High-Resolution Time-to-Digital Converter in Field Programmable Gate Array

A. Aloisio [a], P. Branchini [b], R. Cicalese [a], R. Giordano [a], V. Izzo [a], S. Loffredo [b], R. Lomoro [b]

[a] Dipartimento Scienze Fisiche, Università di Napoli "Federico II" and I.N.F.N. Sezione di Napoli - Italy
[b] Dipartimento di Fisica, Università di Roma Tre and I.N.F.N. Sezione di Roma Tre - Italy

loffredo@roma3.infn.it

Abstract

Two high-resolution time-interval measuring systems implemented in a SRAM-based FPGA device are presented. The two methods ought to be used for time interpolation within the system clock cycle. We designed and built a PCB hosting a Virtex-5 Xilinx FPGA. We exploited high stability oscillators to test the two different architectures. In the first method, dedicated carry lines are used to perform fine time measurement, while in the second one a differential tapped delay line is used. In this paper we compare the two architectures and show their performance in terms of stability and resolution.

I. INTRODUCTION

High-resolution Time-to-Digital Converters are often required in many applications in high-energy and nuclear physics. Furthermore, they are widely used in many scientific equipments such as Time-Of-Flight (TOF) spectrometers and distance measurements. Different configurations of tapped delay lines are widely used to measure sub-nanosecond time intervals both in ASIC and FPGA devices. However, the design process of an ASIC device can be expensive, especially if produced in small quantities, while FPGAs lower the development cost and offer more design flexibility. In 1997, Kalisz et al. [1] proposed an FPGA-based approach: their design used a variation of conventional delay line and offered a time resolution of 200 ps. In 2000 [2], rapid progress in electronics technology allowed them to achieve a time resolution of 100 ps. Resolution values between 50 ps and 500 ps have been achieved with this technology [3]. Two different digital delay line circuits have been designed and tested by the authors thus far [4].

II. PRINCIPAL OF OPERATION

We have designed two types of TDC architectures in the newest available Xilinx Virtex-5 FPGA [5]. Both approaches use the classic Nutt method [6] based on the two stage interpolation. The timing acquisition process consists of three phases shown in Fig.1. First, the time interval Δt_1 between the rising edges of the START signal and the subsequent reference clock edge is measured. Then, a coarse counter measures the time interval Δt_{12} between the two rising edges of the reference clock immediately following the START and the STOP signals. The same procedure is exploited to measure the time interval Δt_2 between the rising edges of the STOP signal and the subsequent reference clock. The time interval between the START and STOP signals, Δt, is: Δt_1+ Δt_{12}- Δt_2. The fine conversion dynamic ranges Δt_1 and Δt_2 are limited to only one reference clock cycle. The simplified circuit block diagram of the TDC architecture is shown in Fig.2. The external clock frequency we used was 550 MHz. The Virtex-5 Digital Clock Managers (DCMs) provide a wide range of clock management features and allow phase shifting. We used one DCM that gives four copy of the same clock signal shifted by 0°, 90°, 180° and 270°. In our work, the coarse TDC is designed around a free running 550 MHz syncronous counter. The coarse counter does not allow us to measure Δt_1 and Δt_2 as shown in Fig. 1. Their measurement has been obtained using the same fine time converter for both Δt_1 and Δt_2. We designed two different fine time converters. The first one consists of tapped delay lines, while the second one uses Vernier delay lines. In order to test the two different TDC architectures, we designed and built a PCB hosting a Virtex-5 FPGA from Xilinx, which is shown in Fig.3. Two high stability oscillators from Valpey-Fisher have been installed in order to compare their performance side by side. Test points for high-bandwidth active probes are used to perform the Virtex-5 clock signal characterization. They are placed just near the FPGA, making the shortest distance for the device output signals. SMA connectors are used to send the START and STOP signals to the board. They may adopt differential lines or single ended signaling schemes.

