Implementation of High-Resolution Time-to-Digital Converters on two different FPGA devices

R. Cicalese*, A. Aloisio, P. Branchini, R. Giordano,
V. Izzo, S. Loffredo

Dipartimento Scienze Fisiche, Università di Napoli Federico II and I.N.F.N. Sezione di Napoli - Italy,
Via Cintia 2 - 80126 Naples, Italy
**E-mail: cicalese@na.infn.it*

This paper describes the development of two high precision Time-to-Digital Converter (TDC) in two different SRAM-based FPGA devices. The time conversion is based on a course counter for long range and on a stabilised delay line for the time interpolation within the system clock cycle. 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.

1. Introduction

Time to Digital Converter (TDC) has been widely used in many applications such as particle detection in high energy physics, laser range finder, frequency counter and on-chip jitter measurement. For many years the main methods used to achieve the hundreds of pico-seconds resolution have been based on time-stretching, Vernier and tapped delay line. These technique were designed both in ASIC and FPGA devices. However, the design process of an ASIC device not only can be expensive, especially if produced in small quantities but also the design process is complex due to the long turn-around time and layout phase. On the other hand, 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 without using any external circuits. One of the first TDC realized on an FPGA-based approach was proposed by Kalisz [1] et al. in 1995. They made use of the difference between a latch delay and a buffer delay of QuickLogics FPGA and achieved a time resolution of 100 ps. In

2003, a TDC was implemented in an ACEX 1 K FPGA from Altera by Wu [2] et al. This TDC used cascade chains of the FPGA and offered a time resolution of 400 ps. In the same year, Szymanowski, et al. implemented a high-resolution TDC with two stage interpolators in a QL12X16B from QuickLogic [3] . The TDC had 200 ps resolution and standard measurement uncertainty below 140 ps. In 2004, a TDC was implemented in a general purpose FPGA device by using dedicated carry lines in the FPGA to perform time interpolation by Qi An [4] et al.

2. Principle of operations

We have designed two types of TDC architectures in different Xilinx FPGAs. Both approaches use the classic Nutt [5] method based on the two stage interpolation. The timing acquisition process consists of three phases. First, the time interval (Δt_1) between the rising edges of the start signal and the succeeding reference clock is measured. Secondly, a coarse counter is activated to measure the time interval (Δt_{12}) between the two rising edges of the reference clock immediately following the START and the STOP signals. Finally, the same procedure for measuring the time interval (Δt_2) between the rising edges of the STOP signal and the succeeding reference clock is also performed. The time interval between the START and STOP signals, (Δt), may be determined as ($\Delta t_1 + \Delta t_{12} - \Delta t_2$). The dynamic range for fine conversion of (Δt_1) and (Δt_2) is limited to only one reference clock cycle. The principle of operations is presented in Fig.1 as a conceptual timing diagram.

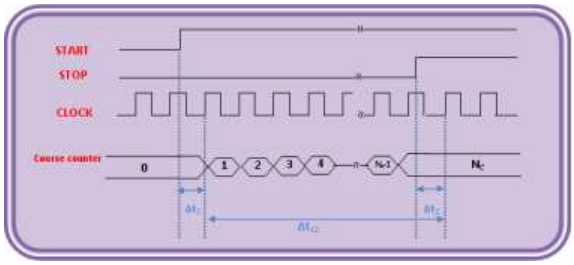

Fig. 1. Measurement of time interval T with the Nutt method.

In the FPGA available today, there are chain structures that the 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. In our works, the fine TDC for the measurement of the short intervals, (Δt_1) and (Δt_2), have been performed in two different methods:

1. The first architecture, shown in Fig.2 (a) uses carry chain delays, like the one present in the newest available Xilinx Virtex 5 FPGA. The START signal after each delay unit is sampled by the pertaining flip-flop on the rising edge of the STOP signal.

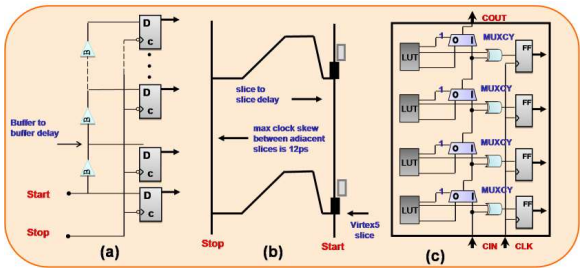

Fig. 2. Carry chain delay line: (a) logic block diagram; (b) Layout obtained using a Xilinx Virtex 5 FPGA; (c) simplified block diagram of the Virtex 5 slice.

