Synchronizing Multiple ADC08xxxx Giga-Sample ADCs

1.0 Introduction

The ADC08xxxx giga-sample family of analog-to-digital converters (ADCs) make the highest performance data acquisition systems possible at very low power – which is often the limiting factor in such systems. There is often a need to sample multiple signals in parallel and time-alignment of these multiple channels is quite a challenge in such "waveform recording" applications. This is a recognized system level issue and this note is not about how to completely align multiple data acquisition channels.

ADCs such as the National Giga-sample family, that "demux" the data in order to reduce the output data rate, introduce an additional consideration to this task. This is because there is now added uncertainty about the correspondence between the CLK input and the DCLK output of the 2 or more ADCs in a system – regardless of whether they are sampling in parallel or in interleaved manner. The ADC08xxxx family of products offers a DCLK_RST input that allows synchronization of multiple ADCs to remove this uncertainty. This is a difficult task to accomplish in a board design, however, due to the very high clock speeds involved. This note provides additional details of how one might accomplish this task.

2.0 The Multi-Channel Application

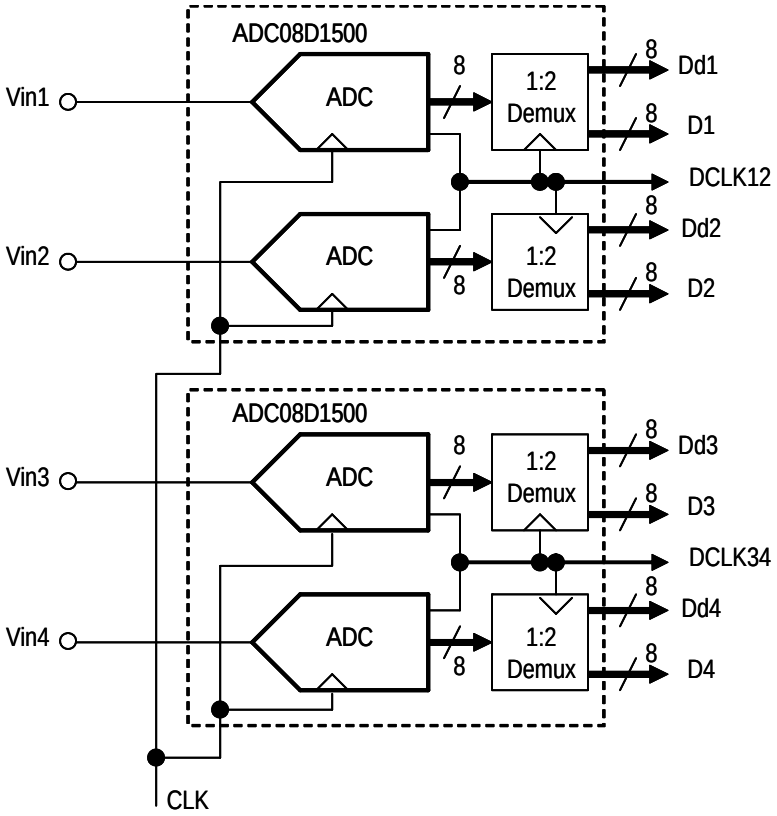

Figure 1 – A 4-channel 1.5Gs/s Digital Scope Application of the ADC08D1500

Figure 1 shows a simple block diagram of two ADC08D1500 dual converters being used in a 4-channel digital oscilloscope application. Sampling clock is shared amongst the four channels. However, there are two different DCLKs from the two chips – DCLK12 and DCLK34. Figure 2a shows how, upon each power-up, DCLK12 and DCLK34 may have a different time relationship with the common sampling clock; or it is equally possible that they may not – thus the uncertainty. Figure 2b shows how by synchronizing with DCLK_RST, DCLK12 and DCLK34, and their associated data busses, can be time-aligned in units of CLK periods. In these figures the SDR (single-data-rate) clock operation is shown and OutEdge=1 is assumed. The effect would be the same in other DCLK configurations.