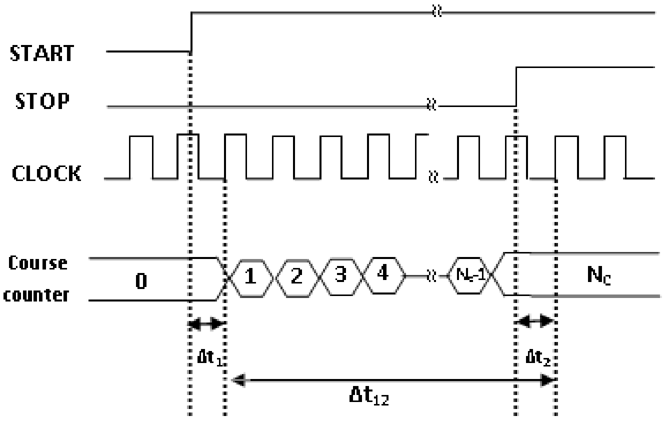

Figure 1: Measurement of time interval T with the Nutt method

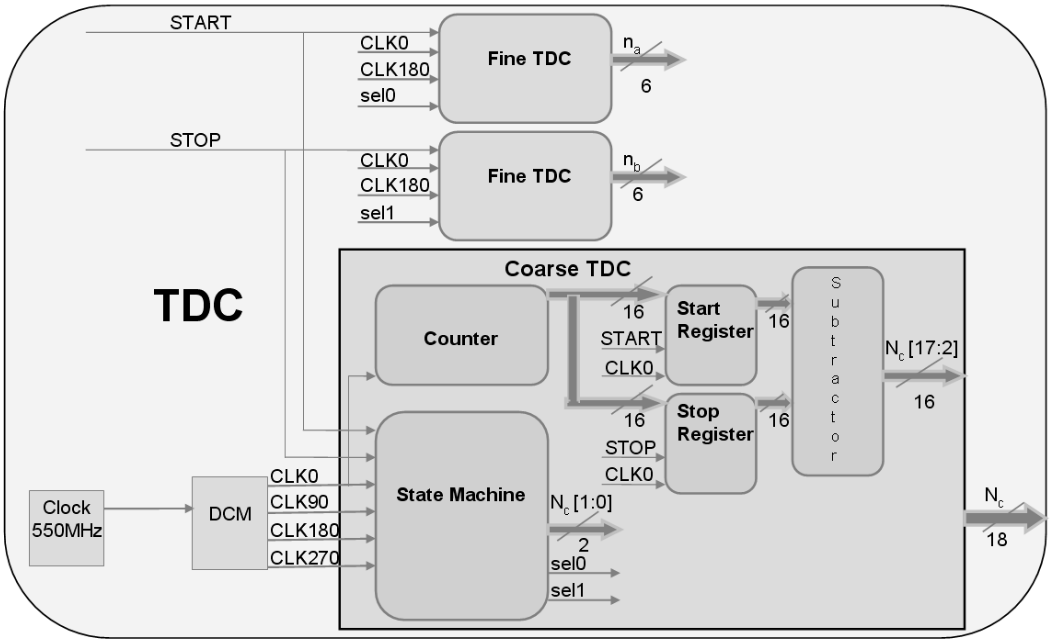

Figure 2: Block diagram of the TDC architecture

A. Time counter

The coarse time measurement is realized by the coarse TDC. The building blocks of the coarse TDC are the 550 MHz syncronous counter and the finite state machine. The counter has a 16 bit data width and is used in free-running mode. When the START signal transition occurs the current state of the counter is sampled by the START register, and the same operation occurs also when the STOP signal is delivered to the TDC.

The difference beetween the STOP and START register is the coarse measurement of the time interval. The state machine is needed to achieve a better resolution of the time interval measurement. It samples the start and stop signals and detects the phase difference between the start and stop rising edges. The least significant bit corresponds to a quarter clock period. The full clock period is recovered by the 2 bit counter $N_c[1:0]$ which labels the phase value.

