EMPLOYING FPGA DSP BLOCKS FOR TIME-TO-DIGITAL CONVERSION

Paweł Kwiatkowski

Military University of Technology, Faculty of Electronics, Gen. S. Kaliskiego 2, 00-908 Warsaw, Poland
(✉ *pawel.kwiatkowski@wat.edu.pl,* +48 261 839 602)

Abstract

The paper presents a novel implementation of a time-to-digital converter (TDC) in field-programmable gate array (FPGA) devices. The design employs FPGA digital signal processing (DSP) blocks and gives more than two-fold improvement in mean resolution in comparison with the common conversion method (carry chain-based time coding line). Two TDCs are presented and tested depending on DSP configuration. The converters were implemented in a Kintex-7 FPGA device manufactured by *Xilinx* in 28 nm CMOS process. The tests performed show possibilities to obtain mean resolution of 4.2 ps but measurement precision is limited to at most 15 ps mainly due to high conversion nonlinearities. The presented solution saves FPGA programmable logic blocks and has an advantage of a wider operation range when compared with a carry chain-based time coding line.

Keywords: time-to-digital converter, time coding line, time interval counter, digital signal processing, field-programmable gate array.

1. Introduction

Precise time-to-digital conversion is widely used in many different areas. Particle and nuclear physics [1, 2], laser range finding [3], ultrasonic flow measurement [4], positron emission tomography [5] or time and frequency distribution [6] are just merely selected examples where it is applicable. Electronic circuits that perform this task are called *time-to-digital converters* (TDCs). The TDCs are widely implemented with the use of *field-programmable gate array* (FPGA) technology. The main advantages of this technology are lower cost and shorter time-to-market comparing with the use of *application-specific integrated circuits* (ASICs). In the past the FPGA was defined as a device that is built from three main types of elements, *i.e.* (1) I/O blocks, (2) a matrix of programmable logic blocks and (3) routing resources placed between them. In general, it is still true but nowadays there can be found in it lots of additional dedicated functional blocks, such as *phase or delay locked loops* (PLL or DLL, respectively), serializers/deserializers, *digital signal processing* (DSP) units, hard processors, memory blocks, *etc*. Some of them greatly facilitate the implementation of TDCs, *e.g.* PLL/DLL can easily generate a multiphase clock [7],

a block memory is often used to store calibration data [8] and a hard processor can process raw measurement data as well as provides common communication interfaces [9].

In the literature a vast amount of works related to FPGA TDC implementation can be found in recent years [5, 8–25]. The vital element of most high-performance TDCs are picosecond-resolution tapped delay lines [26]. A *time coding line* (TCL) directly converts a single delay buffer propagation time into measurement resolution. This design is commonly implemented in FPGAs with the use of carry chain elements that are often the core part of a programmable logic block [27]. Several methods were proposed to deal with carry chain-based TCL conversion nonlinearities as well as to improve resolution beyond buffer propagation time, *e.g.* a tuned delay line [11, 22], multiple TCLs [10, 13] or a wave union [12, 24, 28]. A Vernier method employs two delay lines with slightly different propagation times and can be implemented using, *i.a.*, FPGA routing resources [29, 30]. A variant of the method is the one using two oscillators of different frequencies [15]. The difference between low-to-high and high-to-low delay element propagation times is used in the pulse shrinking method [21, 31]. The mentioned methods are vulnerable to *process, voltage and temperature* (PVT) variations, but can provide outstandingly low measurement uncertainty. When PVT resistance or logic resource saving is expected more than the picosecond resolution and measurement uncertainty, the PLL can be used to create a multiphase clock. The multiphase clock is then used in a user-defined circuit implemented in programmable logic to perform time-to-digital conversion [7, 20, 32]. Similar functionality is provided also by another dedicated block, the serializer/deserializer, that is commonly implemented as a part of FPGA I/O buffers [33–36].

Regardless a vast amount of FPGA-based TDC circuits the use of DSP blocks in this context is a new concept. The DSP blocks contain, *i.a.*, fast adders whose output can be directly connected to registers. Thus, this structure seems to be suitable to implement a high-resolution TDC. First works regarding this topic have been published very recently [37, 38] and this paper presents the results obtained in independently and simultaneously performed research.

2. DSP slice configuration

A typical DSP operation is called *multiply-and-accumulate* (MAC). Such an operation computes the product of two numbers and adds that product to the accumulator. In hardware it is implemented as a multiplier followed by an adder and a register. In an FPGA device the MAC operation can be performed using programmable logic blocks. However, this solution either consumes a lot of logic resources or provides low processing speed. The response of the main FPGA device manufacturers (mainly *Xilinx* and *Intel*, formerly *Altera*) to fast signal processing was to place dedicated DSP blocks in a chip. Modern FPGA devices can contain even thousands of DSP blocks that can be either connected in series to perform operations on wide data buses or work in parallel to speed up the processing. While the detailed DSP block structure may differ between manufacturers and chip generations, the core elements such as the multiplier, adder and register are rather common.

The typical digital way of TDC implementation employs a tapped delay line and a register and is known as the *time coding line* (TCL). One signal (*start*) propagates along the delay line and the other (*stop*) causes latching of the register flip-flops states. The first signal reaches subsequent register flip-flop data inputs with the delay equal to a single delay line buffer propagation time. Thus, the number of flip-flop outputs that store logic states "1" is proportional to the time difference between active edges of *start* and *stop* signals. The result is then represented in the unary code (thermometer code). However, in modern TCL solutions, especially implemented in

FPGA devices, it is hard to achieve strictly this code [11, 13]. Looking carefully at different tapped delay line architectures used in TDCs [25] one may conclude that it is important to obtain a possibly high number of output codes whose occurrence is evenly spread in the converter measurement range. Such reasoning has led to the creation of, *e.g.*, the BOUNCE architecture [39]. Therefore, a delay line can be implemented as any kind of one-input multiple-output combinational circuit that directly transmits an input signal to outputs but with different propagation times for each output tap. Following this reasoning, a binary adder with one data input set to the maximum value (logic states "1...1") and another one connected to the start signal (one bit that changes from logic state "0" to "1") is also applicable as a delay line. Thus, the fast DSP adders that can be directly connected to the registers look perfect for the TCL implementation.