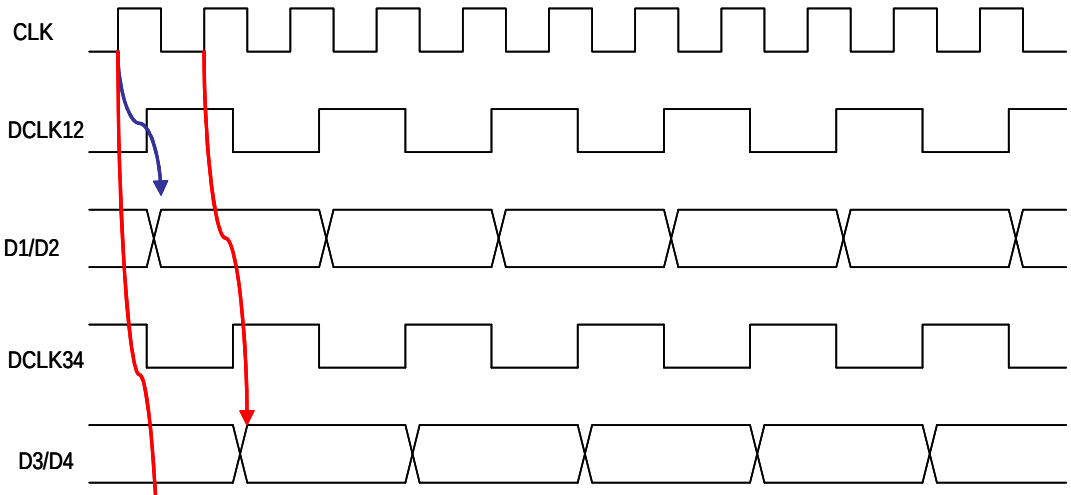

Fig. 2a – Two devices' outputs out of synchronization

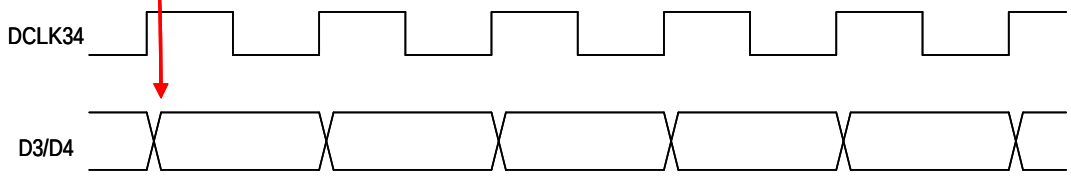

Fig. 2b – Two devices' outputs in synchronization

[Note: SDR mode operation shown; OutEdge=1; Effect of Tod variation ommitted]

Figure 2 – The multichannel synchronization problem

Figures 2a and 2b omit an important timing consideration since they are drawn roughly in units of CLK cycles: the CLK-input-to-Data-Output delay Tod. Tod can vary significantly on a given device and also from device-to-device. In fact, depending on CLK frequency being used, maximum possible Tod variation can be greater than one CLK cycle. For this reason, the data busses from each device should be captured into the ASIC or FPGA with the DCLK from the same device. In the FPGA, separate Data FIFOs should be employed for each channel and the data streams from the two discrete devices (i.e., D1 and D2 versus D3 and D4) should be deskewed by varying the relative Read Clock timing of the FIFOs.

3.0 DCLK Reset Function and Timing

Once the system has powered-up and stabilized, the multiple ADC devices should be synchronized by resetting the DCLK generation logic with the DCLK_RST signal. This has to be done synchronous with the CLK input and thus there is a demanding timing requirement. Figure 3 shows the DCLK_RST function and timing.

