Several Key Issues on Implementing Delay Line Based TDCs using FPGAs

Jinyuan Wu

***Abstract*— This paper discusses implementation of the Wave Union TDC, a novel scheme of FPGA TDC to improve time measurement precision using multiple measurements, along with several other topics in FPGA delay line based TDCs. FPGA specific issues such as considerations on the delay line choice in different FPGA families and encoding logic are first examined. Next, common problems for both FPGA TDCs and ASIC TDCs such as schemes of coarse time counter implementation, bin-by-bin calibration and noise issues due to single ended signals are discussed. Several resource/power saving design approaches for various processing stages are described in the document.**

***Index Terms*— Fast Timing, Front end electronics, Time to digital converters**

I. Introduction

CHAIN structures existing in FPGA families can be used for time-to-digital conversion (TDC) purposes[1-10]. Given the structures inside the devices, the most common architecture for the FPGA TDC consists of a chain delaying the input hit signal followed by a register array driven by an on-chip clock as shown in Fig. 1. Similar to implementing TDCs in ASICs, implementing TDCs in FPGAs is an analog design job that requires careful considerations beyond typical digital design practices. Although designers usually cannot redesign analog properties of the circuits or alter the logic elements and routing matrices inside FPGA devices, deriving a set of strategies to overcome the drawbacks due to undesired pre-existing conditions can be useful.

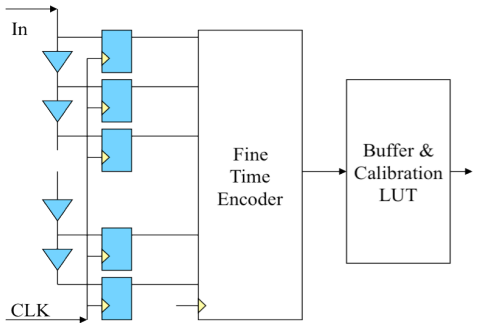

Fig. 1. Delay chain/register array structure and FPGA TDC block diagram

A block diagram of the FPGA TDC as shown in Fig. 1 is used as a model for discussion. The carry chain and register array structure is followed by an encoder to convert the register array bit pattern into the raw TDC timing bin number. The raw TDC numbers are temporarily stored in a memory buffer and upon reading them out, the raw bin numbers are used to address the calibration look-up table (LUT) to find calibrated fine time.

This paper describes the Wave Union TDC, a novel scheme of FPGA TDC to improve time measurement precision using multiple measurements [7-9] and also serves as a summary of various technical issues that may be encountered during the design of FPGA TDCs. The remaining parts of the paper are organized as the following:

In Section II, the design of the delay chain is discussed. Contrary to regular digital logic designs that can be easily ported from one FPGA device to another, TDCs rely on internal gate delay and therefore a correct choice of an FPGA family and internal clock frequency is crucial before any design work can begin. Additionally, the wave union launcher implemented in the delay chain structure is also discussed.

In FPGAs, an imperfection of the thermometer bit pattern, the "bubble" can happen fairly often. An appropriate logic design in the encoder is discussed in Section III.

Unlike ASIC TDCs in which differential non-linearity (DNL) can be well controlled and a lot of times the raw TDC bin number can be directly used without calibration, in FPGA TDCs, the DNL is large and a bin-by-bin calibration is indispensable. Several topics on automatic calibration are covered in Section IV.

Gray code counters and dual counters are common approaches used to implement coarse time counters in ASIC TDCs. In FPGA TDCs (and several ASIC TDC architectures), however, these complex approaches are not necessary. Issues regarding coarse time counters are discussed in Section V.

Single-ended signals can generate large amounts of noise to cause degradation of time measurement precision. Guidelines on using single ended input/output pins are discussed in Section VI.

