A 17ps Time-to-Digital Converter Implemented in 65nm FPGA Technology

Claudio Favi and Edoardo Charbon
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
{claudio.favi, edoardo.charbon}@epfl.ch

ABSTRACT

This paper presents a new architecture for time-to-digital conversion enabling a time resolution of 17ps over a range of 50ns with a conversion rate of 20MS/s. The proposed architecture, implemented in a 65nm FPGA system, consists of a pipelined interpolating time-to-digital converter (TDC). The TDC comprises a coarse time discriminator and a fine delay line, capable of sustained operation at a clock frequency of 300MHz. A Turbo version of the circuit implements a pipelined interpolating TDC with suppressed dead time to reach a conversion rate of 300MS/s at the expense of a systematic asymmetry that requires fast error correction. The TDCs proposed in this paper can be compensated for process, voltage, and temperature (PVT) variations using a conventional charge pump based feedback or a digital calibration technique. Results demonstrate the suitability of the approach for a variety of applications involving high-precision ultra-fast time discrimination, such as optical lifetime sensing, time-of-flight cameras, high throughput comlinks, RADARs, etc.

Categories and Subject Descriptors

B.m [**Hardware**]: Miscellaneous

General Terms

Measurement, Experimentation

Keywords

TDC, time-to-digital converters, time-correlated instrumentation, deep sub-nanosecond time resolution, optical communications, ultra-fast digital electronics, high-speed readout, 65nm FPGA

1. INTRODUCTION

Detecting the precise relative time occurrence of an event is critical in many applications where the time difference between two electrical pulses needs to be measured with picosecond precision. Examples of such applications are time-of-flight sensors, optical correlation spectroscopes, positron emission tomography instruments, and pulse position demodulators, just to name a few [1, 2, 3]. Besides high time resolution, typically 100ps or less, it is often useful to achieve high throughput to optimize system speed and detection efficiency. For instance, in metrological applications, high throughput has the effect of achieving the wanted accuracy in a shorter time, thus improving the speed of a measurement or making it possible to achieve unprecedented accuracies whenever a phenomenon is fast occurring. Similar advantages could be achieved in RADARs, high-energy physics, and time-resolved imaging [4, 5].

There exist several techniques to measure a time difference between two subsequent digital signals. The simplest solution is to use a counter that can be started and stopped by a reset signal. In principle, the clock frequency of the counter determines the resolution. Thus, if metastability effects are ignored, to achieve 100ps of resolution, a 10GHz clock is required. However, due to limitations of bandwidth, this solution is not feasible in most commercially available FPGAs. An alternative is the use of time-to-amplitude converters (TACs). The principle of operation of a TAC is generally based on a controlled discharge of a capacitance, whereby the time difference between a start and stop signal is inferred from the voltage difference before and after the completion of the discharge [6]. The problem with this approach is that it requires high-speed analog switches, high-precision current sources, and high-resolution A/D converters. These components are not available in conventional FPGAs.

Another alternative is the use of a time-to-digital converter (TDC). TDCs are generally digital components in nature and, except for the calibration circuitry, they can be implemented in a fully-digital design style. Thus, TDCs are good candidates for implementation in a commercial FPGA. Indeed the literature of FPGA based TDCs is quite extensive [7, 8, 9]. While these designs achieved sub-nanosecond resolution, throughput was usually limited.

High time resolution and high throughput are often contradictory specifications, however by proper use of design techniques and heavy use of pipelining, it is generally possible to achieve a good compromise. In the design proposed in [10] for example, a 3-stage pipelined TDC was used to achieve 10MS/s with a resolution of 97ps. However, the TDC was designed in full-custom style and it occupied a surface of approximately 0.4mm^2. In addition, the design was highly optimized so as to achieve PVT variability con-

FPGA'09, February 22–24, 2009, Monterey, California, USA.

trol via a feedback loop comprising a charge pump, a loop filter, and a digital comparator. A design of this kind is not feasible for implementation in a commercial FPGA.

In this paper, we propose a new design based on a pipelined interpolating time-to-digital conversion architecture implemented in a commercial 65nm FPGA. The TDC achieves an overall resolution of 17ps. The component is relatively small, requiring only 1208 Slices and it achieves a throughput of 20MS/s in normal operating mode and 300MS/s in Turbo mode. The TDC takes advantage of a number of design techniques to best utilize the FPGA technology while maintaining the utilization efficiency above a minimum threshold. In particular, the repetitive fabric of CLBs is utilized in this design so as to minimize clock net distribution mismatches and to limit the skew across the TDC. In addition, two calibration techniques are proposed for countering PVT variations.