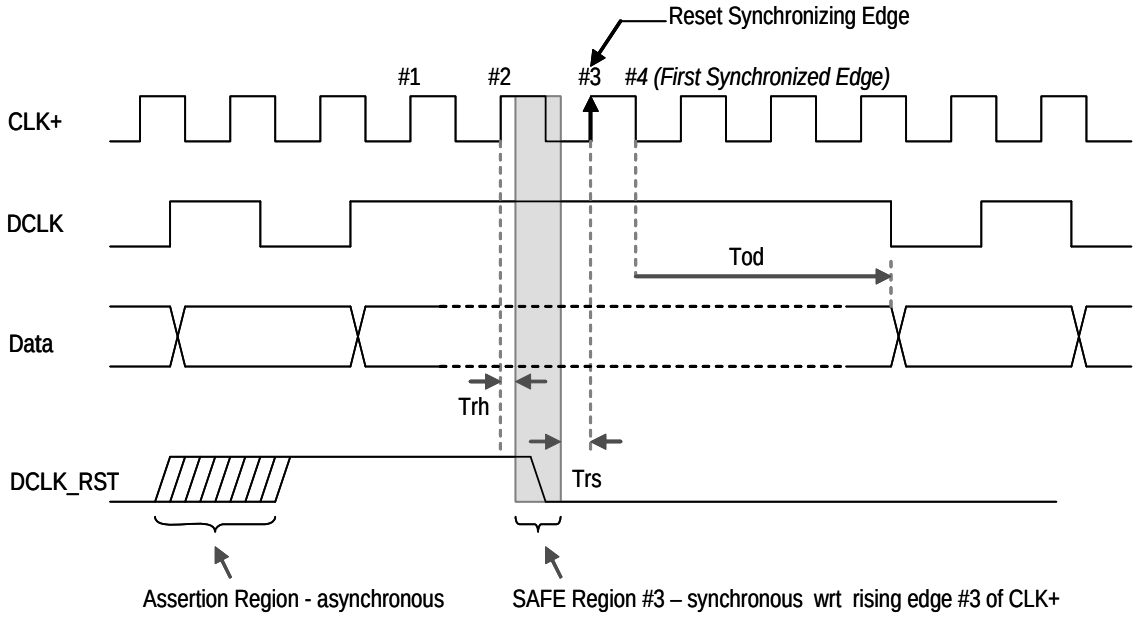

Figure 3 – DCLK Reset Function and Timing

When the DCLK_RST is asserted by the user, the device disables the DCLK output. If the DDR mode of operation is selected, then DCLK is held in the positive state. In the SDR mode, the state of the disabled DCLK depends upon the OutEdge pin or register bit. DCLK and the Data bus are enabled again upon deassertion of DCLK_RST.

DCLK_RST can be asserted asynchronously. Minimum duration of DCLK_RST is specified in the datasheet (Trpw). The device times the deassertion edge of DCLK_RST with the CLK input and thus DCLK_RST signal must meet setup and hold time requirements with respect to it. In Figure 3, deassertion of DCLK_RST will be captured by the device with CLK (rising) edge #3 if (a) DCLK_RST signal deasserts no later than Trs before edge #3 AND (b) no earlier than Trh after edge #2. In other words, DCLK_RST deassertion edge must occur within the SAFE region indicated in the diagram. If these conditions are met at the pins of both devices, then edge #4 will be the first “synchronized edge” and will provide the synchronized DCLK output after delay Tod.

SAFE Region Width = Tcyc – (Trs + Trh)

[where, (Trs + Trh) = Transition Region]

Condition (a) is the setup condition and condition (b) is the hold condition. If either of these conditions is violated then the two devices may synchronize with different edges of CLK and not achieve synchronization amongst them.

Trs and Trh are not production tested on the ADC08xxxx devices. Therefore, their minimum values cannot be guaranteed. Typical values for these parameters at room temperature and nominal Supply voltage are:

Trs = 35 psec; Trh = -5 psec => Transition Region = 30 psec

Limited testing of these devices and simulations indicate that over a range of operating conditions and process spread the minimum values for these parameters can be expected to be:

Trs = 80 psec; Trh = 40 psec => Transition Region = 120 psec

It must be stressed that these are not guaranteed specification values, but rather design guidelines.