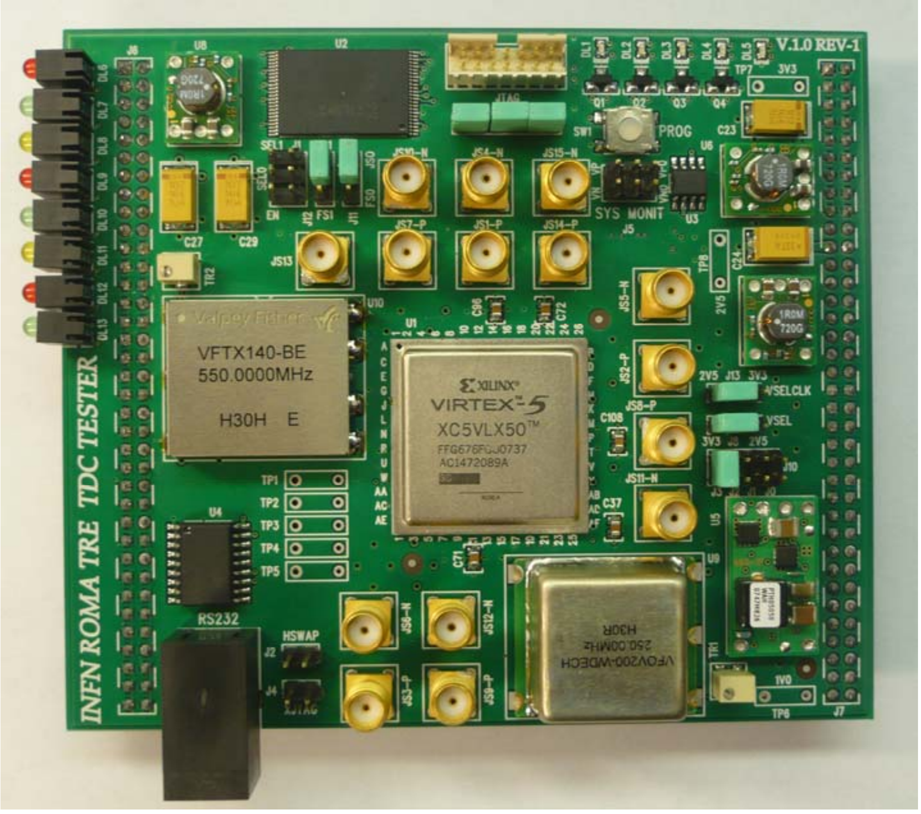

Figure 3: TDC Tester board

The output binary value $N_c[1:0]$ increases the data out width of the coarse TDC, N_c wich is a 18 bit wide word. Therefore the state machine allows us to obtain a coarse resolution of quarter of the CLK0 period (454 ps). Furthermore it is usefull to the delay line selection of the fine TDC performed in the carry chain delay line architecture. The sel0/1 output bits, shown in Fig. 2, follow the phase difference between the START/STOP and the CLK0 signal. This value is 0 if the phase difference is lower than π, 1 if it is bigger. The selection of the tapped delay line of the fine time measurements reflects the phase difference between the START/STOP signal and CLK0. The measurement range of the coarse TDC is limited due to the counter width and the resolution is limited due to the clock frequency.

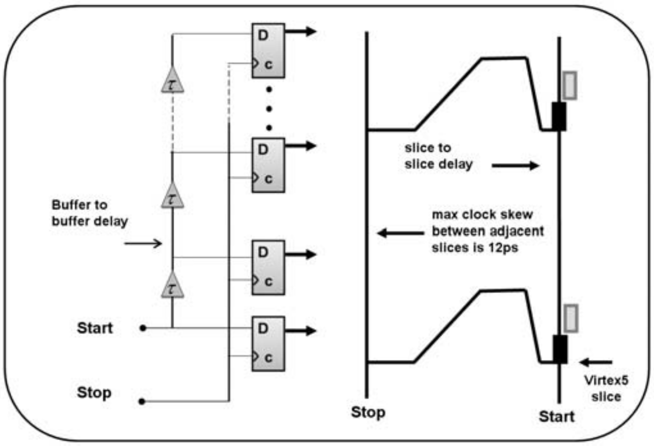

Figure 4: Carry chain delay line. Left: logic block diagram. Right: layout obtained using a Xilinx Virtex-5 FPGA