The application that inspired this design involves the use of single-photon avalanche diodes (SPADs [11]) to ensure chip-to-chip and intra-chip optical communication. While SPADs are photodetectors with exceptionally high time resolution, they lack speed and require a relatively high detection cycle or dead time. To ensure high throughput nonetheless, we have proposed to use pulse position modulation (PPM) applied to the optical pulses [12]. With this scheme it is necessary to implement high throughput time difference discriminators that can be implemented in a very small surface. The proposed TDC attempts to address these issues in a preliminary FPGA implementation.

The PVT variability control is achievable in two ways. The first is based on the use of power supply control feedback that can be achieved with a conventional analog loop. Such an implementation may be achieved with discrete components. The second mechanism, actually implemented in this design, is based on a digital scheme. In the scheme, the range of the TDC is periodically measured and the underlying fine delay line is identified. In case of temperature variations, each delay element will get faster or slower, thus the same range will require a lower or higher number of delay elements. To maintain the same resolution, the raw codes are mapped to the correct time difference using an appropriate table whereby inter-value interpolations may be used. The same technique can be used when migrating the TDC configuration from one FPGA to another and from one power supply to another, so as to achieve PVT independence.

The paper is organized as follows. First the two architectures are presented in Section 2. After a discussion of non-idealities, we present the two digital calibration techniques in Section 3. The results are presented in Section 4; the section comprises a description of our setup and a comparison of normal and Turbo modes. A discussion of the results is presented in Section 5.

2. ARCHITECTURE

2.1 General Considerations

Traditional TDC architectures use different interpolation methods for fine and coarse time measurement. A counter based technique is usually used for coarse measurements, generally achieving nanosecond resolution. For finer resolutions, several techniques exist. The Vernier method for instance uses two oscillators slightly out of tune [13, 14] or two tapped delay lines with slightly different delay [15, 16].

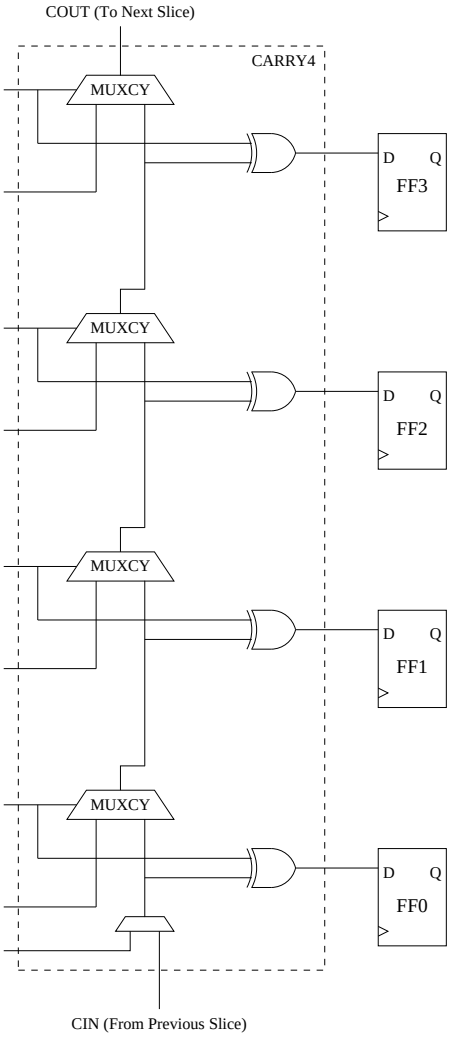

Figure 1: Xilinx Virtex-5 slice (partial) with carry logic and sequential logic

Methods using compensated [10, 17, 18] or non-compensated [9, 19, 20] tapped delay lines also exist.

Several of these techniques have been implemented in FPGA [7, 8, 9, 19, 20]. In this paper we focus on the tapped delay line method since we can take advantage of the dedicated carry lines present in all FPGA fabrics. As they are usually employed in adder logic, each adder's bit has a flip-flop very close to the line. Figure 1 shows the carry logic and sequential logic in Virtex-5 slices.

The overall architecture (Fig. 2, 3) consists of a free running coarse counter, a tapped delay line implemented in a carry chain, an input signal filter, encoder and reset logic, and readout logic. The principle of operation is the following. We split time in frames of 16 clock cycles of which one is reserved to reset the fine delay line and the filter. Therefore one measurement may be performed per frame. The state of the fine line is latched and a synchronous pulse is generated for each hit event. The active elements of the line are selected to lie in the middle of it. To minimize metastability effects, we use a three stage synchronizer. The thermometer encoded value in the line is converted to binary. Finally, the

coarse and fine values are input into the FIFO for external readout.

We implemented the TDC in VHDL. The design is automatically placed and routed with the exception of the delay chain which was constrained to be placed in desired locations. No manual routing was performed.

