FERMILAB-Conf-03/358-E November 2003

Firmware-only Implementation of Time-to-Digital Converter (TDC) in Field-Programmable Gate Array (FPGA)

Jinyuan Wu, Zonghan Shi and Irena Y. Wang for CKM Collaboration

Abstract— A Time-to-Digital Converter (TDC) implemented in general purpose field-programmable gate array (FPGA) for the Fermilab CKM experiment will be presented. The TDC uses a delay chain and register array structure to produce lower bits in addition to higher bits from a clock counter. Lacking the direct controls custom chips, the FPGA implementation of the delay chain and register array structure had to address two major problems: (1) The logic elements used for the delay chain and register array structure must be placed and routed by the FPGA compiler in a predictable manner, to assure uniformity of the TDC binning and short-term stability. (2) The delay variation due to temperature and power supply voltage must be compensated for to assure long-term stability. We used the chain structures in the existing FPGAs that the venders designed for general purpose such as carry algorithm or logic expansion to solve the first problem. To compensate for delay variations, we studied several digital compensation strategies that can be implemented in the same FPGA device. Some bench-top test results will also be presented in this document.

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

I. Introduction

THE purpose of Fermilab's CKM experiment is to measure the branching ratio of $K^+ \rightarrow \pi^+ \nu\nu$ events. The experiment uses a continuous kaon beam and detects decay products. Time-to-digital converters (TDCs) are used on all channels to measure the arrival time of hits coming from charged decay products. The binning size required is 1 ns. The experiment also requires that the TDC device have no intrinsic dead time.

Tapped delay lines have been used in various TDC designs. In custom-designed, application-specific-integrated-circuit (ASIC) TDC chips [1-5], the delay speed is dynamically adjusted by setting a reference voltage that is derived from a delay-lock-loop (DLL) or phase-lock-loop (PLL) circuit. The delay speed variation due to temperature is thus automatically compensated with an analog method.

Low cost, fast development cycle and commercial availability are several driving motivations for using general-purpose field-programmable gate arrays (FPGAs) to implement the TDC [6-8] without using any external circuits. The TDC can be simply included in the FPGA that handles data for the data acquisition (DAQ) system. We chose the architecture shown in Fig. 1. In the FPGA, a counter keeps clock count as the higher bits of the input time, which provides a coarse measurement accurate to a clock cycle. The lower bits that represent measurements finer than the clock cycle are generated in a delay chain and register array structure. Note that, in theory, depending on the arrangement of the input signal IN and clock signal CLK and the connection between the delay chain and the register array, there can be 4 different TDC architectures:

	CLK → delay IN → common	IN→ delay CLK → common
Common D: Delay chain taps drive clock inputs.	Type A: ASIC TDC chips	Type B:
Common clock: Delay chain taps link to D inputs.	Type C:	Type D: FPGA-only implementation

Most of ASIC TDC chips use Type A architecture—they delay the CLK while sending the IN signal to a common D input. In the FPGA, however, we have found that Type D architecture, which delays IN signal and uses CLK to drive common clock ports, has more advantages.

To implement the delay chain and the register array in FPGA, one must address two major problems:

1. In the FPGA development software, a logic cell (LE) can be physically placed in nearly any place, depending on the optimization algorithm used. When left up to the program, routing between LEs may also

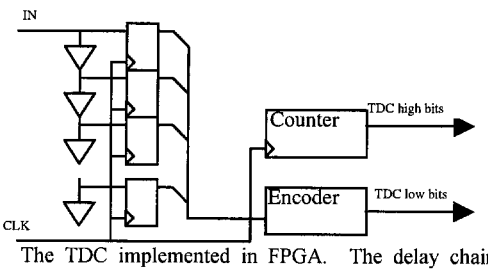

Fig. 1. The TDC implemented in FPGA. The delay chain and register array structure provides fine time measurement.

J. Wu and Z. Shi are with Fermi National Accelerator Laboratory, Batavia, IL 60510, USA, (phone: 630-840-8911; fax: 630-840-2950; e-mail: jywu168@ fnal.gov).

