FERMILAB-COPNF-09-275-E

On-Chip Processing for the Wave Union TDC Implemented in FPGA

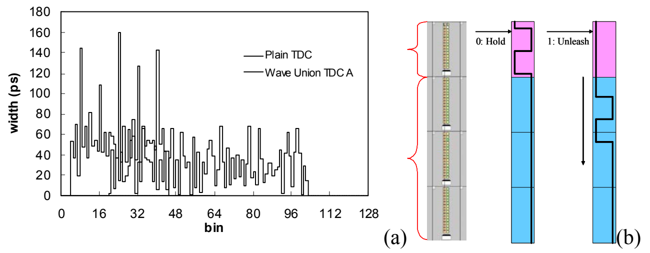
Jinyuan Wu

***Abstract*— The wave union TDC implemented in FPGA utilizes multiple measurement method to reach time resolution beyond the natural carry cell delay in FPGA. Lacking of analog compensation for bin width control available in ASIC, the wave union TDC takes the after-fact digital calibration approach. In addition to the temperature drift, non-uniformity of the carry chain structure in FPGA causes complicate differential non-linearity pattern which imposes significant on-chip calibration challenge. In this paper, processing strategies for the wave union TDC are discussed. Actual implementations in low-cost FPGA with 20ps and 10ps RMS resolutions are also presented..**

***Index Terms*— Front End Electronics, TDC, FPGA Firmware**

I. Introduction

CARRY chain structure in existing in FPGA families can be used in time-to-digital conversion (TDC) purposes[1-7]. A special feature of the FPGA TDC is its large differential nonlinearity (DNL) as shown in Fig. 1(a) which is represented as apparent width of each TDC bin.



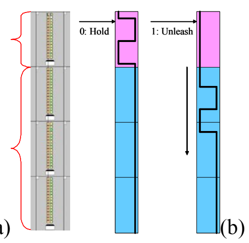


Fig. 4. The bin width plot (A) and a wave union launcher (b)

The most significant origins of DNL is the logic array block (LAB) structure. When the input signal in the carry chain passes across the LAB boundaries (and also the half-LAB boundaries in some FPGA families), extra delays added cause periodic “ultra-wide bins”.

In our previous work [7], an approach called the “wave union TDC” is developed to sub-divide the ultra-wide bins and to improve measurement resolution. The key part in the wave union TDC is the “wave union launcher” as shown in Fig. 1(b). A wave union launcher creates a pulse train or “wave union” with several 0-to-1 or 1-to-0 logic transitions for each input hit and feed the wave union into the TDC delay chain/register structure, making multiple measurements.

There are two types of the wave union launchers: (1) the Finite Step Response (FSR) ones and (2) the Infinite Step Response (ISR) ones. This classification is an analogue of Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) for linear systems except the inputs for the wave union launchers are logic steps. A FSR wave union launcher, like FIR linear systems, employs no feedback and generates a pulse train with finite length and limited number of logic transitions. An ISR wave union launcher, like IIR linear systems, uses feedback to generate an infinite pulse train.

In our work, we have studied two wave union launchers: the “wave union launcher A”, a FSR launcher with two useable logic transitions and the wave union launcher B”, an ISR one based on a ring oscillator.

Based on our measurement, in an Altera Cyclone II device (EP2C8T144C6) [8], the typical raw bin width is about 60ps, while the ultra-wide bins can be as large as 165ps. With the wave union launcher A, the maximum bin width can be reduced to 65ps and an RMS resolution of 25ps can be achieved. With the wave union launcher B and grouping of multiple TDC channels, a delta T RMS resolution of 10ps was reached.

This document serves as supplemental material for Reference [7] and covers several crucial design precautions and implementation details. The remaining sections first describe actual implementation of the wave union launchers followed with explanations on the automatic calibration functions and discussions on the issues of coarse time counter implementation.

II. The Wave Union Launchers

A wave union scheme, “wave union launcher A”, has been tested as a proof of concept. The wave union launcher A belongs to the FSR type. It generates a pulse train with three logic transitions of which two are encoded.

Another version of the wave union TDC, the “wave union TDC B” has also been tested. The “wave union launcher B” used in this test is simply a ring oscillator enabled by the input and it belongs to ISR type. After the input level turns from 0 to 1, a pulse train with unlimited length and logic transitions is generated. More measurements can be made for one input so that the average of these measurements yields better TDC resolution.