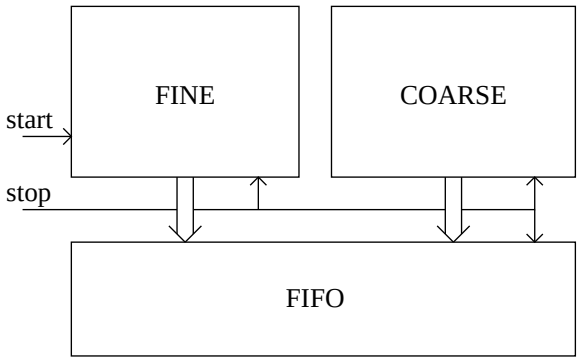

Figure 2: Architecture of the TDC

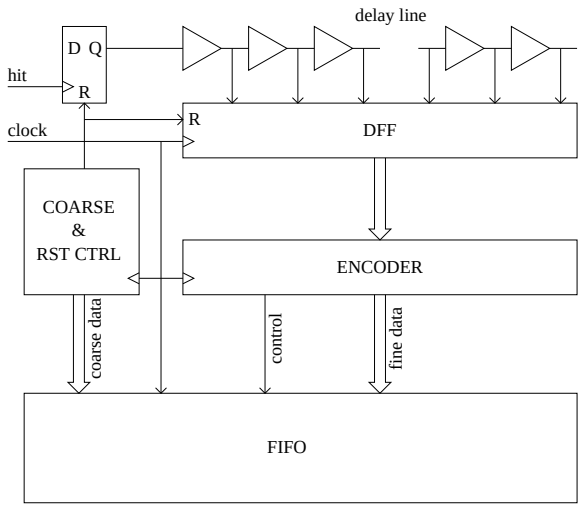

Figure 3: Normal mode architecture

2.2 Reducing Dead Time

The fine delay line operates by propagating the hit signal. Assuming a reset state of all ones, zeros propagate from the beginning toward the end of the line. Since the line is sampled at each clock cycle, the resulting pattern of *fine state* could be (bits flow from right to left):

coarse	fine line
0	11111111111111111111
1	11111111111111100000
2	11100000000000000000
3	00000000000000000000
...	...
14	00000000000000000000
15	11111111111111111111

At cycle 1 the measurement is armed while cycle 15 resets the line to start a new measurement. The dead time is defined as the minimum time between the end of a measurement and the start of the next one. In this case the dead time is one clock cycle, however, in principle, already at cycle 2, the beginning of the line is stable. We use a toggling filter to trigger a propagation of ones and zeros alternatively (Fig. 4). In this fashion the dead time is effectively reduced to the minimum toggling time of the input filter flip-flop (< 500ps).

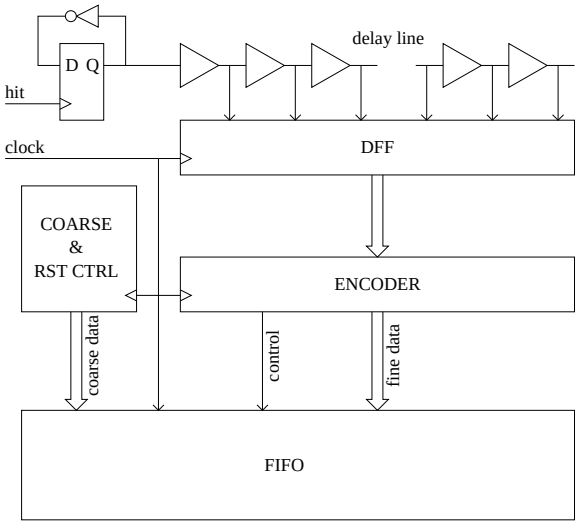

Figure 4: Turbo mode architecture

Another advantage of this architecture, referred to in the remainder of the paper as Turbo mode, is the increase of readout speed and the ability of handling multiple hits per frame. In this scenario the speed limiting design factor is the encoder (from thermometer to binary).

There are however some limitations. First, since the polarity of the travelling pulse can be high-to-low or low-to-high, an asymmetry in propagation delay will arise. Second, the likelihood of metastability in the system is higher. This is due to the fact that the flip-flops latching the delay chain are never reset and the input filter can toggle several times per cycle.

2.3 Bubbles and Bubble Suppression

For each measurement there may exist a flip-flop of the fine chain for which the setup or hold time is violated. The resulting value will eventually resolve into a one or zero. Even assuming an unskewed clock, this can result in *bubbles* in the code.

For the encoder to handle bubbles in the code, we implemented two techniques. The first is a basic bit counting method. We just count the number of propagation bits present in the line (assuming only one hit per cycle). If the number of bubble bits is small (one or two), then the difference in code will be small as well. The second technique, tries to find the position of the first propagation bit in the chain. Although more computationally intensive, the latter yields more realistic results. This is the technique used in the remainder of this paper.