I. Wang is with Nappervill North High School, Nappervill, IL, 60563.

be unpredictable to the user. If the logic cells used for the delay chain and register array are placed and routed in this fashion, the propagation delay of each delay step will not be uniform. To avoid this, one could certainly place and route these logic cells manually, but that would be very time consuming. A methodology to automatically control the physical location and routing of LEs is thus demanded.

2. The logic cell delays vary with temperature and power supply voltage. In many applications such as serial communication and custom-designed TDCs, a reference voltage is adjusted to make the compensation. In FPGAs, however, a digital compensation method that requires no extra external circuits is more attractive than analog compensation.

In this document, we will discuss these two problems in detail and we will also present results from bench-top tests.

II. Using Existing Chain Structures

A. FPGA Chain Structures

Unpredictable non-uniformity of delay taps in the delay chain is a primary problem in the FPGA-only TDC implementation. The reason is that delay elements are often placed and routed unpredictably by the design software. A simple solution is to use existing chain structure in the FPGA.

In the FPGA available today, there exist chain structures that the vendors designed for general-purpose applications. A few well-known examples are carry chains, cascade chains, sum-of-products chains, BY-YB chains, etc. [9-10]. These chain structures provide short predefined routes between identical logic elements. They are ideal for TDC delay chain implementation.

It should also be pointed out that the uniformity of the register array is as important as the one of the delay chain. The clock inputs of flip-flops in the register array must have minimal skew, and the connection between the delay chain taps and the D inputs of the flip-flops must be as short as possible.

B. Altera Cascade Chain Structure

For our TDC bench top tests, we used a "cascade chain" structure originally designed for the fast logic expansion in the Altera ACEX 1K devices [9] (EP1K10QC208-1 and -2).

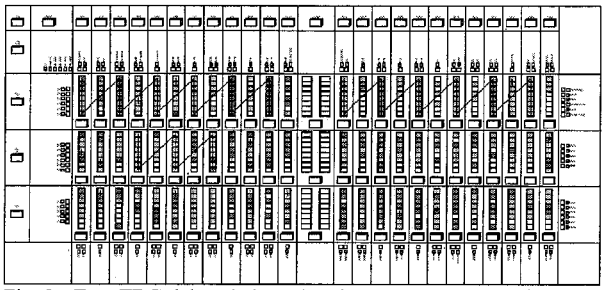

Fig. 2. Two TDC delay chain and register array structures (top row) are compiled in this Altera EP1K10QC208 device.

The cascade chain in an Altera device links adjacent Logic Elements (LEs) together. The output of the cascade chain tap is then sent to the D-input of a flip-flop in an LE.

The device data sheet shows a wide range for the propagation delay of a cascade chain tap. However, based on our measurements, the tap delay is typically around 0.4 ns.

There are 8 LE's in a Logic Array Block (LAB). At the end of a LAB, the cascade output from its last LE skips one LAB column and connects to the cascade input of the first LE in the next LAB to form a longer cascade chain. Each cascade chain can have a maximum of 48 taps. The FPGA development software MAX+plus II can be set to preserve the "cascade" primitives during logic simplification. The software compiles the chain structure to the correct configuration automatically (since there is only one possible set of LE assignments once the first LE of the chain is chosen.) A compiled example is shown in Fig. 2.

III. Digital Compensation Strategies

A. Digital Compensation for Temperature Stability

The delay time of the logic element varies with temperature and power supply voltage. However, the input clock provides a known and stable period that can be used as an on-chip timing standard. To make use of this timing standard, we can implement a delay chain with total propagation time longer than the clock period. With this longer delay chain, some input signals being registered by a clock edge can be registered again by the next clock edge as shown in Fig. 3. For instance, an input leading edge registered at N_1 by the first clock edge may be registered again at N_2 after a clock period T. Then, the propagation delay per tap, or the LSB bin size, would be $T/(N_2 - N_1)$. Using this information, one can compensate for variations in the delay chain speed and adjust the TDC output accordingly.