In Kintex-7 FPGA, manufactured by *Xilinx*, the basic DSP element is called the DSP48E1 slice [40]. Its simplified structure along with configuration as a TDC is presented in Fig. 1. Each DSP slice has four data inputs (A, B, C and D) and one data output (P), a 25-bit pre-adder, a 25×18 multiplier and a 48-bit *arithmetic logic unit* (ALU). To speed up DSP operation pipeline registers are placed after each combinational circuit as well as at the data inputs. Several DSP slices can be cascaded using dedicated connection paths. While there are two data-paths and three DSP slice outputs that provide the cascade capability, only one path (ACIN-ACOUT) enables to connect several DSP slice adders or ALUs and bypass pipeline registers (the other connections are not shown in Fig. 1. for simplicity). Thus, this path is suitable to implement a long-range delay line for the start pulse. The DSP slice configuration is supported by software tools, such as *Xilinx LogicCORE DSP48 Macro*, making TDC implementation convenient.
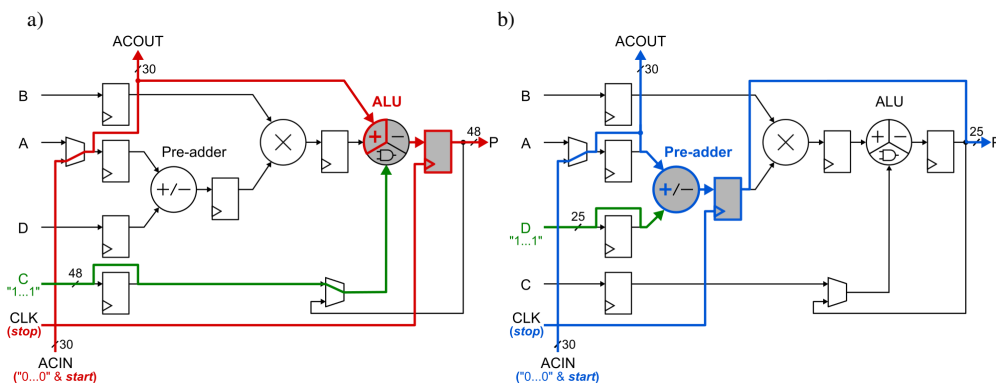

Fig. 1. A simplified DSP48E1 slice [40] configuration as a TDC with the use of ALU (a) or a pre-adder (b).

In this paper two DSP slice configurations are considered as TDC. In both configurations the start signal is connected to the first bit of data port A and then propagates along the ACIN-ACOUT path to neighbouring DSP slices. In the first configuration (Fig. 1a) the ALU operates as a 48-bit adder and all nets of data port C are set to logic high ("1"). In the second configuration (Fig. 1b) a pre-adder is employed and all nets of data port D are set to logic high ("1"). In both designs the occurrence of the start pulse causes overflow in the adder. The time to change the logic states on the adder outputs differs giving various output codes depending on the measured time interval. While the detailed pre-adder and ALU hardware structures are unknown to the FPGA designer, in terms of TDC implementation it is important to obtain a possibly high number of output codes evenly distributed over a narrow operation range. An advantage of the solution with ALU gives a possibly higher resolution due to a wider data output (48-bit in comparison to 25-bit with the

pre-adder). The question is how evenly are the output codes distributed over the TDC operation range and how sophisticated are the data-paths in the pre-adder and ALU. The more sophisticated path from the start input to the register flip-flops, the worse measurement precision due to jitter added to the measured signal by each element in the signal path.

3. Time interval counter design

The DSP TDCs were implemented and tested as an interpolator circuit in an integrated time interval counter shown in Fig. 2. The main counter counts subsequent periods of a clock signal (T_{CLK}) in the continuous mode. The clock signal frequency is set to 700 MHz ($T_{CLK} \approx 1.429$ ns). An input signal is synchronised in a double synchronizer with the clock signal. Then, it causes storing the main counter content in the related channel register and is used as a *stop* signal in the DSP TDC. Simultaneously, the input signal is distributed through a delay line to the DSP TDC as a *start* signal. The delay line is used to compensate the double synchronizer delay. The delayed input signal propagates along DSP slice carry path (ACIN-ACOUT, Fig. 1) causing a change in the pre-adder or ALU output states from logic "1" to "0". The changes on a particular pre-adder or ALU bits reach inputs of flip-flops in the pipeline registers with different delays. The pipeline registers detect and store the number of bits which have changed their logic state to "0" before the *stop* signal appeared. A zeros-counter converts raw DSP TDC data (output P) into a natural binary code. The number of output P bits that store low logic state is proportional to the length of the time interval measured in the interpolator. The data stored in the channel register, in the form of the number of full clock periods counted in the main counter, is the coarse part of the measurement result (T_{coarse}). The output of DSP TDC converted in the zeros-counter is the fine part (T_{fine}), measured in the range of the clock period. These data are then processed in a code processor [13, 18] to calculate a measurement result as $T_{coarse} \times T_{CLK} - T_{fine}$ with picosecond resolution. The final result is stored in a FIFO memory and can be read by a control computer. Both counter channels are designed in the same way.
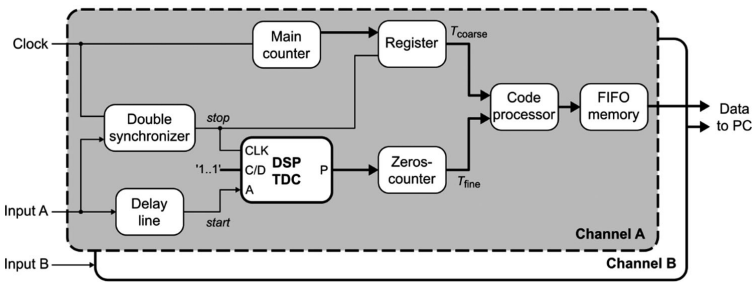

Fig. 2. A block diagram of the integrated time interval counter.