3. DIGITAL CALIBRATION

3.1 Interpolation

Adjusting delays in the fine line to compensate PVT variations in FPGA is not trivial. In this section, we present how PVT compensation is performed through digital techniques.

We call interpolation the process by which we convert a code from its natural interval $(]0, N]$, with $N < N_{max}$ where N_{max} is the total number of elements in the chain) to an output interval $(]0, 2^b])$. As first approach, this can be done linearly: $c' = c * 2^b/N$.

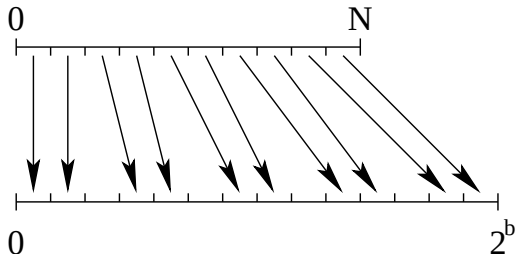

Figure 5: Interpolation principle

The implementation can be easily performed through mapping. The map information can reside in a block RAM and a simple look-up in the RAM yields the interpolation result. The RAM content can be computed on-the-fly every time N changes. We must note, as portrayed in Figure 5, that some codes may never appear in the output of the mapping (if $2^b > N$).

3.2 Automatic Range Adjustment (ARA)

The delay line has to be designed to cover at least the propagation of 1 clock cycle. Let $N(x)$ be the number of propagated bits in the fine delay line registered at cycle x. Measurement $N(x)$ is valid if $N(x) > 0$ and $N(x-1) = 0$. Measurement $N(x+1)$ is a range measurement if $N(x)$ is valid and $N(x+1) \neq N_{max}$. We compute the incremental resolution r or LSB by

$$r = \frac{T}{N(x+1) - N(x)}$$

where T is the clock period. In the example of table 1, $x+1$ is a range measurement and therefore $r = T/10$.

Table 1: Automatic Range Adjustment Example

coarse	fine line	$N(x)$
$x-1$	1111111111111111111	0
x	1111111111111000000	6
$x+1$	1111000000000000000	16
	|---------|	

This method has the advantage of being capable of adjusting range on-the-fly while measurements are performed. One can keep a running mean of the range value and therefore adapt the interpolation process. However, this method can only adapt to relatively slow changing variations in PVT (in the order of a few clock cycles).

3.3 Downsampling

We expect the delay of inter-slice routing to be relatively large compared to intra-slice routing. This will induce non-linearities that can be mitigated by implementing only 1 or 2 taps per slice. Note that, although we expect an improved linearity, the resolution will also be impacted negatively.

We decided to use the maximum resolution the fine line can offer by implementing 4 taps per slice (see Fig. 1). However, we present some results of the downsampling technique in section 4.

3.4 Pseudorandom Bin Dithering

Given the result of a statistical code density test[1] one has an approximation on the real bin time distribution [21]. We can improve the interpolation with this knowledge. If we interpolate to a higher dimension output code (i.e. $2^b > N$), some source bin will be mapped to more than one destination bin. We can use a pseudorandom *dithering* to distribute the original bin value to the corresponding output bins. For example, in Figure 6, given the statistical code density test result, we can build an intermediate representation of the binning interval. This interval is the real-value time interval of one clock cycle where each bin has it's *size* set according the density test. We can then quantize this interval in 2^b slots. A given bin will then map to the corresponding slots to the extent of the bin coverage. For example the last bin in Figure 6 will map to output codes $2^b - 1$ and $2^b - 2$ with likelihood of 40% each and to $2^b - 3$ with likelihood of 20%. This non-deterministic mapping can be implemented with a pseudo random number generator or a true random number generator.

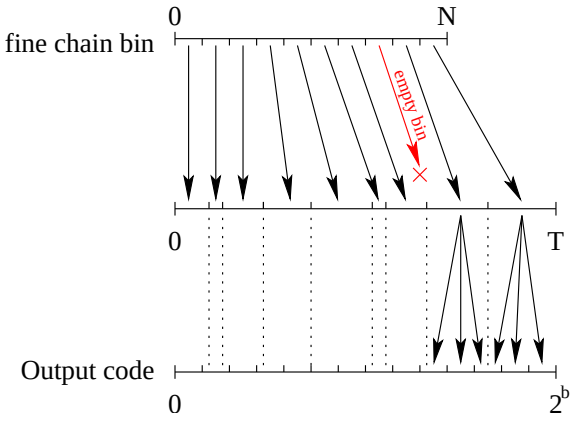

Figure 6: Pseudorandom bin dithering based on statistical code density test results.