B. Bin-by-Bin Digital Calibration for Uniformity

In today's FPGA, digital operations on data do not take as large an amount of hardware resource as people once expected. For example, to correct the non-linearity due to non-uniformity of the delay structure, we can use look-up

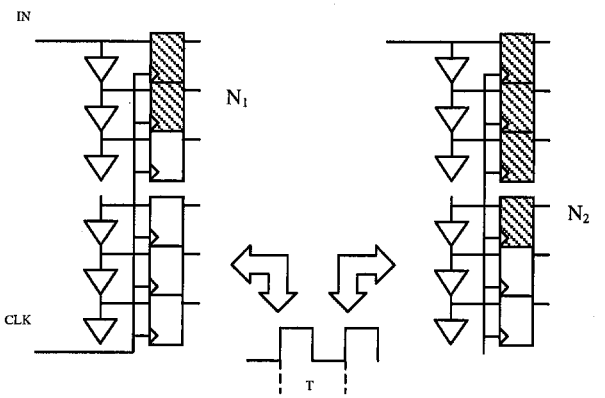

Fig. 3. An input leading edge is registered by the first clock edge at N_1 and then by the second clock edge at N_2.

tables available in each LE. Each LE has a 4-input 1-output look-up table. Assume we have a 4-bit raw TDC result from a 16-tap delay structure. Due to non-uniformity of the taps, we would need to map the 4-bit raw number into another number representing the corrected time. If this value needed N bits (N is normally larger than 4), the entire function would require only N Logic Elements.

IV. Bench Top Test Results

As shown in Fig. 2, we compiled two TDC chain structures in an Altera ACEX 1K device (EP1K10QC208-1 or -2) for our bench top tests. Each TDC structure had 48 steps. The external clock frequency we used in the tests was 35 MHz. In some tests, we boosted the frequency on the chip to 70 MHz. The leading edge caught by the register array was encoded and sent out at every clock cycle.

A TDC chain structure can be placed in any row (A, B or C from the top down). The first LE can be placed to columns 1, 2, 13 and 14. Therefore, there are 12 possible locations for a TDC chain structure in this device, which we denoted as A1, A2, A13, A14, B1, B2, B13, B14, C1, C2, C13 and C14. We implemented and tested all of them.

A. Short-Term Stability and Similarity of Device Structure

Fig. 4 shows a test result of the TDC output as a function of the signal input time. The input signal was derived from the system clock and its arrival time was changed by using delay cables of various lengths. More than 1000 measurements were made for each point and the average of each set of measurements was plotted. This was repeated for all 12 possible TDC chain locations. A simple linear fit showed that the least significant bit (LSB) bin size of the TDC structure in this device was about 0.4 ns. One would also expect some non-uniformity due to internal layout structure of the device. However, even without using any knowledge about the device structure, a simple linear fit matched the measurement points to better than 1 LSB. The measurement points for 12 different locations are essentially on top of each other, which suggests that the performance of the TDC structure is not very sensitive to the placement of the entire structure. Of course, the 12 different locations are not exactly identical, as will be discussed later.

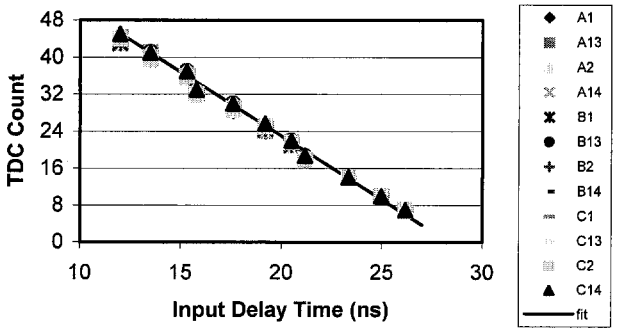

Fig. 4. TDC output as a function of the input time delay.

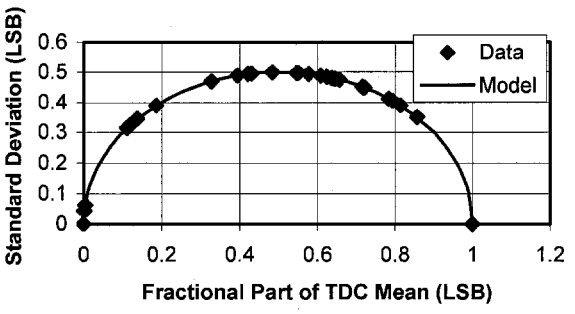

Fig. 5. Standard deviation plotted as a function of fractional portion of the TDC mean.