These values indicate following safe region widths as a function of CLK frequency:

CLK Frequency	Tcyc	SAFE region
500 MHz	2.0 ns	1.88 ns
750 MHz	1.33 ns	1.21 ps
1.0 GHz	1.00 ns	880 ps
1.25 GHz	800 ps	680 ps
1.50 GHz	667 ps	547 ps

It should be clear from this table that synchronizing multiple ADCs above Fs = 1 GHz is a challenging feat. Figure 4 shows the general scheme that can be used to generate the DCLK_RST signal.

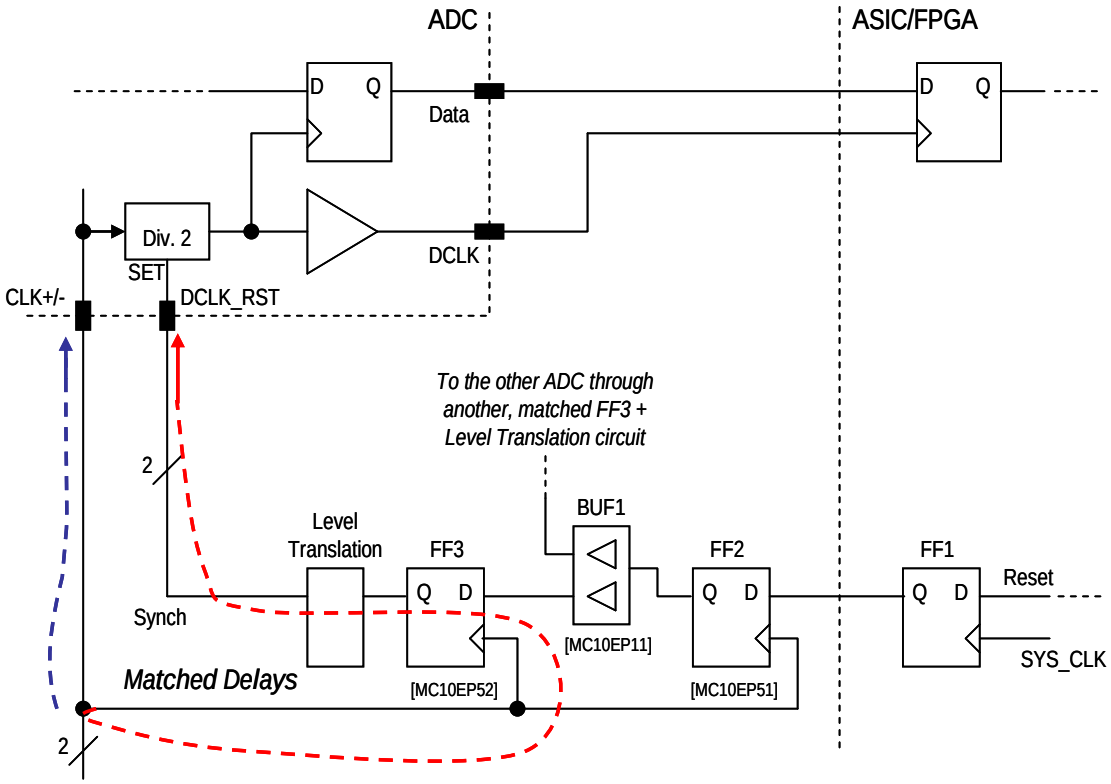

Figure 4 – Generating the DCLK Synch Signal

The reset generation circuit generally originates in an ASIC or FPGA that is also the device receiving the ADC's output data. Therefore, the DCLK signal is available to time the DCLK_RST signal. However, as Figure 3 shows, DCLK is stopped during the reset process - making it impractical to time DCLK_RST with DCLK. There

are digital PLLs available in many FPGA and ASIC technologies that can be employed to generate a replica of DCLK that can remain operational even when DCLK signal is momentarily stopped. These DPLL facilities also make it possible to skew one signal with respect to another, so the replica DCLK's phase could be adjusted to make the DCLK_RST signal's timing better match the ADC's setup time requirements.