4. RESULTS

4.1 Test Setup

The experiments were conducted with the Xilinx ML505 board. The on-board Virtex-5 device is fabricated in 1V, 65nm triple-oxide process. The hit signal is fed to the FPGA through one of the clock input SMA connectors. Measurement data is transmitted from the FIFO to the PC through a USB interface. All reference measurements, such as input

[1]Also see section 4.1.

clock jitter, cable time delay and time distribution of the random input signal, were performed on LeCroy WaveMaster 8600A digitising oscilloscope.

The *stop* signal of our design is generated on board by a low jitter frequency synthesiser. Its differential output running at 300MHz shows a measured jitter of less than 12 picoseconds after going through the FPGA fabric. The coarse counter is free running at this frequency and shows an integral linearity error below 0.03 coarse LSB. We focus the remainder of the results and discussion on the fine interpolation technique.

We performed three kinds of measurements. They are described in the following paragraphs. The results will be presented in the next section and discussed in section 5.

The first experiment measures the linearity of the TDC. Although there are several ways to conduct such test, the most common method is a statistical one called 'statistical code density test' [7]. The idea is to generate a large number N of pulses randomly distributed in time, and collect the result of the TDC interpolation into an histogram. In the ideal case the histogram has C code bins containing $\overline{n} = N/C$ each. In reality there is a differential non-linearity $\mathrm{DNL_c}$ and for each code $c \in [1, C]$, $n_c \neq N/C$. n_c representing the depth of bin c.

$$\mathrm{DNL_c} = \frac{\mathrm{n_c}}{\overline{\mathrm{n}}} - 1.$$

The cumulative sum of $\mathrm{DNL_c}$ yields the integral non-linearity $\mathrm{INL_c}$

$$\mathrm{INL_c} = \sum_{\mathrm{i}=1}^{\mathrm{c}} \mathrm{DNL_i}.$$

INL and DNL values refer to the 1-bin unit hence are expressed in LSB units.

The second measurement is related to the ARA. While we run a statistical code density test we collect the ARA measurements in a histogram. We use this measurement to compute the mean bin width of the fine codes.

Finally the last test yields the standard uncertainty or random error. We inject a delayed version of a clock synchronous signal into the fine delay line. We used several fixed length cables to generate several delays. The worst standard deviation of the resulting codes is reported.

4.2 Measurements

To perform the statistical code density test, we generated random hit pulses with a photo-detector exposed to ambient light. The sensor based on single photon avalanche diode (SPAD) [11] displays a uniform distribution of pulses separated by at least 30 nanoseconds. Figure 7 shows the linearity of the channel. The obtained DNL varies between -1 and $+3.55$. The INL lies in $[-3, +2.58]$ range.

The same test has been applied to the same fine chain with downsampling by 4. The result is shown in Figure 8. The INL's range is $[-0.49, +1.18]$.

Figure 9 shows the histogram of the automatic range adjustment measurement. At 20°C, the full range is achieved with 207 elements. Therefore the bin width is 16.1 picoseconds. The standard deviation of the histogram is of about 2 bins.

The standard uncertainty can be measured with a fixed cable delay. The worst-case standard deviation is reported in Table 2. For the example given in Figure 10 its value is 20.46ps.

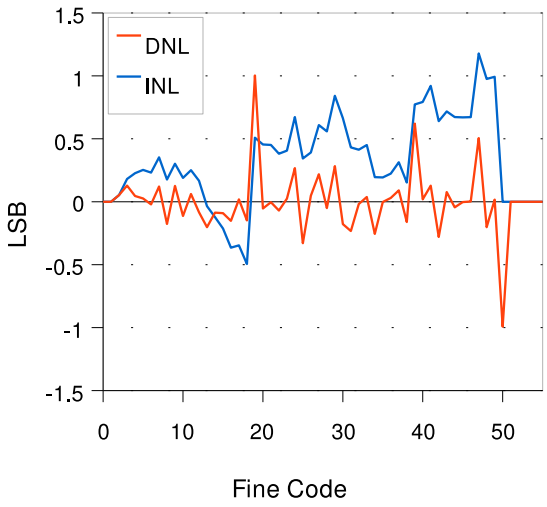

Figure 8: DNL and INL after downsampling by 4 (1 tap per slice)

The results of the Turbo mode are similar to the *normal* mode. The mismatch between the propagation of zeros and ones is of $2 - 3$ bins. We were not able to see metastability errors in the codes.

Table 2 summarizes the characteristics of this work.