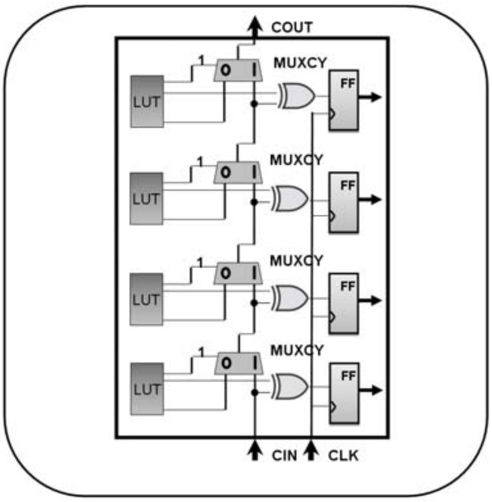

Figure 5: Simplified block diagram of the Virtex-5 slice

B. Carry chain delay line

In the FPGAs available today, there are high-speed chain structures that vendors designed for general-purpose applications. A few well-known examples are carry chains, cascade chains, sum-of-products chains, etc. These chain structures provide short predefined routes between identical logic elements. They are ideal for TDC delay chain implementation. The first architecture, shown in Fig.4 (left) uses carry chain delays. In this configuration the STOP signal is the 550 MHz system clock. The START signal after each delay unit is sampled by the corresponding flip-flop on the rising edge of the STOP signal. The tapped delay line layout is presented in Fig.4 (right) while in Fig.5 a simplified block diagram of the Virtex-5 slice is shown. In this configuration the delay line is created by a train of 64 multiplexers. The selection bit of every multiplexer is set to logic one, in order to let the START signal propagate through the line. The time quantization step of the TDC is determined by the multiplexers propagation delay τ. Due to the short delay of the tapped delay line, it's necessary to use two delay lines in order to cover the duration of a clock period. The first line is clocked by CLK0 and the second line by CLK180. The state machine decides which line is selected by setting the sel0/1 line to 0 or 1 logic level. We used the multiplexing of two delay lines rather then a longer line, to reduce the possible non linearity introduced by the clock distribution delay time between neighboring slices. Furthermore in this way, the output from the tapped line is more easily converted from thermometric code into binary natural code by using a priority encoder.