To help us understand short-term stability, the means and standard deviations of the measurements were calculated. In most cases, the TDC simply output a single number for all 1000 measurements. In these cases, the mean was an integer and the standard deviation was 0. There were cases when the input signal arrival time was around the boundary between two consecutive bins N and (N+1), so the output could be either. In these instances, the mean had a fractional part and the standard deviation was non-zero. The measured data were plotted in Fig. 5.

Assume that the TDC can output two numbers, N and (N+1) only, not any other numbers for a given input signal delay. The probability of outputing (N+1) is p. Then the fractional portion of the mean will be p and the standard deviation will be $\sqrt{p(1-p)}$. We have plotted the curve representing this simple binning module in Fig. 5.

From the agreement between the model and data, one can see that the measured standard deviation can be explained by this binning effect, and that the effect of other sources is negligible. Therefore in this test device, the intrinsic short-term instability, if there was any, was significantly smaller than 0.5 LSB.

B. Digital Compensation for Long-Term Stability

To study the digital compensation method, we drove the TDC structure at 70 MHz. At 0.4 ns/LSB, an input leading edge travels about 36 taps in a clock period T = 14.28 ns. The delay line length in our test device was 48 taps. Therefore, some input leading edges could be registered twice. We recorded these two values for a given input signal as "N_1" and "N_2" in Fig. 6.

To see what would happen under the worst possible conditions, we changed power supply voltage from 2.5 V to 1.8 V, which is about equivalent to a temperature change from 100 to 0 °C. Both TDC numbers changed significantly.

From the two TDC numbers, we can calculate the delay chain speed. The speed at a given temperature/voltage can be averaged over multiple measurements to reduce the digitization errors. Using the delay chain speed information, the corrected signal input time T_c can be calculated:

$$T_c = T\frac{(N_1+N_0)}{\langle N_2-N_1\rangle} = \frac{T}{L}(N_1+N_0)$$

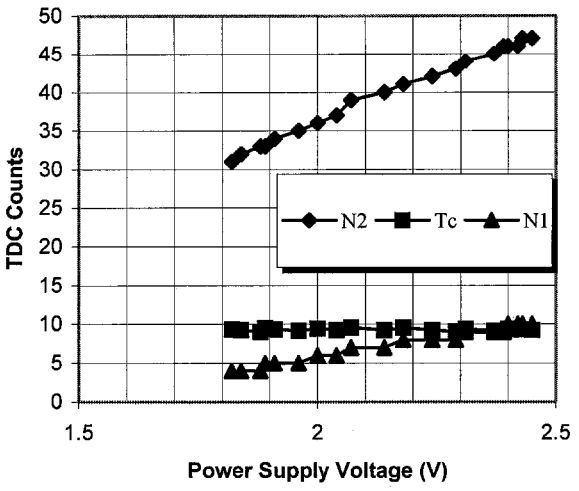

Fig. 6. TDC outputs vary with power supply voltage. The signal time compensated with digital algorithm is shown with square marker.

The value $L = \langle N_2 - N_1 \rangle$ above is the average of quantity ($N_2 - N_1$) over multiple measurements. The constant N_0 reflects propagation delays caused by the signal path before entering the delay chain plus the required setup time of the register array. These additional delays have same temperature and voltage dependence as the delay chain.

The corrected results are shown in Fig. 6. They vary by less than 1 LSB over the entire voltage range, compared to more than 6 LSB for the uncorrected N_1 value.

C. *Issues of Bin Size Variation*

To understand non-uniformity of the delay chain and register structure, we used random input signals and recorded the number of occurrences of each TDC output value in more than 50,000 measurements. We repeated these measurements for each of the 12 possible TDC positions.

Since arrival time was random, the number of events for a TDC value should have been proportional to the probability of the arrival time falling into that TDC bin. We calculated the relative bin widths based on this method. The results are plotted in Fig. 7.