The measurement data are provided in the form of timestamps, *i.e.*, time moments of input pulse occurrence with regard to the virtual time scale created by the main counter [13, 18]. The main counter size is 28 bits and is driven by a clock of 700 MHz frequency. This enables to create about 383 ms-long time scale. Then, the main counter overflows and starts to count from the beginning. The time intervals between particular input pulses can be calculated as a difference between selected timestamps.

The T_{fine} value serves as the address input to a memory built in the code processor block [13, 18]. During initialization (counter calibration) the memory is filled up with the result of

the *statistical code density test* (SCDT) [41]. Then, during measurement, T_{fine} indicates the memory address where the actual interpolator transfer characteristic value is stored. This procedure minimizes the influence of DSP TDC integral nonlinearities on the time counter accuracy and precision.

The DSP TDC operation range has to be at least equal to the clock signal period. It was experimentally tested that the measurement resolution is either 4.2 ps (ALU) or 8.1 ps (preadder). In both cases there can be generated 340 and 176 output codes, respectively. While the DSP slice output bus width is either 48-bit or 25-bit (output P, Fig. 1) at least seven DSP slices have to be connected in series to build the TDC. Taking into account voltage and temperature drift, the TDC was implemented with the use of ten DSP slices. This size provides the overall operation range of about 2 ns ($10 \times 48 \times 4.2$ ps or $10 \times 25 \times 8.1$ ps) and is crucial during the design implementation in the FPGA.

4. Implementation issues

The Kintex-7 FPGA takes advantage of the *Advanced Silicon Modular Block* (ASMBL) architecture created by *Xilinx* [42]. The idea is that the chip is divided into several columns of different kinds of logic, such as *configurable logic blocks* (CLBs), DSP slices, memory blocks, I/O ports, *etc*. A large size of modern FPGA chips, in terms of logic resources, forces the necessity to use extensive clock distribution networks. These networks further divide programmable logic resources to several areas called clock regions. A clock skew along a single clock region is only several picoseconds. However, it can reach even tens of picoseconds at the border of clock regions. The clock network is often used to distribute the stop signal with a possibly low skew to all register flip-flops. The difference between propagation times of the TDC start and stop signals to two successive flip-flops in TCL determines the size of quantization step width (bin size). A large clock skew may result in a loss of transfer function monotonicity or the existence of so called *ultra-wide bins* (UWBs) [27]. Thus, as a rule of thumb, the designers try to shorten the TDC tapped delay line length to fit a single clock region. The commonly used carry chain-based TDC implemented in one clock region of the Kintex-7 device provides an operation range of about 2.2 ns (200 carry multiplexers with a mean propagation time of 11 ps each). A similarly implemented DSP TDC, *i.e.* in the full range of the clock region, has about twice as wide operation range (4 ns).

5. Experimental tests

5.1. Test setup

The designed DSP TDC was implemented and tested in the measurement module of MTC 108 multichannel time interval counter (Fig. 3) [18]. The module provides *i.a.* a low-noise power supply, a high-frequency clock generator based on a low-jitter programmable digital frequency synthesizer and a built-in square-wave generator for calibration purposes. The FPGA device was reprogrammed to implement the presented integrated time interval counter design (Fig. 2). Input circuits form the steep edge of measured input pulses and standardize their amplitude. A stable 10 MHz clock signal (0.5 ppm) from the *temperature-compensated crystal oscillator* (TCXO) is used in the synthesizer to produce a 700 MHz clock signal for the integrated counter. The square-wave generator is used to perform calibration procedures with the use of SCDT [41] and the code processors. The code processor unit automatically processes the data obtained from the SCDT and calculates the DSP TDC transfer function.
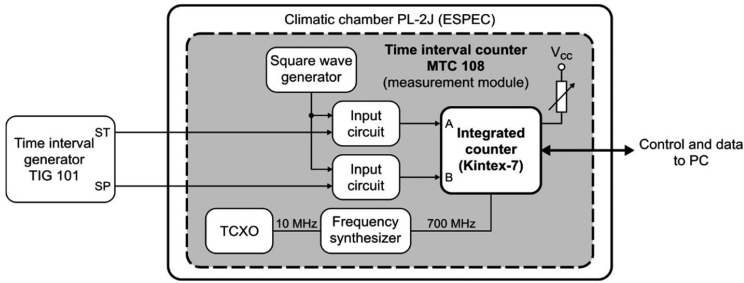
Fig. 3. The test setup.

Reference time intervals are produced by a programmable time interval generator TIG 101 [43]. Generator outputs ST and SP produce pulses that correspond to the beginning and the end of time intervals, respectively. These pulses are then measured by integrated time interval counter channels A and B. Most tests were executed in the laboratory conditions with ambient temperature of about 21°C. For the temperature tests the MTC 108 counter was placed in a climatic chamber PL-2J (*Espec*).

The tests were executed for three time interval counter designs that differ as to the TDC used. A single carry chain-based TCL was implemented (TDC$_{carry}$) as a reference design [13, 27]. This design employs the well-known TDC architecture built with the use of 200 fast carry multiplexers. The second and third designs differ in the DSP slice configuration described in Section 2 (Fig. 1) and – depending on DSP logic resources used – they will be further referred to as TDC$_{pre-adder}$ and TDC$_{ALU}$.