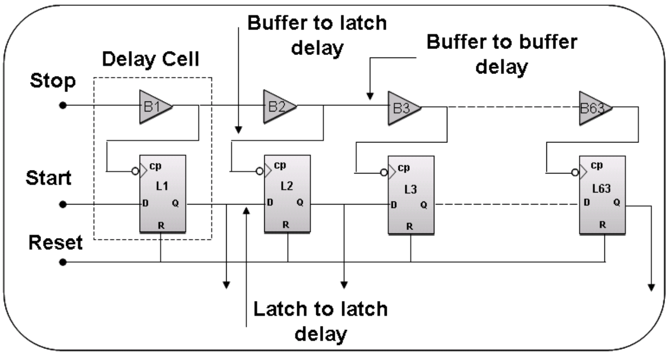

Figure 6: Vernier delay line: logic block diagram

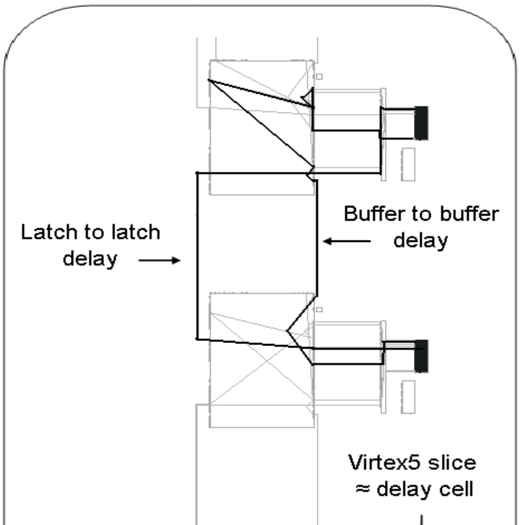

Figure 7: Layout of the Vernier delay line obtained using a Xilinx Virtex-5 FPGA

C. Vernier delay line

The second architecture, shown in Fig.6, consists of two tapped rows working in differential mode. The first is created as a chain of the latch flip-flops L1, L2, ..., L63 and the other as a chain of noninverting buffers B1, B2, ..., B63. Hence the basic delay cell contains one latch having the delay τ_1 and one buffer having the delay τ_2. If the latch delay is longer than the buffer one, the time quantization step of the TDC is determined by their difference τ_1- τ_2. An advantage of that direct coding is very short conversion time and very short dead time equal to the readout time of the output time. The time to be measured is defined between the rising edges of the pulses START and STOP. During the time-to-digital conversion process, the STOP pulse follows the START pulse along the line and all latches from the first cell up to the cell where the START pulse overtakes the STOP pulse are consecutively set. In the used configuration, the output from the tapped line is obtained in thermometric code and then converted into binary natural code. This is been possible thanks to an array of priority encoder realized on the FPGA. In this approach, the reset input signal is given to all latches contemporaneous only after the end of the acquisition time. As it is known, the current FPGA technologies offer logic cells to implement logic defined by the user and in particular the delay cell of the TDC. Fig.7 shows the layout of the Vernier delay line realized using the Virtex-5 slices.

To implement the design in FPGA, one must address one major problem: in the FPGA development software, a logic element (combinatorial or sequential) can be physically placed in nearly any place, depending on the optimization algorithm used. When left up to the program, routing between logic elements may also be unpredictable to the user. If the logic elements used for the architectures are placed and routed in this fashion, the propagation delay of each delay step will not be uniform. To avoid this, the designer is forced to place and route the logical resources by hand.

III. Test bench

Preliminary tests have been made on our delay lines using the two architectures on the TDC Tester board. Each TDC structure has 64 steps. To execute our tests we have used an architecture based on an embedded microprocessor, as shown in Fig. 8.

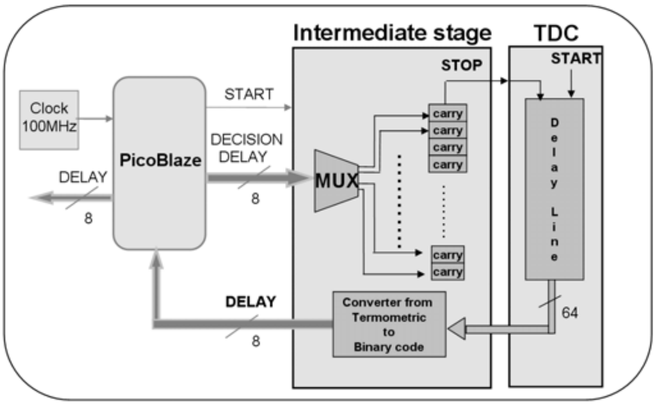

Figure 8: Communication between PicoBlaze and the TDC delay line

PicoBlaze [7] is a FPGA based microprocessor which has an 8-bit address and data ports to access a wide range of peripherals. The PicoBlaze firmware allows the user to enter a delay value via a RS232 link. The intermediate stage receives data bus, decodes it and establishes which is the value delay. Each signal is connected to the respective carry. In this way arrival time (STOP) is changed by using carry of various lengths. Carry chain has been used to generate the delays because for each step they can be considered fixed for the particular physical technology, rail voltage and temperature range.

IV. Test results

The time interval between START and STOP has been calibrated and then it is measured by TDC. Fig. 9 show a test result of the two architecture TDC outputs as a function of the signal input time. More than 1000 measurements were made for each point and the average of each set of measurements was plotted. In Fig. 9 a linear fit, to guide the eye, is superimposed on experimental data. Comparable resolution, of about 80 ps have been measured for the two different delay line designs. Some non-uniformity are due to the internal layout structure of the device.