II. Fine Delays in FPGA and Multiple Measurements Using the Wave Union Launcher

In this section, several considerations of delay chains are discussed. The designers normally cannot redesign the delay chains in an FPGA so choosing an appropriate FPGA family with suitable delay chain structure is a crucial step toward a

Manuscript received December 15, 2009, revised February 12, 2010. This work was supported in part by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy and University of Chicago's Fermilab Strategic Collaborative Initiative.

The author is with Fermi National Accelerator Laboratory, Batavia, IL 60510 USA (phone: 630-840-8911; fax: 630-840-2950; e-mail: jywu168@fnal.gov).

successful TDC design.

A. Using Dedicated Chain Structure

The routing between arbitrary logic elements may need to pass several interconnecting matrices and the propagation delays can be very long with large variations even after laborious hand layout of the logic elements.

Carry chain structures are available in most FPGA families designed for implementing adders, accumulators and counters for digital processing applications. The carry chains are dedicated routes between FPGA logic elements with minimal propagation delay so that counters with many bits can operate at high frequencies. Therefore, it is recommended to use a dedicated carry chain structure for the TDC instead of using generic interconnects between logic elements. In a low cost family Altera Cyclone II [11] FPGA device EP2C8T144C6, the typical propagation delay between two logic elements connected via carry chain is about 60 ps.

It should be pointed out that a carry chain that is too fast is not suitable for TDC implementation purposes. An ideal delay chain should have relatively uniform propagation delays in each delay cell so that the differences of the input signal arrival times can be recorded in the register array.

In some high end FPGA families, advanced carry generation schemes such as carry selection may be utilized to optimize the performance of high speed adders. Implementing TDCs using these families is significantly difficult, if not impossible, compared to implementing TDCs using low cost families with plain carry chains.

B. Delay Chain Length and Clock Frequency

The delay line length should be kept as short as possible to reduce both logic resource usage in the encoder and also the measurement errors especially at the middle of the delay chain. To reduce delay chain length, the clock frequency driving the register array should be chosen as high as can be reasonably achieved. Typically, at a relatively high frequency in an FPGA, the delay chain length is around 32 to 64 steps.

Different logic resources in FPGAs usually have different maximum operating frequencies. The high frequency chosen for the register array is likely to be too high for circuits in later stages, especially memory blocks. Schemes interfacing a fast register array with a slow back stage are normally necessary, and one of these schemes is discussed later.

C. An Implementation of the Wave Union Launcher

A wave union launcher generates multiple 0-1 transitions in the delay chain to improve time measurement precision. A wave union scheme described in [7] and [9], the "Wave Union Launcher A", is studied here as an example. The Wave Union Launcher A generates a pulse train with three logic transitions, two of which are encoded.

The primary motivation of using the wave union launcher is to subdivide the "ultra-wide bins" caused by uneven physical structure of the FPGA carry chain. In the EP2C8T144C6 device with typical bin width 60 ps, for example, the ultra-wide bins can be as wide as 160 ps. For principle of subdividing ultra-wide bins with the wave union, please refer [7].

The Wave Union Launcher A in each channel of the TDC is implemented with a Logic Array Block with 12 logic elements. (The first 4 bits in the array are used for diagnosis purposes.) It feeds into other 48 cells in a 64-cell carry chain/register array. The bit pattern of the array is shown in Fig. 2, which is directly output from the FPGA via a serial port.

Each line in the bit pattern is a snapshot of the register array driven by a 387.5 MHz clock signal, which is synthesized from a 100 MHz external clock by multiplying/dividing a factor of 31/8 using the phase-lock-loop (PLL) circuit inside the FPGA. The frequency is chosen for the convenience of generating calibration test pulses in other part of the FPGA.