5.2. Resolution and nonlinearities

After the design start-up the code processors automatically execute the statistical code density test using 2×10^6 measurement samples generated by the square-wave generator. Then, the code processor memory stores the data that represent DSP TDC bin sizes and transfer function. The results for three different TDC designs are presented in Fig. 4, where LSB stands for least significant bit and represents the resolution value.

The TDC$_{ALU}$ has about 2.5 times better resolution (LSB = 4.23 ps) than the reference design TDC$_{carry}$ (LSB = 10.7 ps). However, the improvement is impressive only as long as nonlinearities are not taken into account. While the TDC$_{carry}$ consists of 200 carry multiplexers to fulfil the measurement range requirement (at least 1.43 ns) with some margin, it needs to be implemented in the full range of programmable logic block column in a single clock region. The clock network crosses the region in the middle resulting in a bit longer carry multiplexers' connection path and, as a result, a bin larger than the other ones (the 101th bin, 30 ps large). The DSP TDC is implemented in one half of the clock region bypassing this problem. However, in these designs (TDC$_{ALU}$, TDC$_{pre-adder}$) UWBs occur at every border of DSP slices (either every 48th or 25th bit) and may reach up to 90 ps. On the one hand, the existence of many *ultra-narrow bins* (UNBs) greatly improves resolution, but on the other – UWBs significantly reduce achievable precision. The TDC$_{pre-adder}$ with 8.12 ps resolution and up to 33 ps wide UWBs is an intermediate solution between TDC$_{ALU}$ and TDC$_{carry}$.

Figure 5 presents *differential nonlinearities* (DNL) calculated for all three TDC designs. In TDC$_{ALU}$ about five sixth of bins are shorter than the LSB value and several bins are more than 5 times wider. The TDC$_{pre-adder}$ has more uniform bin sizes but still the DNL reaches high values
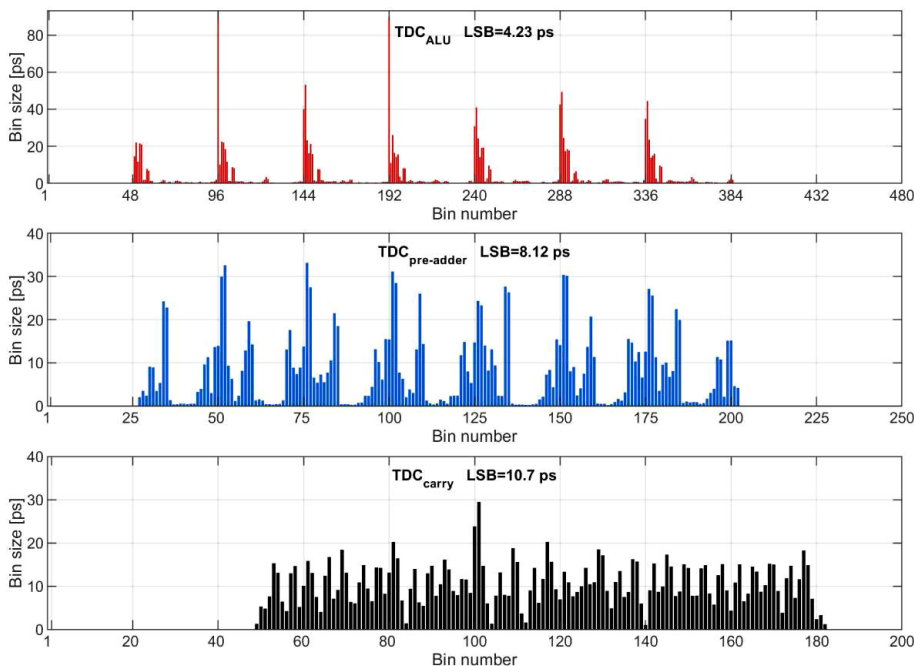
Fig. 4. Bin sizes in TDCs based on the DSP's ALU, the DSP's pre-adder and the carry chain.
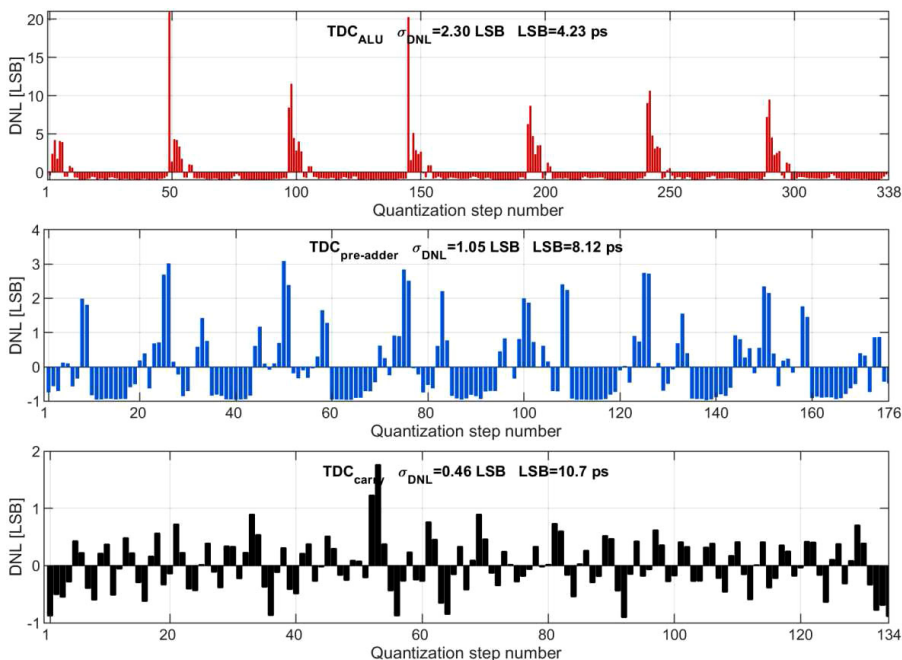

Fig. 5. Plots of *differential nonlinearities* (DNL) of TDCs based on the DSP's ALU, the DSP's
pre-adder and the carry chain.