2. The second architecture, shown in Fig.3 (a),
uses two rows of slightly different cell delays and it makes a differential delay line by using the Xilinx Virtex II' s slices. 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.

To implement the designs in FPGA, one must address one major problem: 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 be unpredictable to the user. If the logic cells 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. The two layouts are presented respectively in Fig.2 (b) and Fig.3 (b). In Fig.2 (c) and Fig.3 (c) simplified block diagrams of the two Virtex 5 and Virtex II are shown.

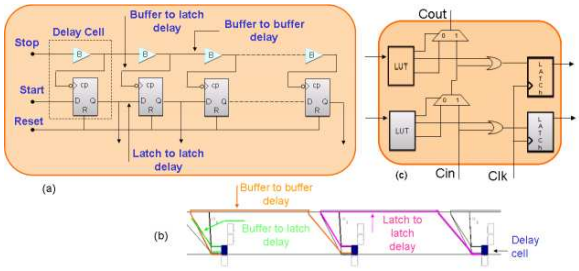

Fig. 3. Vernier delay line: (a) logic block diagram; (b) layout obtained using a Xilinx Virtex II FPGA; (c) simplified block diagram of the Virtex II slice.

3. Test bench results

Preliminary tests have been made on our delay lines using the first architecture on a Xilinx Virtex 5 [6] demo board and the second one on the Xilinx Virtex 2 [7] demo board. Each TDC structure has 64 steps. The external clock frequency we used in the tests was 100 MHz. To execute our tests we have used an architecture based on an embedded microprocessor Fig.4.

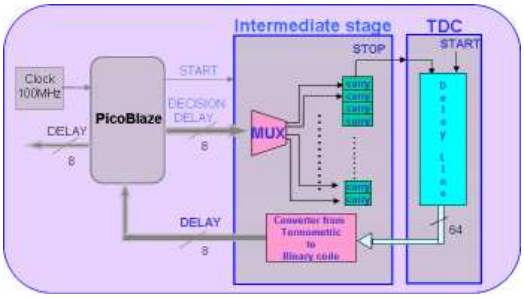

Fig. 4. Communication between PicoBlaze and the TDC delay line.

PicoBlaze is a FPGA based microprocessor which has an 8-bit address and data port to access a wide range of peripherals. PicoBlaze allows the user to input 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. So

the time interval between START and STOP has been determined and then it is measured by TDC. Fig.5 shows a test result of the second architecture TDC output as a function of the signal input time.

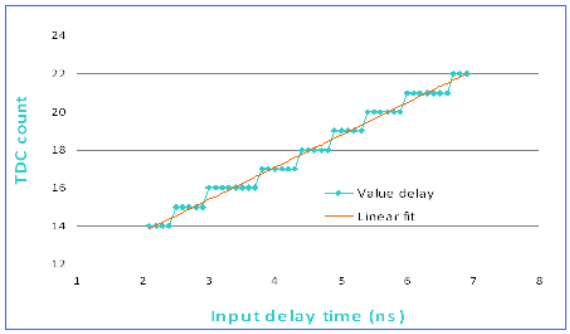

Fig. 5. TDC output as function of the input time delay.

More than 1000 measurements were made for each point and the average of each set of measurements was plotted. A simple linear fit shows that the least significant bit (LSB) bin size of the TDC structure in this device was about 0.5 ns. One would also expect some non-uniformity due to internal layout structure of the device.

References

1. J. Kalisz, R. Szplet, 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. 5155, Feb. 1997.
2. J. Wu, Z. Shi, and I. Y. Wang, *Firmware-only implementation of time-to-digital converter in field programmable gate array*,in Proc. IEEE Conf. Rec. NSS., vol. 1, 2003, pp. 177181.
3. R. Szymanowski and J. Kalisz, *Field programmable gate array time counter with two-stage interpolation*, Rev. Sci. Instrum., vol. 76, 2005. 045 104.
4. Jian Song, Qi An, and Shubin Liu, *A high-resolution Time-to-Digital Converter Implemented in field programmable gate array*, IEEE Trans. Instrum. Meas., vol. 53, no. 1, Feb. 2006.
5. J. Kalisz, *Review of methods for time interval measurements with picosecond resolution*, Metrologis, vol.41, pp.99-145.
6. *Virtex-5 User Guide (2007, Feb.)[Online].* http://direct.xilinx.com/bvdocs/userguides/ug190.pdf
7. *Virtex-II Platform FPGA User Guide (2007, March). [Online].* Available: http://direct.xilinx.com/bvdocs/userguides/ug002.pdf