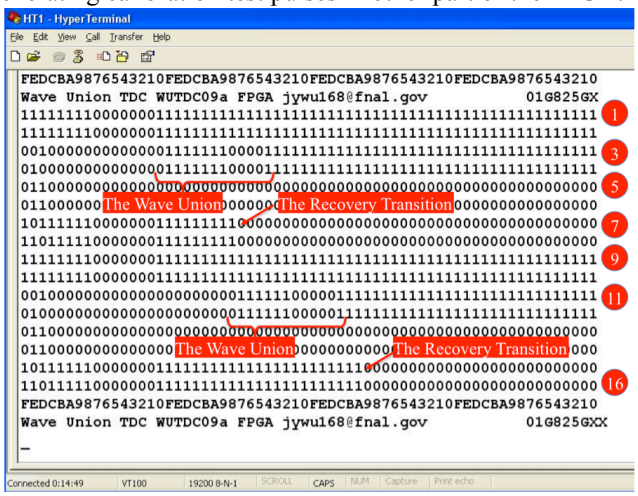

Fig. 2. Bit pattern of wave union launcher response to input pulses.

When the input is low, the 64-bit array stays in its initial state as shown in the first two lines. A bit pattern of 0's and 1's is formed in the wave union shown from the 5th column to the 16th column in each line. The remaining 48 bits down in the carry chain, i.e., the columns 17 to 64 are all held at 1.

Between the second and third clock cycles, the input becomes high and the wave union is launched in the carry chain. The wave union is captured by the clock edge as shown in line 3. The position of the wave union represents the arrival time of the input signal. The earlier the arrival time is, the further right the wave union is captured.

When the wave union is captured, the register array clock is disabled for one or two clock cycles. Therefore, the wave union pattern in line 4 remains unchanged. This feature guarantees that the wave union pattern will stay in the register array for at least two clock cycles, allowing the encoder in the later stage to operate with a clock of half frequency (193.75 MHz) to simplify the design and reduce power consumption.

When the input stays high, the entire array is all 0's as shown in lines 5 and 6.

When the input returns to low, the wave union launcher and the carry chain recover to their initial state. The recovery process is captured as shown in line 7. A 0-to-1 transition (marked as "The Recovery Transition", since the transitions propagate from left to right, the "0-to-1" transition is actually a bit pattern "10" in line 7) in the 48-bit carry chain represents

the time of the input falling edge. Multiple transitions exist only in the wave union launched by the input rising edge. Only a single measurement is available for the falling edge as it is in regular non-wave union TDC schemes. Therefore a coarser timing resolution for falling edges is anticipated.

The falling edges also cause the register array to be disabled for one or sometimes two clock cycles. When the register array is enabled again, the initial condition is shown in lines 9 and 10.

Another pulse is captured as shown in the remaining six lines. The two pulses shown in this example are 10 ns wide and separated by 10 ns.

III. The Encoder

A design detail that should be mentioned here is that the encoder must be "bubble proof". In the ideal cases, the 0-1 transitions recorded by the register array are clean thermometer codes, such as 000001111. However, "bubbles" at the transition edges such as 000010111 may occur due to uneven propagation delays in the FPGA structure. The encoder should be designed to output a reasonable value when the transition edge bubbles occur. The transition detecting logic shown in Fig. 3 is a simple scheme to eliminate the problems caused by the bubbles.

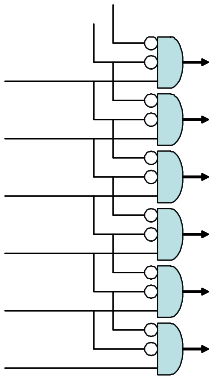

Fig. 3. Encoder transition detecting logic

The transition detecting logic above recognizes a 3-bit pattern "001" as a transition rather than a 01 pair. In the thermometer code with a bubble: "000010111", only the left most transition is recognized, while the 01 transition caused by the bubble is ignored. Of course if a bubble causes a larger hole like in "000100111", this particular 3-input logic will fail. In this case, one may add an additional input to the AND gates so that only 4-bit patterns "0001" are recognized as transitions. In reality however, bubbles with a hole larger than 1 bit have not been observed in our work.