5. DISCUSSION

The non-linearity shown in Figure 7 needs some explanation. The large variations in DNL around bin 76 are due to the clock distribution between slices. The clock signal travels from the pad to a global clock buffer (BUFG) and is then distributed to each slice. The delay from the BUFG to some slice is reported in Table 3. Note that the difference between slice X28Y39 and X28Y40 is 57ps (= 3.3 LSB).

Table 3: Clock distribution delays for fine TDC line

Slice	Delay	fine position
X28Y21	1.614ns	3
X28Y22	1.612ns	7
X28Y23	1.609ns	11
	...	
X28Y37	1.613ns	67
X28Y38	1.615ns	71
X28Y39	**1.617ns**	75
X28Y40	**1.560ns**	79
X28Y41	1.558ns	83
X28Y42	1.556ns	87
X28Y43	1.553ns	91
X28Y44	1.549ns	95
X28Y45	1.544ns	99

There is also another non-linearity that lies internally in all the slices. The result of our statistical code density tests show that inside each slice there is a disparity between the four latches (see Fig. 1). The problem is similar to the global clock distribution. The distribution of values varies from slice to slice. However the second latch in every slice never

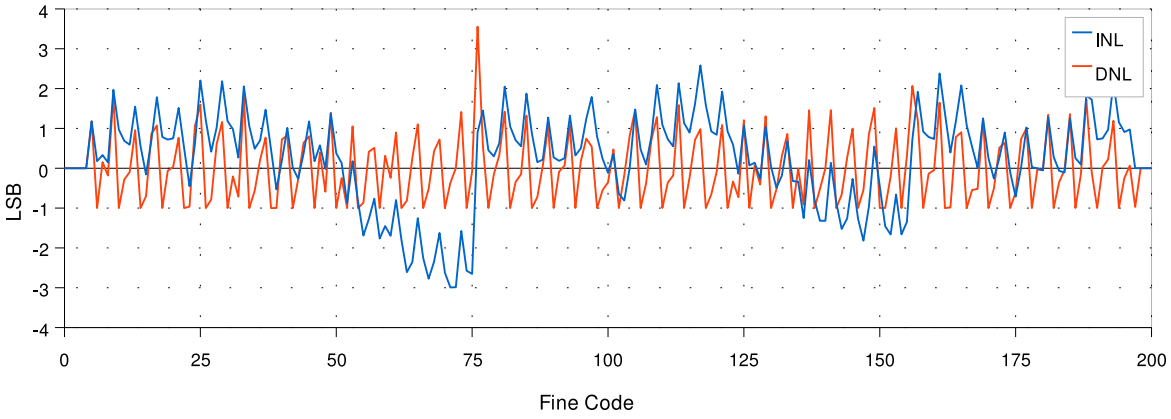

Figure 7: Performance of the fine measurement. Differential and Integral Nonlinearity

Table 2: Performance summary

	Min	Typ	Max	Unit
Clock frequency		300		MHz
Standard uncertainty	9.8		24.2	ps
Resolution		16.9		ps
DNL	−1		3.55	LSB
INL	−2.99		2.58	LSB
Normal Mode				
Measurement Range (MR)		50		ns
Dead Time (DT)	3.33		50	ns
Readout speed		20		MSample/s
Turbo Mode				
Measurement Range (MR)		53.33		ns
Dead Time (DT)	0.5		3.33	ns
Readout speed		300		MSample/s

Table 4: Carry4 structure delay from CIN to FF (from simulation)

Point	Delay [ps]	Diff [ps]
CIN	0	0
FF0_D	33	33
FF1_D	47	**14**
FF2_D	81	34
FF3_D	104	23

becomes the most significative propagation bit. This systematic error can be explained by the combination of two factors. First, the propagation delay inside the carry4 structure is not uniform. The simulation results shown in table 4 display a delay for bin 2 that is substantially smaller. Second, the delay in clock distribution between the slice's latches is not identical. This is similar to the inter-slice clock distribution problem of the previous paragraph. It is also to be noted that theses phenomena contribute to the creation of bubbles.

We tried different placements of the delay chain. This non-linearity was found in all the designs. Figure 11 shows one of the results referred to another placed configuration. With further inspection we noticed that the anomaly arises in specific places of the device where the slices spacing is slightly different to allow more routing resources. This happens regularly and the maximum clock delay difference occurs every 40 slices. To be able to diminish the effect one would need to shorten the fine line and place it in such a way to avoid the particularly bad spots. However when reducing the fine line, one has to increase the clock frequency in order to always have a complete clock cycle covered.

Downsampling techniques can be used to obtain better INL. This comes at a cost of lower resolution. With a downsampling factor of 4, we achieve results (Fig. 8) that are a little better than [9]. For about the same resolution, the INL of our system [-0.49,+1.18] is narrower [-2.003,+1.855]. Note however that the device used in [9] (Virtex II) is fabricated in a 130nm, 1.5V CMOS process. The clock distribution problem was also reported in that work.