(2–3 LSB). In the TDC$_{ALU}$ the DNL$_{max}$ is 20.93 LSB and the standard deviation of the DNL (σ_{DNL}) values is 2.48 LSB. In the TDC$_{pre-adder}$ the same parameters reach 3.08 LSB and 1.05 LSB, respectively. A classic TDC$_{carry}$ design, paradoxically known for relatively high nonlinearity, has only two bins twice as wide than the mean value and "only" 0.46 LSB standard deviation of DNL values.

Another parameter that can be used to directly compare highly nonlinear TDCs is called an equivalent resolution. It was originally proposed by Jinyuan Wu and firstly described in the papers [44, 45]. The equivalent resolution (w_{eq}) represents the capability of measurement precision that TDC is able to achieve due to quantization noise and DNL, and is calculated as follows:

$$w_{eq} = \sqrt{\sum_i \left(\frac{w_i^3}{W} \right)}, \tag{1}$$

where w_i is the i-th bin size and W is the total range of TDC. In the considered case W is equal to the clock period $T_{CLK} = 1.429$ ns. The w_{eq} calculated for TDC$_{ALU}$, TDC$_{pre-adder}$ and TDC$_{carry}$ is 40.8 ps, 19.3 ps and 13.8 ps, respectively. The DSP TDC based on ALU has at the same time 2.5 times better mean resolution and 3 times worse equivalent resolution. This has a significant effect on TDC precision.

5.3. Process variation

The TDCs were implemented in several different columns of DSP slices. The resolution changes depended on a selected location. The biggest differences were observed in the case of TDC$_{ALU}$ and w_{eq}. Fig. 6 presents bin sizes for an example of TDC$_{ALU}$ implementation in three FPGA programmable logic areas. The most interesting fact is that the widest UWBs (>90 ps) occur at the border of only some DSP slices but not of all. Thus, it is possible to find the optimal DSP location in a particular FPGA chip that maximizes resolution. In the selected Kintex-7 FPGA device there are 6 columns of 100 DSP slices each, giving plenty of possible implementation locations. While this process is cumbersome and time-consuming it may improve w_{eq} by about 20% in the case of TDC$_{ALU}$ (from about 46 ps to 37 ps).
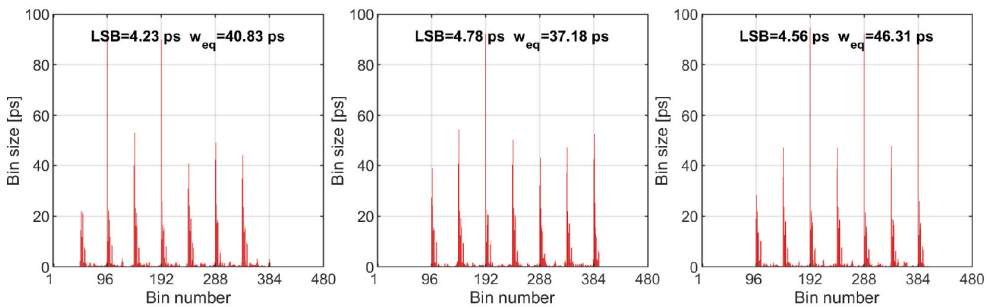

Fig. 6. UWB occurrence in the case of TDC$_{ALU}$ implementation in 3 different columns of DSP slices.

5.4. Voltage and temperature sensitivity

The voltage and temperature tests were performed in two steps. First, the MTC 108 measurement module was tested in the laboratory conditions with ambient temperature of about 21°C and the FPGA core voltage (V$_{cc}$) was changed from 0.970 V to 1.027 V with a 5 mV step. Then, the

voltage was set to 1.027 V and the module was placed in the climatic chamber where ambient temperature was changed from −10°C to +70°C with a 10°C step. Actual core voltage values and device temperatures were checked using an *Xilinx Analog-to-Digital Converter* (XADC) buil-in the FPGA device. In each test the FPGA chip temperature was about 10°C higher than ambient temperature. Thus, the FPGA device was operating in the full range of allowed temperatures (0°C − 85°C). The integrated time interval counter was calibrated at each core voltage and temperature. Then, the mean and equivalent resolutions were calculated. The obtained results are summarized in Fig. 7.
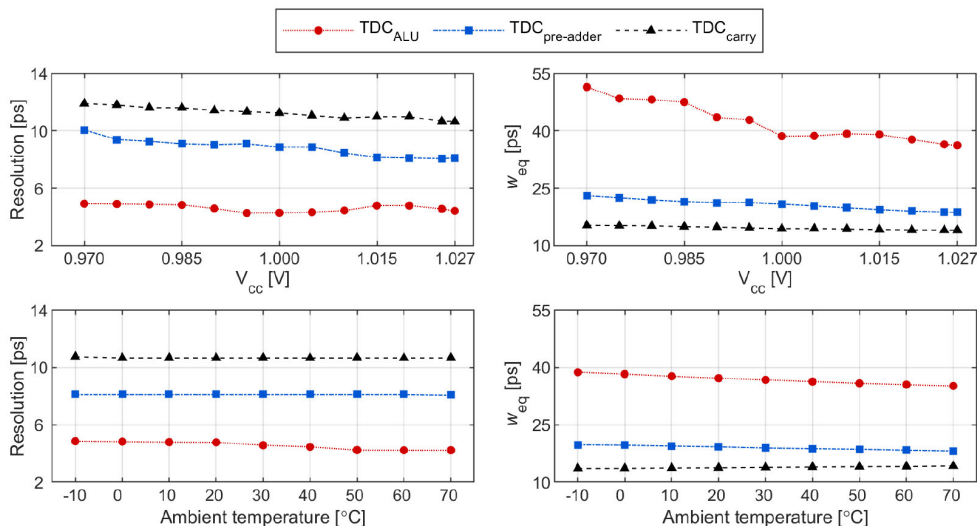

Fig. 7. The voltage and temperature influence on the mean and equivalent (w_{eq}) resolutions of TDC.