IV. Automatic Calibration

A special issue of the FPGA-based TDC is its large differential nonlinearity (DNL), which is seen as large variation of apparent width of each TDC bin. Furthermore, the propagation speed of the delay cell is a function of temperature and power supply voltage. Therefore, it is necessary to calibrate the delay line as frequently as possible, preferably online. In this section, several topics regarding automatic calibration are discussed.

A. Average Delay Approach vs. Bin-by-Bin Approach

There are at least two approaches of digital calibration, i.e., the average delay approach and the bin-by-bin approach.

In the average delay scheme [4], the total delay time of the delay line is designed to be longer than the clock period t_p. Sometimes, an input logic transition will be recorded by the register array twice. If the positions of the two registered logic transitions are N_1 and N_2, respectively, then the average cell delay is:

$$t_d = \frac{t_p}{N_2 - N_1} \tag{1}$$

Sometimes, the number of delay taps propagated in a clock period can be viewed as a fractional value rather than an integer [10]. This value is calculated from multiple measurements, and will provide a more accurate calibration.

The advantage of this scheme is its fast response time. However, it does not provide bin-by-bin calibration when the bin widths are different since only the average cell delay is measured in this scheme.

For FPGA-based TDCs, bin-by-bin calibration is recommended since the widths of the bins vary by a large range.

B. Calibration to the Centers of the TDC Bins

Assuming that the widths of all TDC bins are measured and stored in an array w_k, then the calibrated time t_n corresponding to the center of bin n can be written as:

$$t_n = \frac{w_n}{2} + \sum_{k=0}^{n-1} w_k \tag{2}$$

It should be emphasized that it is crucial to calibrate to the centers of the bins, i.e., the first term representing the half width must not be omitted. It is not impossible for one to implement the sum term only and omit the half width term when the calibration algorithm is buried in complicate codes.

It can be shown that the RMS measurement errors are the minimum when the times are calibrated to the centers of the bins. Consider the RMS error σ contributed by one bin with lower and upper limits of t_1 and t_2 respectively. If this bin is calibrated to a value t_c between the lower and upper limits, the contribution of the error can be written:

$$\sigma^2 = \frac{1}{(t_2 - t_1)} \int_{t_1}^{t_2} (t - t_c)^2 dt = \frac{(t_2 - t_c)^3 - (t_1 - t_c)^3}{3(t_2 - t_1)} \tag{3}$$

When the bin is calibrated to the center, i.e., $t_c = (t_1 + t_2)/2$, the error above reaches a minimum which is $(t_2 - t_1)^2/12$.

The sum term in equation (2) represents the calibration to the edges of the bins. When all the bins have identical width, the half width term is a constant offset and calibrating to either bin edges or to bin centers will result in the same RMS errors. However, when the widths of the bins are different, the RMS errors will increase.

Integrating the look-up table (LUT) for the calibration to the centers of the bins takes a few extra steps than calibrating to

the edges. Once the widths of all TDC bins become available, usually as contents of the DNL histogram, a sequence controller starts to build the LUT in the FPGA's internal memory. The process is as following:

1. Half of the width of the first bin becomes the time at its center.
2. Another half bin width of the first bin and the half bin width of the second bin are added to get the center time of the second bin.
3. This sequence is repeated for remaining bins.

C. Calibration Hits: Random vs. Correlated

The widths of the TDC bins are measured through booking a DNL histogram. After booking a large amount of hits into the histogram, the count in each bin is proportional to its widths. For example, if 4095 hits are booked into the histogram and assume these hits are evenly spread over 2580 ps, the period of 387.5 MHz clock driving the TDC, then the width of an N-count bin is N*(2580 ps)/(4095) = N*0.63 ps.