A measurement campaign for the two implementations of the TDC (normal and Turbo mode) under PVT variability regime is currently underway. In simulations it was concluded that the minimum resolution at 95% of power supply and 85°C was 26ps. It is conceivable that the degradation of this value with respect to the measured value of 17ps can be attributed entirely to variability and thus no important temperature/power supply induced degradation ought to be expected.

To improve the design further, we are planning to implement several fine delay chains to mitigate the problem of

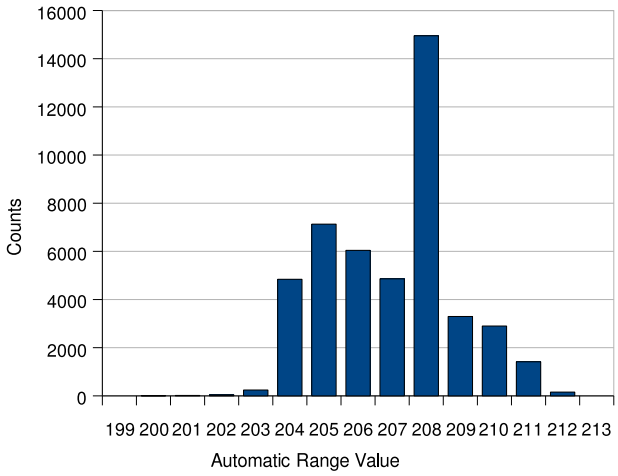

Figure 9: Histogram of the automatic range adjustment method

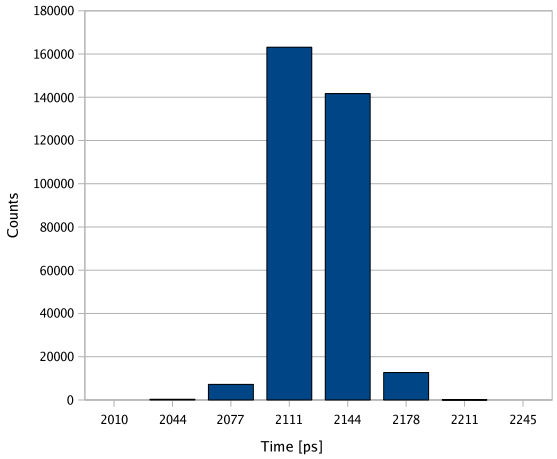

Figure 10: Time delay measurement, std dev: 20.46ps, binning after downsampling by 2

non-linearity described in this section. However, to be able to *shift* the delay lines, one must take extra care of routing the hit signal to all the chains.

6. CONCLUSIONS

A new TDC family is presented enabling a deep sub-nanosecond time resolution over a 50ns range and a conversion rate up to 300MS/s in Turbo mode. The TDC was implemented in a 65nm FPGA with efficient area utilization (1208 Slices). To the best of our knowledge, the proposed TDC has the highest reported resolution for an FPGA implementation. In addition, the conversion rate is comparable with state-of-the-art ASICs reported in the literature. A fully FPGA compatible compensation mechanism was also proposed for the TDC to account for PVT variability. Results show that the approach is suitable for many applications where high-precision ultra-fast time discrimination is critical. Due to the reduced size of the TDC and its excellent PVT variation rejection it could be used in large

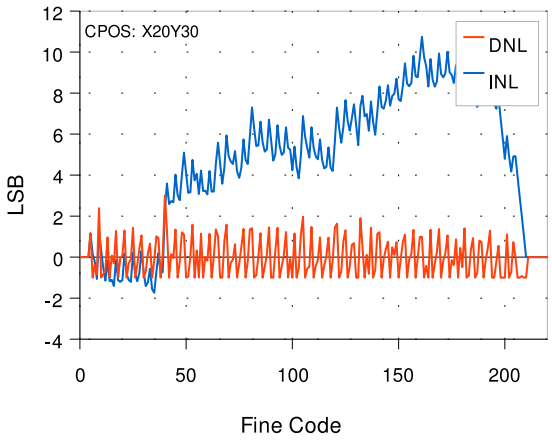

Figure 11: Alternate fine chain position (X20Y30). Differential and Integral Nonlinearity

arrays, useful in applications including optical lifetime evaluation, time-of-flight cameras, high-speed communication, RADARs, etc.

7. ACKNOWLEDGEMENTS

The authors would like to thank T.H.J. Kluter, P. Brisk, and C. Schneeberger for guidance and support during this work. This research was supported by the Mobile Information and Communication System (MICS) [22], the Swiss National Fund (SNF), and Xilinx University Program [23].