As a general case, however, we must assume that for the purpose at hand the reset generation circuit in the ASIC/FPGA is timed by a System Clock (SYS_CLK) that is asynchronous with respect to the ADC's CLK input. Therefore, in Figure 4 this signal is double-synchronized by FF2 and FF3 by the CLK signal. FF2 and FF3 are high-speed, differential PECL flip-flops – such as the MC10EP52 from ON Semiconductor. These devices are capable of operating at frequencies up to 4 GHz. Their typical CLK->Q propagation delay (Tpd) of 330psec is guaranteed to vary no more than 160 ps over process and temperature. With this amount of variation, it is possible to meet the ADC's Trs/Trh setup and hold time requirements. Note, however, that it requires very careful delay adjustment and matching in the board design.

The output of the second PECL flip-flop has to be level-shifted to match the ADC's input requirements. The DCLK_RST input on the ADC08xxxx devices is a digital CMOS input. Scaling the PECL signal to CMOS, even at 1.9V Supply levels of the ADC08xxxx family, requires active circuitry because of the signal gain that is required in addition to the level shift. A conceptual circuit is shown in Figure 5 (this circuit has *not* been built and verified). This circuit requires a very high-speed comparator that can support a signal rise time of less than 1ns.

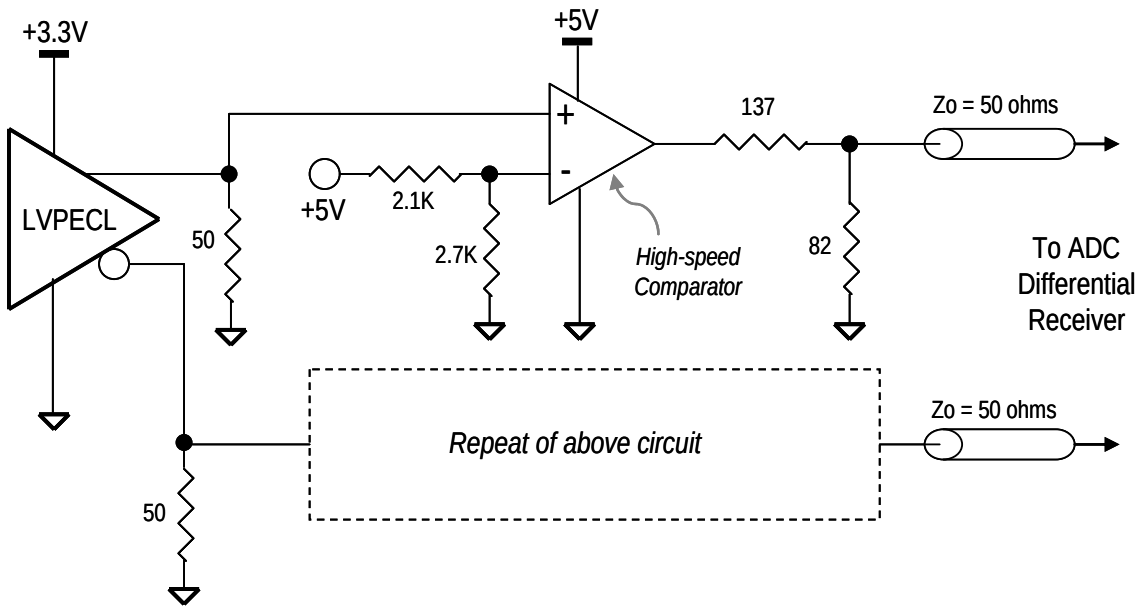

Figure 5 – Conceptual LVPECL-to-CMOS level translation circuit

4.0 Improved DCLK_RST Timing

The latest members of the ADC08xxxx family, the ADC08D1020 and ADC08D1520, have improved the DCLK_RST input in order to make it easier for the board design to accomplish the very tight reset timing explained in Section 3.0. The DCLK_RST input can be optionally configured (see footnote*) as a differential input. In this differential configuration, the DCLK_RST+/- input is better matched to the CLK+/- input and the board level CLK_SYNC signal's PECL voltages can be level-shifted with a simpler, passive circuit. The result is significantly better timing margin as shown here:

Tcyc @ 1.5GHz =	667 ps	
Transition Region =	-120 ps	

Allowance for timing variation =	547 ps	(SAFE Region)
D-Flipflop timing variation =	-160 ps	
Trace mismatch and delay variation =	- 90 ps	
Level translator and other =	- 50 ps	

System Margin =	247 ps	

On the ADC08D1020/1520, the DCLK+/- differential input is provided on pins 15(+) and 14(-) and the electrical specifications are as follows (the signal must be dc-coupled and 100-ohm termination resistor must be applied as close as possible to the pins):

Differential Amplitude (Vid) =	0.4V to 2.0V (0.6V nominal) peak-to-peak
Common Mode Voltage (Vcm) =	1.2V nominal (1.0V to 1.4V worst-case)

Figure 6 shows the level translator circuit required to adapt the PECL output of the D-FF to the LVDS input of the ADC08D1020/1520.

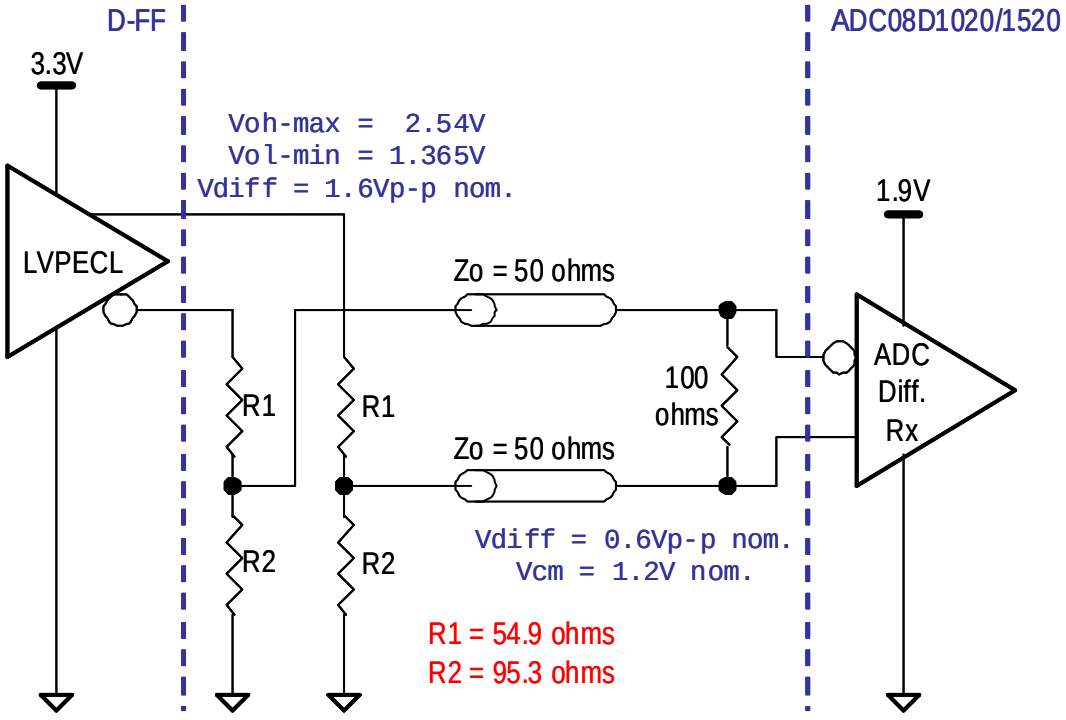

Figure 6 – PECL to ADC08D1020/1520's "LVDS-Like" Differential Input

* **Pin 52 = Low** makes pin 14 the DCLK_RST- input. FSR input is now internally set at High level and the Serial Interface is not enabled. To enable the Serial Interface, set **Pin 41 = Low**.