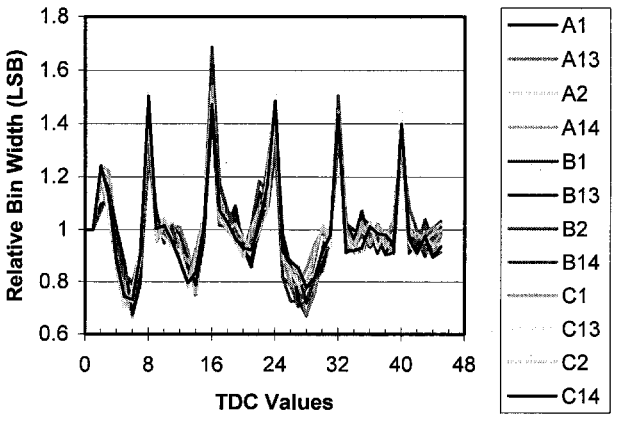

Fig. 7. Relative bin width for different TDC values.

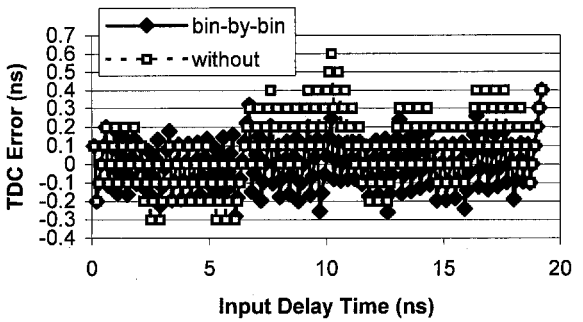

Fig. 8. TDC errors with or without bin-by-bin calibration.

As with the short-term stability tests, similar patterns appeared for all 12 locations. The global variation across the device was small.

The largest bin size variations were at values 8, 16, 24, 32 and 40. This irregularity was due to the cascade chain output at the end of one Logic Array Block (LAB) having to travel through a longer path before reaching the following LAB. *Within* each LAB, the variation of the relative bin width was about 20% of the typical bin width of 0.4 ns (80 ps).

The worst-case TDC errors (at location B2) for the 48-step delay chain in our device with and without bin-by-bin calibration are plotted in Fig. 8. With bin-by-bin calibration, the arrival time is corrected at the center of each digitization bin. The error is essentially due to the finite bin width. The maximum error in this case is 0.3 ns. Without bin-by-bin calibration, we simply assume all the bins have the same width. The maximum error is 0.6 ns.

Variation of bin width increases rms error by compounding on the normal digitization error of $1/\sqrt{12}$ or 0.29 LSB (0.12 ns if 1 LSB = 0.4 ns). In our test device, the worst-case rms error without bin-by-bin calibration was about 0.5 LSB (0.2 ns).

With bin-by-bin calibration, the rms error in each bin became $1/\sqrt{12}$ of the bin size. Since some bins were larger than the average LSB, the weighted average of the rms errors was slightly larger than 0.29 LSB. In our case, it was about 0.32 LSB (or 0.13 ns).

V. Discussions

A. *Hardware Implementation of Calibration and Compensation*

As we pointed out before, the bin-by-bin calibration can be implemented with look-up tables in the hardware. The function should use only a small amount of silicon resource.

The real challenge is the hardware implementation of compensation for temperature and power supply voltage variation. There may be two divisions needed in the correction:

$$T_c = \frac{T}{L}(N_1 + N_0)$$

One division is (T/L). The other is in the average $L = \langle N_2 - N_1 \rangle$ itself.

The average $L = \langle N_2 - N_1 \rangle$ can be implemented with an accumulator in FPGA. To avoid the division, we can simply let the accumulator to add 16 (or any 2^n) times. The bit pattern of the resulting sum automatically becomes the average, with the lower 4 bits being the fractional portion and the higher bits being the integer portion.

In the division (T/L), there is only one variable L, and only a few lower bits of L will change when the temperature variation is not too large. Thus, the division can be viewed as a single variable function of L and can be implemented with a small look-up table.

B. *Bin-by-Bin Calibration*

Earlier, we described that it is possible to do digital calibration and compensation on the TDC values. These corrections can be done either with software in the DAQ system or directly with hardware (firmware) in the FPGA itself.

However, in most applications, sub-LSB accuracy is not needed since an LSB is sufficiently small. When this is the case, bin-by-bin calibration is unnecessary. We can compare the measurement errors of several TDC devices in the following table.