We will discuss the implementations of the wave union

Manuscript received May 30, 2009. 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).

launchers in this section.

A. The Wave Union Launcher A

The wave union launcher A is implemented in a LAB with 16 logic elements as shown in Fig. 2. It is connected with rest of 48 cells (16 are shown) in the 64-cell carry chain/register array.

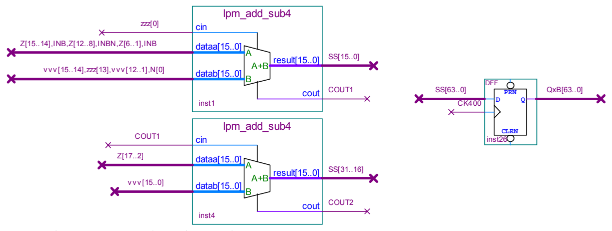


Fig. 2. The wave union launcher A

The wave union launcher and the delay chain are implemented with adders. Adders provide a natural structure suitable for TDC and allow the compiler to automatically place the bits in the delay line/register array along the carry chain provided in the FPGA family. The sum bits are connected to the D-flip-flops that are in the same logic element as the combinational lookup tables. Most inputs of the adder are assigned as either logic 0 or logic 1. Two types of logic 0 or 1 assignments are created, constants or variables. The signals zzz[] and vvv[] are constant logic 0 and logic 1 and the signals Z[] and N[] are "variable" logic 0 and 1, respectively. The variable logic 0 and logic 1 signals are output bits of a shift register and their logic values are constant 0 or 1 after power-up initialization. The reason of having the "variable" logic 0 and 1 is to prevent the compiler from simplifying the adder structure. If the inputs of an adder were all assigned with constant 0 and 1, the compiler would eliminate the adder and create simple logics that would be optimal in speed and resource in other applications.

In the wave union launcher A, the bits for one input are assigned with 0 except bit 0 and 13 which are assigned with the TDC input INB and bit 7 which is assigned with INBN, the inverted version of INB. Bits of another input are assigned with 1 except bit 13. When input INB=0, the output of the adder SS[0..15] = 1111111000000111 with no carry to the next adder. When the input leading edge arrives, i.e., INB = 1 and INBN = 0, three logic transitions start to propagate from along the carry chain: two 1-to-0 transitions starting from bit 0 and bit 13 and one 0-to-1 transition starting from bit 7. The wave union with these three logic transitions are launched into the carry chain and recorded in the register array for further encoding.

In our design, we encode the two 1-to-0 transitions primarily for simplicity and resource saving in the encoder and post-processing circuits. The sum of the bin number output from the encoder is fed into the memory buffers and the automatic calibration block in later stages.

B. The Wave Union Launcher B

The wave union launcher B is implemented in a LAB with 16 logic elements as shown in Fig. 3. It is connected with rest of 48 cells (16 are shown) in the 64-cell carry chain/register array.

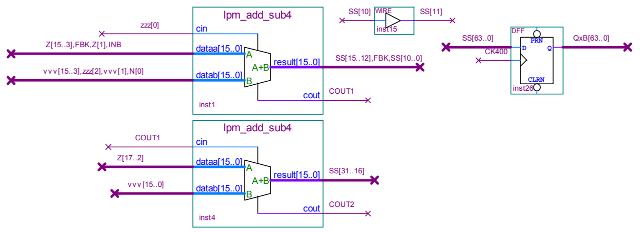


Fig. 3. The wave union launcher B

The wave union launcher B is essentially a ring oscillator enabled by the input. The close loop delay of the ring oscillator is controlled by assigning the feedback tap in the delay line. In Fig. 3, the feedback tap is marked as signal FBK at output bit 11 and is input at bit 2 of the adder. When input INB = 0, the SS bits are all 1 and the oscillation is not enabled. Once input arrives, i.e., INB = 1, a 1-to-0 transition propagates in the carry chain. After certain delay, FBK at output bit 11 becomes 0 which causes a 0-to-1 transition to start from bit 2. The oscillation repeats until INB returns to 0 and a wave union with many transitions is launched into the carry chain.

III. AUTOMATIC CALIBRATION BLOCK

The propagation delay of a delay cell depends on temperature and power supply voltage. In ASIC TDC it is possible to compensate the delay variation using analog method, i.e., to generate a control voltage from the phase difference of external crystal oscillator and the internal ring oscillator and to use the control voltage to fine tune the internal cell delays via a negative feedback.

