-Speed Digital Design, Prentice Hall, 1993.
- [3] D.C. Keezer, D. Minier, M. Paradis, F. Binette, “Modular Extension of ATE to 5 Gbps,” Proc. of the IEEE Intl. Test Conf. (ITC’04), pp.748-757, Charlotte, Oct. 2004.
- [4] D.C. Keezer, D. Minier, P. Ducharme, “Source-Synchronous Testing of Multilane PCI Express and HyperTransport Buses,” *IEEE Design and Test of Computers*, vol. 23, no. 1, pp. 46-57, January 2006.
- [5] A. Borgioli, Yu Liu, A.S. Nagra, R.A. York, “Low-loss distributed MEMS phase shifter,” *Microwave and Guided Wave Letters*, Volume: 10 , Issue: 1 , Jan. 2000, Pages: 7 – 9.
- [6] A.S. Nagra, R.A. York, “Distributed analog phase shifters with low insertion loss,” *IEEE-MTT*, Volume: 47, Issue: 9, Sept 1999, Pages: 1705 - 1711.
- [7] N.S. Barker, G.M. Rebeiz, “Distributed MEMS true-time delay phase shifters and wide-band switches,” *IEEE Trans. Microwave Theory Tech.*, vol. 46, no.11, pp. 1881-1890, Nov. 1998.
- [8] H. Zhang, A. Laws, K.C. Gupta, Y.C. Lee, V.M. Bright, “MEMS variable-Capacitor Phase shifters Part I: loaded-line phase shifter,” *Wiley Periodicals, Inc. Int RF and microwave CAE* 12:321-337, 2003.