	Measurement errors (ns)	
	(max)	(rms)
Perfect 1.2-ns-LSB TDC	0.6	0.34
Perfect 0.7-ns-LSB TDC	0.35	0.2
Test FPGA 0.4-ns-LSB TDC W/o bin-by-bin calibration	0.6	0.2
Perfect 0.6 ns-LSB-TDC	0.3	0.17
Perfect 0.45 ns-LSB-TDC	0.225	0.13
Test FPGA 0.4-ns-LSB TDC With bin-by-bin calibration	0.3	0.13

The values in the table show that, although our test device is not a perfect 0.4-ns-LSB TDC, even without bin-by-bin calibration, its maximum measurement error is nevertheless the same as a perfect 1.2-ns-LSB TDC, and its rms error is as small as a perfect 0.7-ns-LSB device. With bin-by-bin calibration, the TDC is at least as accurate as a perfect 0.6-ns-LSB one.

C. *Other FPGA-Based Architectures*

Semiconductor devices are becoming faster and faster. It is very possible that, in the future, the clock period alone in an FPGA counter or shift register will be short enough to meet the TDC LSB requirement for some applications. In this situation, the TDC device would become just a counter or shift register. However, if sub-clock-period precision is needed, delay line structure would still be a good choice. The delay elements are becoming faster along with every other component in the device.

Take the ACEX 1K device EP1K10QC208-1 we used as an example: the typical cascade chain delay we measured was 0.4 ns, while the maximum 16-bit counter clock speed given in the data sheet is 285 MHz, corresponding to a 3.5 ns clock period. This fact would also hold true in faster devices.

Intrinsically, a master-slave flip-flop device needs to establish positive feedback at each clock edge; therefore, the clock period can never be shorter than 4 NAND gate delays. The delay between the Q-output of a flip-flop to the D-input of the next flip-flop further adds onto the clock period. On the other hand, delay elements like the cascade chain we used can be implemented with as few as 2 NAND gate delays.

High-speed serial I/O ports are now becoming popular in FPGA devices. Serial input ports can be used directly as TDCs. However, it is still possible to use cell delay in these devices to exploit the sub-clock-period precision so that a slower device can be used for lower cost.

ACKNOWLEDGMENT

The authors would like to thank Sten U. Hansen and Bill W. Hynes of Fermi National Accelerator Laboratory for many helpful discussions regarding this work.

REFERENCES

[1] Y. Arai, et. al., "Development of a new TDC LSI and a VME module," *IEEE Trans. Nuclear Science*, vol. 49, pp. 1164–1169, Jun. 2002.

[2] J. Christiansen, *32 channel TDC with on-chip buffering and trigger matching* [PDF file]. Available: http://micdigital.web.cern.ch/micdigital/jorgen/tdc32_leb97.PDF

[3] *TDC-F1, High-performance 8-channel TDC* [PDF file]. Available: http://www.acam.de/documents/DBkurz_F1_d.PDF

[4] *CDF TDC documents* [Web site]. Available: http://umwnt1.physics.lsa.umich.edu/tdc/proto_docs.htm

[5] P. Bailly, et. al. , "A 16-channel digital TDC chip," in *Nuclear Science Symposium, 1998. Conference Record*, 1998, vol. 1 pp. 447–452.

[6] J. Kalisz, et. al., "Field-programmable-gate-array-based time-to-digital converter with 200-ps resolution," *IEEE Trans. Instrumentation and Measurement*, vol. 46, pp. 51–55, Feb. 1997.

[7] R. Pelka, et. al., "Nonlinearity correction of the integrated time-to-digital converter with direct coding," *IEEE Trans. Instrumentation and Measurement*, vol. 46, pp. 449–453, Apr. 1997.

[8] M. S. Andaloussi, et. al. , "A novel time-to-digital converter with 150 ps time resolution and 2.5 ns pulse-pair resolution," in *Microelectronics, The 14th International Conference on 2002 - ICM*, 2002, pp. 123–126.

[9] *ACEX™1K Programmable Logic Device Family Data Sheet* [PDF file]. Available: http://www.altera.com/literature/lit-acx.html

[10] *Xilinx FPGA Data Sheets* [Web Site]. Available: http://www.xilinx.com/xlnx/xweb/xil_publications_index.jsp