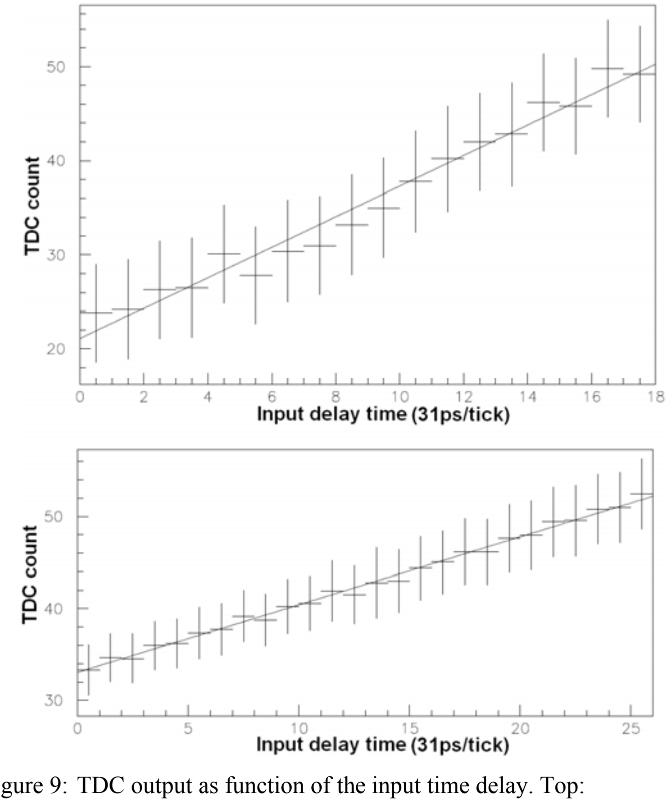

Figure 9: TDC output as function of the input time delay. Top: implementation of the carry delay line. Bottom: implementation of the Vernier delay line

V. Conclusion

Semiconductor devices are becoming faster and faster. This allows us to have high resolution digital counter and short delay elements. Therefore, it is possible to develop a low cost and high resolution TDC exploiting FPGAs based techniques. By using SRAM-based FPGAs, the user benefits from the in-system-programming (ISP) and reconfiguration features increasing the flexibility and reliability of the measuring system. Resolution values of about 80 ps have been achieved.

VI. References

[1] J. Kalisz, R. Szplet, J. Pasierbinski, and A. Poniecki, “Field-Programmable-Gate-Array-Based Time-to-Digital Converter with 200-ps Resolution,” IEEE Trans. Instrum. Meas., vol. 46, no. 1, pp. 51-55, Feb. 1997.

[2] R. Szplet, J. Kalisz, and R. Szymanowski, “Interpolating Time Counter with 100 ps Resolution on a Single FPGA Device,” IEEE Trans. Instrum. Meas., vol. 49, no. 4, pp. 879-883, Aug. 2000.

[3] J. Song, Q. An, and S. Liu, “A High-Resolution Time-to-Digital Converter Implemented in Field-Programmable-GateArrays,” IEEE Trans. Nucl. Sci., vol. 53, no. 1, pp. 236-241, Feb. 2006 and references therein.

[4] A. Aloisio, P. Branchini, R. Cicalese, R. Giordano, V. Izzo and S. Loffredo “FPGA Implementation of a High-Resolution Time-to-Digital Converter,” 2007 IEEE Nuclear Science Symposium Conference Record, Volume 1, 2007, 504-507.

[5] Virtex-5 FPGA User Guide (2008, September). [Online]. Available:
http://direct.xilinx.com/bvdocs/userguides/ug190.pdf

[6] J. Kalisz, “Review of methods for time interval measurements with picosecond resolution,” *Metrologis*, vol.41, pp.99-145.

[7] PicoBlaze 8-bit Embedded Microcontroller User Guide (2008, June). [Online]. Available:
http://www.xilinx.com/support/documentation/ip_documentation/ug129.pdf