There are two options for generating the calibration hits. The most common approach is to use random real event hits during the operation. Alternatively, hits generated by an oscillator different from the one driving the TDC clock may be used as the calibration hits. Care must be taken to ensure that the calibration hits have no correlation with the TDC clock. In high energy physics applications, for example, it is safer to use an on-board free running clock rather than a clock signal derived from the accelerator RF. A disadvantage of the random hit approach is that the statistical fluctuation may cause errors in the calibration LUT. The errors will reduce as $1/\sqrt{N}$ when the total number of hits N increases. Sometimes a very large amount of calibration hits is necessary in order to reach required precision.

Another approach is to use correlated hits, i.e., the calibration pulses derived from the same crystal oscillator that drives the TDC device. In a work described in Reference [9], two phase-lock-loop (PLL) circuits are used to generate TDC clocks and calibration pulses as shown in Fig. 4.

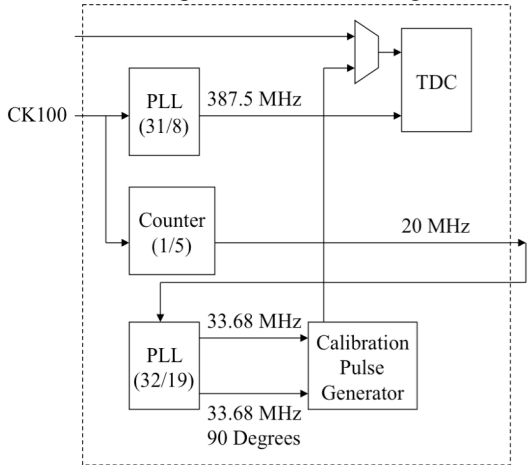

Fig. 4. Correlated calibration pulse generation

An external clock signal at 100 MHz is used to drive the first PLL to generate the TDC clock which multiplies/divides the input by a factor of (31/8), creating a clock at 387.5 MHz with a period 2580 ps. The second PLL is used to generate the clocks for calibration pulses. The 100 MHz clock is first divided by a factor of 5, becoming 20 MHz and output to an I/O pin. The 20 MHz signal is input back into another PLL with a multiplying/dividing factor of (32/19) and the output clock frequency becomes 33.68 MHz. There are 256 different relative timing relations between the 33.68 MHz and 387.5 MHz clock edges. These 256 possibilities are evenly spread over a 2580 ps range and the separation between the two possibilities is 10.08 ps. To show this, simply calculate mod(n*(29687.5 ps), (2580 ps)) with n = 0 to 255 and note that 29687.5 ps = 1/(33.68 MHz). Two 33.68 MHz clocks with a 90-degree relative phase are generated so that four different clock edges, 0, 90, 180 and 270 degrees are available to produce the calibration pulses. These four phases correspond to n = 0 to 255, n = 0.25 to 255.25, n = 0.5 to 255.5 and n = 0.75 to 255.75. All together, there are 1024 evenly spread relative timing conditions between the clock edges, and the nominal timing separation between two conditions is 2.52 ps.

The advantage of this approach is that the precision of the calibration LUT is predictable and can be achieved with a small number of hits. The drawback is that the precision of the LUT is limited (in the case above, 2.52 ps) and it will not simply improve by collecting more hits. However, it should be possible to use two different crystal oscillators to drive the TDC and the calibration pulses to achieve a finer precision beyond the limit given above, by taking advantage of the phase shift between the two oscillators.

D. Resource Saving Implementation in FPGA

Online calibration in FPGAs needs internal memory space to store both the DNL histogram and the calibration LUT. If the times of both the rising edge and the falling edge of the input signal are digitized, then two sets of the DNL histogram and LUT are required. For different channels, the timing properties are different and therefore, each channel needs to have its own calibration functional block. Additionally, each channel needs to store raw hits temporarily. If the memory resources are not utilized efficiently, multi-channel TDCs may not fit in low cost FPGA devices.

Collectively, the temporary raw hit storage, DNL histograms and calibration lookup tables can be implemented in one dual port RAM as shown in Fig. 5.