In FPGA TDC, analog compensation is not convenient and digital calibration is more preferable. The automatic calibration functional block we developed in our work is shown in Fig. 4.

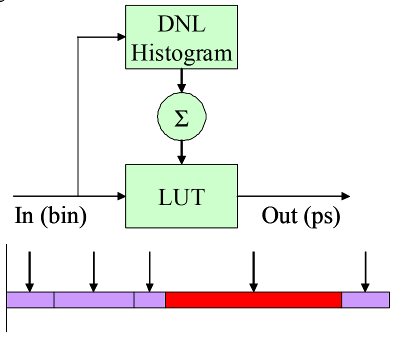


Fig. 4. The automatic calibration functional block

After power up or system reset, all TDC inputs are fed with calibration hits to book the DNL histogram and then generate the calibration lookup table LUT. The timing of these hits should have no correlation with the clock signal driving the TDC, so the hits should be generated from an independent oscillator. It is also possible to use real event hits as calibration hits if the hit rate of the real events is sufficiently high. The calibration block updates the DNL histogram and the calibration LUT automatically and semi-continuously as the real events flow through.

The DNL histogram and the calibration LUT are actually implemented in a single dual-port memory block as shown in Fig. 5.

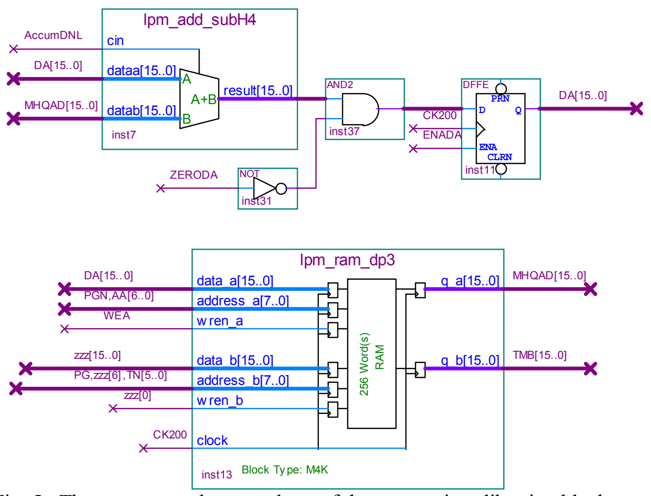


Fig. 5. The memory and accumulator of the automatic calibration block

The 256-word x 16-bit memory is slit into two pages controlled by a page selection signal PG and its inverse PGN. The port B of the memory is used to convert the bin number TN from the encoder to the calibrated time TMB through the valid LUT in the current memory page. The port A is used to book the histogram and integrate the LUT. The adder and register forms an accumulator that can be also be held to zero or add input from MHQAD by 1 with appropriate setting of control signals. The process is controlled by a finite state machine with the following four states:

1. Clearing memory area. The memory page addressed by PGN is looped through and over-written with DA held = 0.
2. Booking DNL histogram.
3. Integrating the calibration LUT.
4. Swapping memory page.

The input from the TDC encoder in our design is a 6-bit number, representing the bin number of the logic transition of the input signal with possible range of 0 to 63. (For wave union launcher A, the sum of two 6-bit numbers is a 7-bit number.) A 64-bin (or 128-bin) DNL histogram is booked in the FPGA internal memory. If the number of total hits is known, then the counts in each bin can be used as its bin width. For example, if 16384 hits are booked into the histogram and assume these hits are evenly spread over 2500ps, the period of 400MHz clock driving the TDC, then the width of a bin with N count is N*2500ps/16384 = N*0.1526ps.

Once all hits are booked into the histogram, a sequence controller starts to build the lookup table (LUT) in the FPGA internal memory. The LUT is integrated from the DNL histogram so that it outputs the actual time of the center of the addressed bin. The time value of the first bin is half of the width of the first bin. Then 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. This sequence is repeated for remaining bins. It is crucial to calculate the calibration values for the centers of the bins. If all the bins had the same width, there would be no difference to calibrate either to the center values or the boundary values of the bins. But when the bin widths are different, calibrating to the center values reduces measurement error significantly.

Once the LUT is built, the memory pages are swapped in a single clock cycle so that no service dead time occurs during LUT update.