When V_{cc} increases, the mean and equivalent resolutions of the reference design (TDC$_{carry}$), improve by about 10% (from 11.9 ps to 10.7 ps and from 15.3 ps to 14.2 ps, respectively). However, the temperature change has a negligible influence on its resolutions. While DSP TDCs have higher resolutions, they are more vulnerable to both voltage and temperature variations. It is especially observed on w_{eq}. Through FPGA core voltage manipulation w_{eq} of TDC$_{ALU}$ can be improved by about 30% (from 52 ps to 37 ps) and in TDC$_{pre-adder}$ by about 20% (from 23 ps to 19 ps). Thus, it is recommended to use the possibly highest V_{cc} value (in the selected FPGA device the allowed core voltage is 1 V ± 3%). The temperature rise of 80°C improves w_{eq} of DSP TDCs only by 3 ps (TDC$_{ALU}$) and 1 ps (TDC$_{pre-adder}$).

5.5. Measurement uncertainty

Measurement uncertainty results from systematic and random errors [46]. The systematic errors represent the discrepancy between the measurement result and the true value. It is mainly caused by MTC 108 input circuit offsets and integrated time interval counter measurement signals' paths. This is not strictly related to TDC implementation by itself, so it was not tested. The random error influences counter precision defined as closeness of the agreement between the measured quantity values obtained by repeated measurements. Assuming the proper board design, *i.e.* low-noise power supply and input circuits, in modern interpolating time counters precision is mainly limited by the TDCs used as interpolators.

Figure 8 presents the obtained precision for three TDCs implemented in the integrated time counter. The precision was calculated as a standard deviation of 1000 samples of time interval measurements in a range from 100 ns (TIG 101 minimum operation range) up to 200 ms. Due to a limited stability of the counter reference clock (10 MHz, TCXO 0.5 ppm) for time intervals longer than 10 ms the precision deteriorates. The best precision, below 10 ps in a range of up to 10 ms, is obtained in the reference design with a carry chain-based TCL. The use of either TDC_{ALU} or $TDC_{pre-adder}$ results in 26 ps and 15 ps precision, respectively. The obtained metrology parameters for both measurement channels (CH A and CH B) are summarized in Table 1, including both *differential* (DNL) and *integral* (INL) *nonlinearities*.
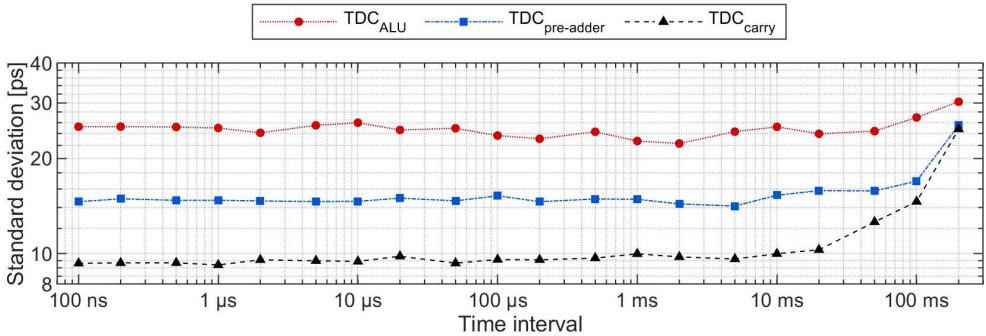

Fig. 8. Precision of the TDCs based on the DSP's ALU, the DSP's pre-adder and the carry chain.

Table 1. Measurement performance of the designed TDCs.

	CH	LSB [ps]	w_{eq}^* [ps]	σ_{DNL} [LSB]	DNL [LSB]	INL [LSB]	σ_{TDC}^{**} [ps]
TDC_{ALU}	A	4.2	40.8	2.48	[−0.96; 20.93]	[−15.14; 31.54]	26
	B	4.8	37.2	2.25	[−0.97; 18.32]	[−26.42; 23.09]	
$TDC_{pre-adder}$	A	8.1	19.3	1.05	[−0.97; 3.08]	[−7.48; 7.42]	15
	B	8.4	19.3	11.03	[−0.97; 2.75]	[−6.26; 6.11]	
TDC_{carry}	A	11.1	15.2	0.51	[−0.96; 1,96]	[−3.28; 3.80]	10
	B	10.7	13.8	0.46	[−0.91; 1.77]	[−2.75; 2.70]	

$^*w_{eq}$ – equivalent resolution,

$^{**}\sigma_{TDC}$ – measurement precision calculated as a standard deviation of measurement samples.

6. Conclusions

A novel implementation of TDC in an FPGA device is presented. The proposed solution employs DSP blocks and has advantages of high resolution (up to 8.1 ps and 4.2 ps, depending on DSP slice configuration) and programmable logic resource saving. The DSP TDC implemented in one FPGA clock region has about twice as wide operation range than a comparable carry chain-based time coding line. Furthermore, DSP configuration is facilitated by software tools making design implementation convenient.

The main drawback of the DSP TDC is its very high conversion nonlinearity. The existence of bins even 20 times larger than the LSB value leads to significant precision deterioration. Paradoxically, the better precision is obtained in a solution with the lower resolution. The DSP configuration with the use of a pre-adder gives 8.1 ps resolution and 15 ps precision while the other configuration, with the use of ALU, gives 4.2 ps and 26 ps, respectively.