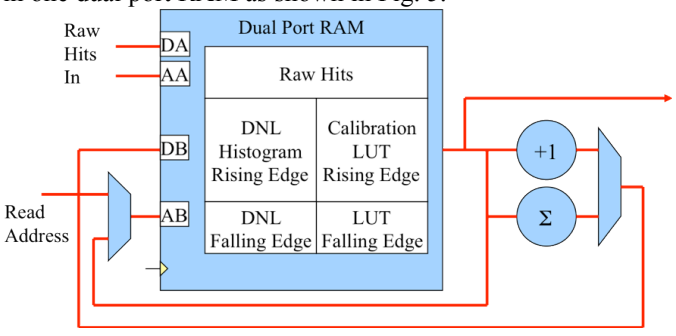

Fig. 5. The temporary hit buffer and automatic calibration functional block

TDC applications usually have high instantaneous hit rates but relatively low average rates. Therefore, a dedicated port

for writing the raw hit data into temporary storage memory is necessary. However, only a small memory space is needed. In Fig. 5, the port A is assigned for raw hit data writing. Since there are not many hits to be read out, port B can be shared for reading out raw hits, clearing memory area, booking histograms, integrating look-up tables and converting raw TDC bin numbers into fine times by checking the LUT.

Since the requirements of the two functions are complementary, i.e., the raw hit storage requires a high writing rate but a small storage space and the calibration operations need a large space but a low access rate, it is possible to combine the functions together by taking advantage of the full potential of the dual port RAM block.

V. Issues Regarding Coarse Time Counters

Long delay lines consume more logic cells not only in the delay line/register array structure, but also in the encoder and post processing stages. Long delay lines also cause larger measurement errors for bins in the middle of the chain even if the automatic calibration scheme described earlier is used. Therefore an optimal delay line length is slightly longer than a clock period. The TDC measurement range is extended with the coarse time counter beyond the length of the delay chain.

Dual counters driven by both edges of the system clock and the Gray code counters are popular choices for TDC coarse time counters, but they are only necessary for one type of TDC architecture found in ASIC-based TDCs. For FPGA-based TDCs, plain binary counters are sufficient. To explain why, we start by reviewing the TDC architectures.

A. The TDC Architectures

The delay line based TDC measures the time difference between the HIT signal and the timing reference clock CLK signal. The TDC can be classified by the signals being delayed and the signals used to clock the register array, as shown in Fig. 6.

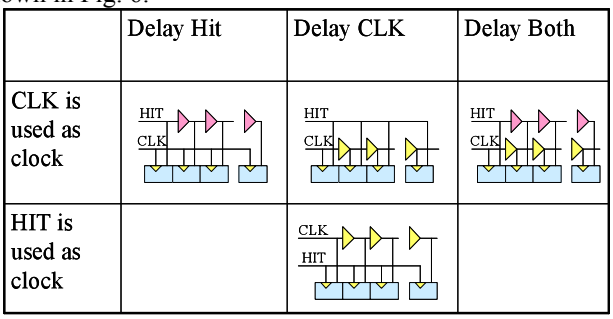

Fig. 6. TDC architectures

In principle, there could be six different TDC architectures but only four are commonly used.

The only TDC architecture that requires dual counters or Gray code counters is when the HIT signal is used as the clock for the register array. When the HIT signal arrives to record the coarse time, the coarse time counter driven by the CLK may be in an unstable condition and an incorrect time may be recorded. In this architecture, two counters driven by both edges of CLK or Gray counters are utilized. With dual counters, at least one of them is stable and is selected based on the most significant bit of the fine time. Using a Gray counter, at most one bit is flipping at each clock edge, so the error of the unstable edge is confined in a single bit and can be corrected later.

B. Coarse Time Counter in FPGA-based TDCs

FPGA-based TDCs use the architecture in which the HIT is delayed and CLK is used as the register array clock. In this case, the coarse time counter can be a plain binary counter and is implemented as shown in Fig. 7.