5.0 Alternative method for resetting DCLK

In those board designs where it may be possible to disable and enable the CLK signal, an alternative reset scheme is possible. Figure 7 illustrates this scheme.

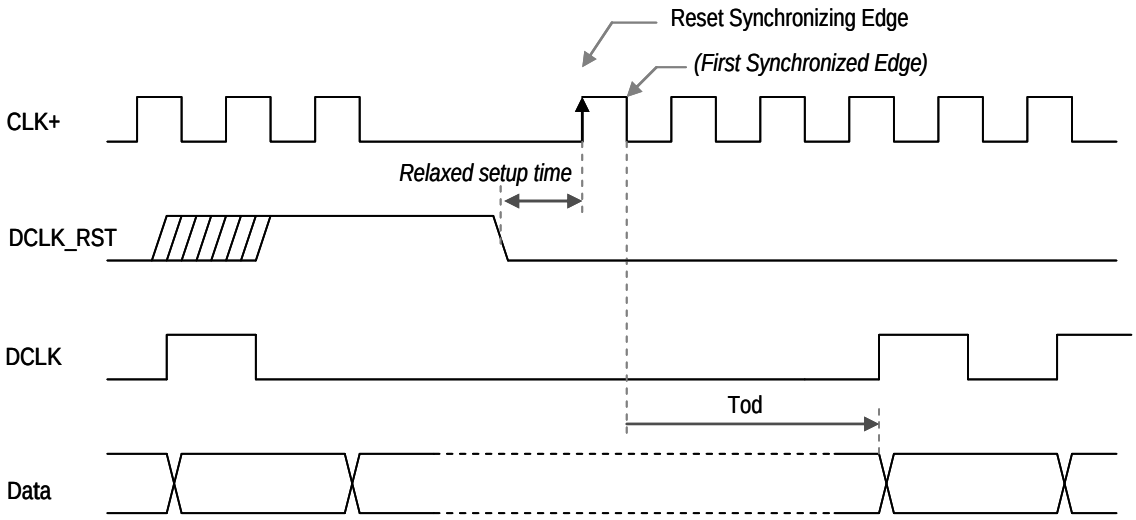

Figure 7 – Synchronizing by stopping and restarting the Clock

Essentially, once DCLK_RST is asserted you stop the CLK signal, deassert the DCLK_RST and then start the CLK signal again allowing generous amount of time to satisfy the Trs setup time requirement. The challenge with this scheme, however, is the ability to cleanly start the CLK+/- signal – i.e., without violating the CLK Pulse Low and High (Tcl and Tch) time requirements.

The following rules must be observed when implementing this scheme:

1. Once the DCLK_RST signal is asserted, CLK+/- must cycle (low-then-high, or high-then-low) at least once before being disabled.

2. The CLK+/- signal, when disabled, may be held in the low or high state

3. The CLK+/- signal must be disabled less than **30 ns** if the device's duty cycle stabilizer function is being used.

4. No narrow pulses are allowed when the CLK+/- signal is enabled again - i.e., the minimum clock pulse (Tpl/Tph) timing requirements must be met.

5. To prevent the AC coupled ADC CLK+/- inputs from de-biasing while the CLK is stopped, the AC coupling capacitors should be 100 nF or higher.

6.0 Conclusion

Synchronizing multiple ADCs is a difficult problem in very high speed multi-channel data acquisition systems. It is shown that with careful design this task can be accomplished with National's Gigasample family of ADC products and the latest members of this family have improved circuitry to ease this task.