The proposed converter can be used as an alternative to easily implemented in FPGA device solutions based on a multiphase clock. While each FPGA manufacturer has their own, unique DSP block architecture, it would also be interesting to examine them in terms of TDC performance. The further work will focus on the search of methods that can take advantage of existence of many ultra-narrow bins on the TDC transfer function.

Acknowledgements

References

[1] Longfei, K., Zhao, L., Zhou, J., Liu, S., Qi, A. (2013). A 128-Channel High Precision Time Measurement Module. *Metrol. Meas. Syst.*, 20(2), 275–286.

[2] Arkani, M. (2015). A High Performance Digital Time Interval Spectrometer: An Embedded, FPGA-Based System With Reduced Dead Time Behaviour. *Metrol. Meas. Syst.*, 22(4), 601–609.

[3] Kurtti, S., Nissinen, J., Jansson, J.-P., Kostamovaara, J. (2018). A CMOS chip set for accurate pulsed time-of-flight laser range finding. *Proc. IEEE I2MTC 2018*, Houston, Texas, United States.

[4] Grzelak, S., Czoków, J. Kowalski, M., Zieliński, M. (2014). Ultrasonic Flow Measurement with High Resolution. *Metrol. Meas. Syst.*, 21(2), 305–316.

[5] Won, J.Y., Lee, J.S. (2018). Highly Integrated FPGA-Only Signal Digitization Method Using Single-Ended Memory Interface Input Receivers for Time-of-Flight PET Detectors. *IEEE Trans. Biomed. Circuits Syst.*, 12(6), 1401–1409.

[6] Krehlik, P., Śliwczyński, L., Buczek, L., Kołodziej, J., Lipiński, M. (2016). ELSTAB – Fiber-optic time and frequency distribution technology: A general characterization and fundamental limits. *IEEE Trans. Ultrason., Ferroelectr., Freq. Control.*, 63(7), 993–1004.

[7] Fries, M.D., Williams, J.J. (2002). High-Precision TDC in an FPGA using a 192-MHz Quadrature Clock. *Proc. IEEE NSS/MIC 2002*, Norfolk, Virginia, United States, 580–584.

[8] Won, J.Y., Kwon, S.I., Yoon, H.S., Ko, G.B., Son, J.-W. Lee, J.S. (2016). Dual-Phase Tapped-Delay-Line Time-to-Digital Converter With On-the-Fly Calibration Implemented in 40 nm FPGA. *IEEE Trans. Biomed. Circuits Syst.*, 10(1), 231–242.

[9] Grzęda, G., Szplet, R. (2016). Time interval measurement module implemented in SoC FPGA device. *International Journal of Electronics and Telecommunications*, 62(3), 237–246.

[10] Chaberski, D. (2016). Time-to-digital-converter based on multiple-tapped-delay-line. *Measurement*, 89, 87–96.

[11] Won, J.Y., Lee, J.S. (2016). Time-to-Digital Converter Using a Tuned-Delay Line Evaluated in 28-, 40-, and 45-nm FPGAs. *IEEE Trans. Instrum. Meas.*, 65(7), 1678–1689.

[12] Szplet, R., Sondej, D., Grzęda, G. (2016). High-Precision Time Digitizer Based on Multiedge Coding in Independent Coding Lines. *IEEE Trans. Instrum. Meas.*, 65(8), 1884–1894.

[13] Szplet, R., Kwiatkowski, P., Jachna, Z., Różyc, K. (2016). An Eight-Channel 4.5-ps Precision Timestamps-Based Time Interval Counter in FPGA Chip. *IEEE Trans. Instrum. Meas.*, 65(9), 2088–2100.

[14] Chen, P., Hsiao, Y.-Y., Chung, Y.-S., Tsai, W.X., Lin, J.-M. (2017). A 2.5-ps Bin Size and 6.7-ps Resolution FPGA Time-to-Digital Converter Based on Delay Wrapping and Averaging. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 25(1), 114–124.

[15] Cui, K., Ren, Z., Li, X., Liu, Z., Zhu, R. (2017). A High-Linearity, Ring-Oscillator-Based, Vernier Time-to-Digital Converter Utilizing Carry Chains in FPGAs. *IEEE Trans. Nucl. Sci.*, 64(1), 697–704.

[16] Chaberski, D., Frankowski, R., Gurski, M., Zieliński, M. (2017). Comparison of Interpolators Used for Time-Interval Measurement Systems Based on Multiple-Tapped Delay Line. *Metrol. Meas. Syst.*, 24(2), 401–412.

[17] Wang, Y., Kuang, J., Liu, C., Cao, Q. (2017). A 3.9-ps RMS Precision Time-to-Digital Converter Using Ones-Counter Encoding Scheme in a Kintex-7 FPGA. *IEEE Trans. Nucl. Sci.*, 64(10), 2713–2718.

[18] Szplet, R., Kwiatkowski, P., Różyc, K., Jachna, Z., Sondej, T. (2017). Picosecond-precision multichannel autonomous time and frequency counter. *Rev. Sci. Instrum.*, 88(12), 125101/1–12.

[19] Pałka. M., *et al.* (2017). Multichannel FPGA based MVT system for high precision time (20 ps RMS) and charge measurement. *J. Instrum.*, 12(8), P08001/1–10.

[20] Sano, Y., Horii, Y., Ikeno, M., Sasaki, O., Tomoto, M., Uchida, T. (2017). Subnanosecond time-to-digital converter implemented in a Kintex-7 FPGA. *Nucl. Instrum. Methods Phys. Res. A*, 874, 50–56.

[21] Zhang, J., Zhou, D. (2018). An 8.5-ps Two-Stage Vernier Delay-Line Loop Shrinking Time-to-Digital Converter in 130-nm Flash FPGA. *IEEE Trans. Instrum. Meas.*, 67(2), 406–414.

[22] Chen, H., Li, D.-U. (2019). Multichannel, Low Nonlinearity Time-to-Digital Converters Based on 20 and 28 nm FPGAs. *IEEE Trans. Ind. Electron.*, 66(4), 3265–3274.