The input hit is recorded in the register array and the location of the wave union is encoded as a fine time bin number. Note that the uncertainty of the relative timing between the hit and the CLK is confined in the register array which is the value to be measured by the TDC. All other signals are derived from the output of the register array and their timing is well-defined by the CLK.

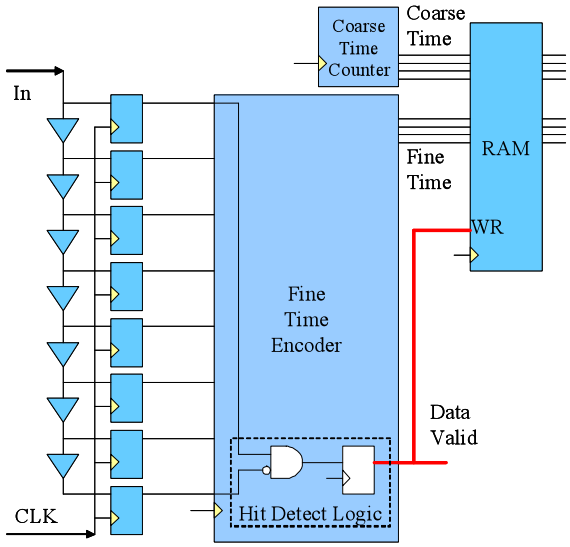

Fig. 7. The coarse time counter implementation in FPGA TDC

While the fine time is being encoded, a data valid signal is generated by the hit detect logic. The simplest hit detect logic senses the logic level difference between both ends of the register array so that a data valid signal is generated for the clock cycle when the wave union is inside the array. This data valid signal is used to enable writing both the fine time and the coarse time into the temporary storage RAM. Note that only at the zero-suppression stage, i.e., when writing the valid hits into the temporary storage, the coarse time is concatenated with the fine time. There may be other pipeline stages to process the fine time before saving it into the temporary storage and it is not necessary to introduce the coarse time in those pipeline stages. The setup and hold times are guaranteed since both the coarse time counter and the RAM block are driven by the same clock signal CLK.

VI. Noise Due To Single Ended I/O Signals

The output pins in FPGA devices are usually designed to support heavy driving capabilities. Once the pins turns on or off, voltage differences between the power supply and ground nets may change. The propagation delay of the CMOS gates

is very sensitive to the power supply voltage and automatic calibration can only correct for relatively slow temperature and voltage variations.

When a device receives input signals driven by other devices, large voltage spikes may also be present at the power and ground nets. Therefore, the time measurement precision can be degraded due to active I/O signals, especially single-ended ones. Based on our own experiences, even a flashing LED driven by the FPGA-based TDC may cause a visible increase of the time measurement error at 15 ps levels.

It is recommended to limit I/O activities to as few as possible during time measurement operations. Below are several design suggestions:

- All channel inputs of the TDC shall be in differential signaling standard such as LVDS to reduce interference between channels.
- If data must be output during the time measurement periods, avoid using single-ended signals. Use differential signaling instead.
- Avoid connecting the TDC to external memory devices or other devices with wide buses. Use differential outputs to send data to another FPGA to interface with the external memory and other devices.
- LED flashing shall be stopped after TDC device initialization.
- If the application of the TDC is "windowed", i.e., the arrival time is within a certain time period, single-ended signals may be used outside the timing window. However, all I/O pins shall be kept to constant states before and during the timing window. Note that pins shall not only stop flipping, but shall also be held at the same state each time since a pin staying high or low may drain different current from the power net that may cause the power supply voltage of the FPGA core to change.

VII. Conclusion

An implementation of the Wave Union TDC, a novel scheme of FPGA TDC to improve time measurement precision using multiple measurements is discussed. Design considerations in various aspects such as interfacing with encoder which operating in lower frequency clock domain, difference in leading and trailing edge resolution and double pulse separation can be seen from its final operation properties.