IV. Issues Regarding Coarse Time Counters

An optimal delay line length is slightly longer than a clock period. 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 cause larger measurement errors for bins in the middle of the chain even with automatic calibration scheme described earlier. The TDC measurement range is extended with the coarse time counter beyond the length of the delay chain.

Double counters driven by both edges of the system clock and the Gray code counters are popular choice for TDC coarse time counters, but we should point out that they are only necessary for one type of TDC architecture found in ASCI TDC. For FPGA TDC, plain binary counter is sufficient. To explain this, we start by reviewing the TDC architectures.

A. The TDC Architectures

The delay line based TDC measures 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.

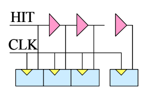
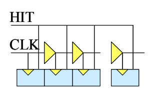
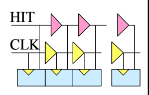
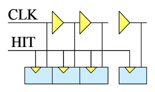
	Delay Hit	Delay CLK	Delay Both
CLK is used as clock			
HIT is used as clock			

Fig. 6. TDC architectures

In principle, there could be six different TDC architectures but there are only four are seen in literatures.

The only architecture of TDC that requires double counters or Gray code counters is when the HIT signal is used as the clock for the register array. When the HIT signal arrives at the register array 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 double counters, at least one of them is stable and is selected based on the most significant bit of the fine time. Using gray counter, at most one bit is flipping at each clock edge, so that the error of unstable edge is confined in a single bit and the error can be corrected later.

B. *Coarse Time Counter in FPGA TDC*

FPGA TDC uses the architecture in which the HIT is delayed and CLK is used as the register array clock. In FPGA TDC, the coarse time counter is a plain binary counter and is implemented as shown in Fig. 7.

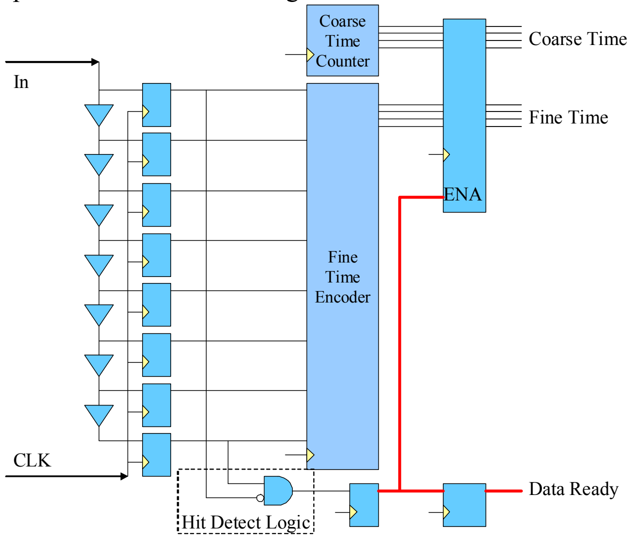


Fig. 7. The coarse time counter implementation in FPGA TDC

The input hit is recorded in the register array and the location of the wave union is encoded as fine time. 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 and they are staged into the process pipeline.

While the fine time is being encoded, a hit valid signal is being 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 hit valid signal is generated for the clock cycle when wave union is inside the array. This hit valid signal, eventually being derived as the data ready signal, is used to enable latching of the coarse time. The setup and hold time are guaranteed since both the coarse time counter and the register are drive by the same clock signal CLK.

V. Conclusion

Several technical details for the wave union TDCs are discussed in this document.

The wave union launcher and the delay chain are implemented with adders that simplify the design and compiling processes. A multi-channel FPGA can be designed with small amount of manual placement and majority of the placement and routing are done by the compiler automatically and the compiler yields desired results.

The practical implementation of the automatic calibration block uses minimum logic element and memory resource, which permits on-chip processing for multi-channel TDC.

The details of coarse time counter implementation ensured simplicity in this aspect.

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 and Qi An of University of Science and Technology of China 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 *IEEE Transactions on Nuclear Science,* 2005, 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 *Nuclear Science Symposium Conference Record, 2003 IEEE*, 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 *Nuclear Science Symposium Conference Record, 2005 IEEE*, 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 *Nuclear Science Symposium Conference Record, 2002 IEEE*, 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 *Nuclear Science Symposium Conference Record, 2008 IEEE*, 19-25 Oct. 2008 Page(s):3440 - 3446.

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