[23] Cao, G., Xia, H., Dong, N. (2019). A 6.6 ps RMS resolution time-to-digital converter using interleaved sampling method in a 28 nm FPGA. *Rev. Sci. Instrum.*, 90(4), 044706/1–8.

[24] Lusardi, N., Garzetti, F., Geraci, A. (2019). Digital instrument with configurable hardware and firmware for multi-channel time measures. *Rev. Sci. Instrum.*, 90(5), 055113/1–13.

[25] Kwiatkowski, P., Szplet, R. (2019). Time-to-Digital Converter with Pseudo-Segmented Delay Line. *Proc. IEEE I2MTC 2019*, Auckland, New Zealand.

[26] Carbone, P., Kiaei, S., Xu, F. (2014). *Design, Modeling and Testing of Data Converters*. Springer Berlin Heidelberg.

[27] Wu, J., Shi, Z., Wang, I.Y. (2003). Firmware-only implementation of Time-to-Digital Converter (TDC) in field-programmable gate array (FPGA). *IEEE Nucl. Sci. Symp. Conf. Rec. 2003*, Portland, Oregon, United States, 177–181.

[28] Wu, J., Shi, Z. (2008). The 10-ps wave union TDC: Improving FPGA TDC resolution beyond its cell delay. *Proc. IEEE NSS/MIC 2008*, Dresden, Germany, 3440–3446.

[29] Aloisio, A., Branchini, P., Cicalese, R., Giordano, R., Izzo, V., Loffredo, S. (2007). FPGA implementation of a high-resolution time-to-digital converter. *Proc. IEEE NSS/MIC 2007*, Honolulu, Hawaii, United States, 504–507.

[30] Wang, H., Zhang, M., Yao, Q. (2013). A new realization of time-to-digital converters based on FPGA internal routing resources. *IEEE Trans. Ultrason., Ferroelectr., Freq. Control.*, 60(9), 17887–1795.

[31] Szplet, R., Klepacki, K. (2010). An FPGA-integrated time-to-digital converter based on two-stage pulse shrinking. *IEEE Trans. Instrum. Meas.*, 59(6), 1663–1670.

[32] Wang, Y., Liu, C., Zhu, W. (2013). Two novel designs of multi-phase clocked ultra-high speed time counter on FPGA for TDC implementation. *Proc. of IEEE NSS/MIC 2013*, Seoul, South Korea, 1082–3654.

[33] Bogdan, M., *et al.* (2005). A 96-channel FPGA-based Time-to-Digital Converter (TDC) and fast trigger processor module with multi-hit capability and pipeline. *Nucl. Instrum. Methods Phys. Res. A*, 554, 444–457.

[34] Arpin, L., Bergeron, M., Tétrault, M.-A., Lecomte, R., Fontaine, R. (2009). A Sub-Nanosecond Edge Detection System using embedded FPGA fabrics. *Proc. IEEE-NPSS Real Time Conf. 2009*, Beijing, China, 299–303.

[35] Imrek, J., Hegyesi, G., Kalinka, G., Molnár, J., Nagy, F., Valastyan, I., Szabó, Z. (2010). FPGA based TDC using Virtex-4 ISERDES blocks. *Proc. NSS/MIC 2010*, Knoxville, Tennessee, United States, 1413–1415.

[36] Xiang, T., Zhao, L., Jin, X., Wang, T., Chu, S., Ma, C., Liu, S., An, Q. (2014). A 56-ps multi-phase clock time-to-digital convertor based on Artix-7 FPGA. *Proc. IEEE-NPSS Real Time Conf. 2014*, Nara, Japan.

[37] Tancock, S., Arabul, E., Dahnoun, N., Mehmood, S. (2018). Can DSP48A1 adders be used for high-resolution delay generation? *Proc. MECO 2018*, Budva, Montenegro.

[38] Tancock, S., Dahnoun, N. (2019). A 5.25 ps-resolution TDC on FPGA using DSP blocks. *Proc. DISP 2019*, Oxford, United Kingdom.

[39] Salomon, R., Joost, R. (2009). BOUNCE: A New High-Resolution Time-Interval Measurement Architecture. *IEEE Embedded Syst. Lett.*, 1(2), 56–59.

[40] 7 Serie DSP48E1 Slice. User Guide UG479, v1.10, Xilinx. *https://www.xilinx.com/support/documentation/user_guides/ug479_7Series_DSP48E1.pdf* (Mar. 2018)

[41] Cova, S., Bertolaccini, M. (1970). Differential linearity testing and precision calibration of multichannel time sorters. *Nucl. Instrum. Methods*, 77(2), 269–276.

[42] 7 Series FPGAs Configurable Logic Block. User Guide UG474, v1.8, Xilinx. *https://www.xilinx.com/support/documentation/user_guides/ug474_7Series_CLB.pdf* (Sep. 2016)

[43] Kwiatkowski, P., Różyc, K., Sawicki, M., Jachna, Z., Szplet, R. (2017). 5 ps jitter programmable time interval/frequency generator. *Metrol. Meas. Syst.*, 24(1), 57–68.

[44] Szplet, R., Jachna, Z., Kwiatkowski, P., Różyc, K. (2013). A 2.9 ps equivalent resolution interpolating time counter based on multiple independent coding lines. *Meas. Sci. Technol.*, 24(7), 035904/1–15.

[45] Wu, J. (2014). Uneven Bin Width Digitization and a Timing Calibration Method Using Cascaded PLL. *Proc. IEEE-NPSS Real Time Conf. 2014*, Nara, Japan.

[46] Szplet, R., Szymanowski, R., Sondej, D. (2019). Measurement Uncertainty of Precise Interpolating Time Counters. *IEEE Trans. Instrum. Meas.*, 68(11), 4348–4356.