Several topics on delay line based TDCs are also discussed. When the TDC is implemented in an FPGA, in addition to the common problems shared with an ASIC TDC implementation, there are FPGA specific issues. Correctly solving these issues such as bit pattern bubble, large DNL etc. allows FPGA-based TDCs to be utilized in practical applications.

Generally, Gray code counters or dual counters are only necessary for a particular ASIC-based TDC architecture and unnecessary for FPGA-based TDCs.

Finally, the timing measurement precision of any TDC is highly sensitive to noise on the circuit board. Correct board design practices are essential.

Acknowledgement

The authors would wish to express thanks to Mike Albrow, Erik Ramberg, Anatoly Ronzhin, Robert DeMaat, Sten Hansen, Rajendran Raja, Holger Meyer of Fermilab, Fukun Tang, Henry Frisch, Jean-Francois Genat, Chien-Min Kao of University of Chicago, Qi An of University of Science and Technology of China, William Moses, Seng Choong, Chinh Vu, Qiyu Peng of Lawrence Berkeley Lab and Yun Wu of Apple Inc. for their helpful inputs over years.

References

[1] A. Amiri, A. Khouas & M. Boukadoum, "On the Timing Uncertainty in Delay-Line-based Time Measurement Applications Targeting FPGAs," in *Circuits and Systems, 2007, IEEE International Symposium on,* 7-10 27-30 May 2007 Page(s): 3772 - 3775.

[2] J. Song, Q. An & S. Liu, "A high-resolution time-to-digital converter implemented in field-programmable-gate-arrays," in 2005 *IEEE Transactions on Nuclear Science,* Pages 236 - 241, vol. 53.

[3] M. Lin, G. Tsai, C. Liu, S. Chu, "FPGA-Based High Area Efficient Time-To-Digital IP Design," in *TENCON 2006. 2006 IEEE Region 10 Conference,* Nov. 2006 Page(s):1 – 4.

[4] J. Wu, Z. Shi & I. Y. Wang, "Firmware-only implementation of time-to-digital converter (TDC) in field programmable gate array (FPGA)," in 2003 *IEEE Nuclear Science Symposium Conference Record,* 19-25 Oct. 2003 Page(s):177 - 181 Vol. 1.

[5] S. S. Junnarkar, et. al., "An FPGA-based, 12-channel TDC and digital signal processing module for the RatCAP scanner," in *2005 IEEE Nuclear Science Symposium Conference Record*, Volume 2, 23-29 Oct. 2005 Page(s):919 - 923.

[6] M. D. Fries & J. J. Williams, "High-precision TDC in an FPGA using a 192 MHz quadrature clock," in *2002 IEEE Nuclear Science Symposium Conference Record*, 10-16 Nov. 2002 Page(s):580 - 584 vol. 1.

[7] J. Wu & Z. Shi, "The 10-ps wave union TDC: Improving FPGA TDC resolution beyond its cell delay", in *2008 IEEE Nuclear Science Symposium Conference Record*, 19-25 Oct. 2008 Page(s):3440 - 3446.

[8] J. Wu, "On-Chip processing for the wave union TDC implemented in FPGA", in *2009 IEEE-NPSS Real Time Conference Record,* 10-15 May. 2009 Page(s):279 - 282.

[9] J. Wu, "An FPGA Wave Union TDC for Time-of-Flight Applications", in *2009 IEEE Nuclear Science Symposium Conference Record,* 25-31 Oct. 2009 Page(s): 299-304

[10] J. Wang, et. al., "A fully fledged TDC implemented in field-programmable-gate-arrays", in *2009 IEEE-NPSS Real Time Conference Record,* 10-15 May. 2009 Page(s):290 - 294.

[11] Altera Corporation, "*Cyclone II Device Handbook*", (2007) available via: {http://www.altera.com/}