8. REFERENCES

[1] B.K. Swann, B.J. Blalock, L.G. Clonts, D.M. Binkley, J.M. Rochelle, E. Breeding, and K.M. Baldwin. A 100-ps time-resolution cmos time-to-digital converter for positron emission tomography imaging applications. *Solid-State Circuits, IEEE Journal of*, 39(11):1839–1852, Nov. 2004.

[2] K. Karadamoglou, N.P. Paschalidis, E. Sarris, N. Stamatopoulos, G. Kottaras, and V. Paschalidis. An 11-bit high-resolution and adjustable-range cmos time-to-digital converter for space science instruments. *Solid-State Circuits, IEEE Journal of*, 39(1):214–222, Jan. 2004.

[3] H. Matsumoto, O. Sasaki, K. Anraku, and M. Nozaki. Low power high resolution tdc with fast data conversion for balloon-borne experiments. *Nuclear Science, IEEE Transactions on*, 43(4):2195–2198, Aug 1996.

[4] K. Maatta and J. Kostamovaara. A high-precision time-to-digital converter for pulsed time-of-flight laser radar applications. *Instrumentation and Measurement, IEEE Transactions on*, 47(2):521–536, Apr 1998.

[5] F. Bigongiari, R. Roncella, R. Saletti, and P. Terreni. A 250-ps time-resolution cmos multihit time-to-digital converter for nuclear physics experiments. *Nuclear Science, IEEE Transactions on*, 46(2):73–77, Apr 1999.

[6] W. M. Henebry and A. Rasiel. Design features of a start-stop time-to-amplitude converter. *Nuclear*

Science, IEEE Transactions on, 13(2):64–68, April 1966.

[7] J. Kalisz, R. Szplet, J. Pasierbinski, and A. Poniecki. Field-programmable-gate-array-based time-to-digital converter with 200-ps resolution. *Instrumentation and Measurement, IEEE Transactions on*, 46(1):51–55, Feb 1997.

[8] R. Szplet, J. Kalisz, and R. Szymanowski. Interpolating time counter with 100 ps resolution on a single fpga device. *Instrumentation and Measurement, IEEE Transactions on*, 49(4):879–883, Aug 2000.

[9] Jian Song, Qi An, and Shubin Liu. A high-resolution time-to-digital converter implemented in field-programmable-gate-arrays. *Nuclear Science, IEEE Transactions on*, 53(1):236–241, Feb. 2006.

[10] C. Niclass, C. Favi, T. Kluter, M. Gersbach, and E. Charbon. A 128x128 single-photon imager with on-chip column-level 10b time-to-digital converter array capable of 97ps resolution. *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, pages 44–594, Feb. 2008.

[11] A. Rochas. *Single Photon Avalanche Diodes in CMOS Technology*. PhD thesis, EPF-Lausanne, Switzerland, 2003.

[12] C. Favi and E. Charbon. Techniques for fully integrated intra-/inter-chip optical communication. *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, pages 343–344, June 2008.

[13] Barton R D and King M E. Two vernier time-interval digitizers. *Nucl. Instrum. Methods*, 97(359-70), 1971.

[14] Barton R G. The vernier time-measuring technique. In *IRE*, pages 21–30, 1957.

[15] Hoppe D R. Time interpolator. *US Patent*, (4,439,046), 1982.

[16] Hoppe D R. Differential time interpolator. *US Patent*, (4,433,919), 1982.

[17] M.J. Loinaz and B.A. Wooley. A cmos multichannel ic for pulse timing measurements with 1-mv sensitivity. *Solid-State Circuits, IEEE Journal of*, 30(12):1339–1349, Dec 1995.

[18] Y. Arai and T. Ohsugi. Tmc-a cmos time to digital converter vlsi. *Nuclear Science, IEEE Transactions on*, 36(1):528–531, Feb 1989.

[19] M.S. Andaloussi, M. Boukadoum, and E.M. Aboulhamid. A novel time-to-digital converter with 150 ps time resolution and 2.5 ns pulse-pair resolution. *Microelectronics, The 14th International Conference on 2002 - ICM*, pages 123–126, Dec. 2002.

[20] Jinyuan Wu, Zonghan Shi, and I.Y. Wang. Firmware-only implementation of time-to-digital converter (tdc) in field-programmable gate array (fpga). *Nuclear Science Symposium Conference Record, 2003 IEEE*, 1:177–181 Vol.1, Oct. 2003.

[21] M. Mota and J. Christiansen. A high-resolution time interpolator based on a delay locked loop and an rc delay line. *Solid-State Circuits, IEEE Journal of*, 34(10):1360–1366, Oct 1999.

[22] http://www.mics.org/.

[23] http://www.xilinx.com/univ/.