An Asynchronous Pipelined Time-to-Digital Converter Using Time-Domain Subtraction

Omer Can Akgun

Section Bioelectronics, Department of Microelectronics, Delft University of Technology, the Netherlands
E-mail: o.c.akgun@tudelft.nl

Abstract—This paper presents the design of a low-power asynchronous pipelined time-to-digital converter (AP-TDC) to be employed in a time-domain signal processing system. The presented AP-TDC utilizes two novel concepts, namely time-domain subtraction and absolute value based algorithmic conversion. The design and simulation of the AP-TDC is done using a standard CMOS 65 nm process. The least-significant-bit resolution of the AP-TDC is designed to be 200 ps and the AP-TDC outputs 7-bit digital words with an ENOB of 6.2 bits. The dynamic range of the TDC is 25.4 ns and the TDC core consumes $38 \,\mu$W from a supply voltage of 1 V and has a total area of 1275 μm^2. When compared to a Flash TDC implementation using the same delay elements, power consumption, total area, and conversion time are reduced by 28.3%, 31.5%, and 24.6%, respectively. The AP-TDC has a figure-of-merit of 9.9-fJ/conversion step.

Index Terms—asynchronous, time-to-digital converter, TDC, pipelined, absolute value based conversion, time subtraction, completion detection

I. INTRODUCTION

With the scaling of the technology nodes, while the performance of the digital systems improve, the supply voltage, and thus, the headroom available for analog signal processing scales down as well. This imposes a strict limit on the analog signal processing capabilities. One solution to this problem is using time-domain signal processing (TDSP) techniques [1]–[4]. A hypothetical time-domain/digital hybrid signal processing chain is shown in Figure 1. In such a system, in addition to signal processing circuitry, two main blocks that are required to be able to communicate with a digital environment are time-to-digital (TDC) and digital-to-time (DTC) converters.

A TDC resolves a time difference into a digital value. The time difference can either be the difference between two separate signals, usually the start and the stop signals, or the time between the edges of a single signal. The result of the conversion is a digital word representing the time difference value, similar to the digital output word in analog-to-digital (AD) conversion. In addition to TDSP systems, TDCs find many applications in scientific experiments [5], such as digital phase-locked-loop (DPLL) applications [6], and on-chip time measurement and testing [7]. The simplest TDC implementation is based on a counter [1]. However, many TDCs employed today are based on simple delay lines and may be summarized as Flash TDCs [8], pipelined TDCs [9], [10], successive approximation TDCs [11]–[14], noise-shaping TDCs [15] and $\Delta - \Sigma$ TDCs [5].

The main goal of the research work presented here is to develop a low-power and small-area asynchronous pipelined TDC (AP-TDC) to be used in a hybrid signal processing chain as shown in Figure 1. As multiple instances of a TDC may be used in such a system, a pipelined architecture is chosen for both low power consumption and small area properties while having moderate conversion speed. An asynchronous TDC implementation was chosen over a synchronous one due to following advantages: i) lower power operation due to lack of a clocking signal, ii) faster average operating speed due to the completion detection, and iii) better interface to TDSP system
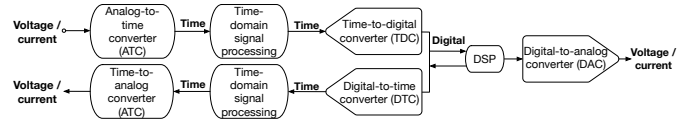

Figure 1. A hybrid time-domain signal processing chain.

as the signal processing is mostly event based [16] , e.g., the rising edge and falling edges of a signal carrying the information to be processed.

The contributions and organization of this paper are as follows: First, a novel time-domain subtraction method is introduced in Section II-A, as subtraction operation is a pre-requisite for pipelined conversion. Second, an absolute value based, power and area efficient pipelined value-to-digital conversion method suitable for asynchronous implementation with completion detection is presented in Section II-B. Combining absolute value based algorithmic conversion, time-domain subtraction and asynchronous operation, the design of an AP-TDC is presented in Section III. Simulation results are given in Section IV and finally conclusions are drawn in Section V.

II. TIME DOMAIN SUBTRACTION AND CONVERSION

This section presents the ideas that are employed during the design of the TDC. The first one is time-domain subtraction operation, and the second one is an algorithm for value-to-digital conversion using the absolute value of a residue with completion detection.

A. Time Domain Subtraction Operation

A pipelined TDC based on a binary-search algorithm (BS-TDC) was introduced in [2]. The BS-TDC is based on employing binary weighted delay elements and operated by either delaying the input or control (clock) signal in the TDC by binary weighted amounts. Such an approach needs twice the number of binary weighted delay elements and results in higher power consumption and a bigger chip area. Moreover, in [2] it was claimed that a time value cannot be subtracted from another one, hence the choice for parallel paths. In this section it is shown that subtraction of a fixed time value from an input pulse is possible and results in absolute value arithmetic.

The basic idea of time subtraction is that by delaying an *input* pulse by a certain amount (value to be subtracted), and XNORing the input signal with the delayed version of itself during a window defined by the rising edge of the original signal and the falling edge of the delayed version, the operation

$$dOut = |dIn - d| \tag{1}$$

is realized where *dOut* is the width of the resulting pulse, *dIn* is the pulse width of the input signal and *d* is the delay amount, i.e., the value to be subtracted. Outside this time window, the pulses on *dOut* are irrelevant. In a subtraction operation, there are three possible outcomes; a) positive, b) negative, and c) 0. The first two cases for
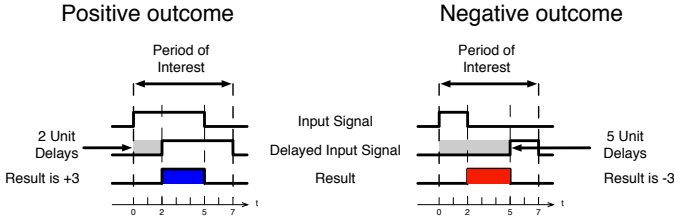
Figure 2. Two example time subtraction operations: Positive outcome operation represents $5 - 2 = 3$ and negative outcome time subtraction operation represents $2 - 5 = -3$.

the time subtraction operation are illustrated in Figure 2, while the 0 outcome case is omitted to be brief.

In Figure 2 even though the pulse-width of the resulting pulses are the same, it may be observed that the former pulse is generated by the AND logical operation and the second pulse is generated by a NOR operation on the signals during the period of interest, representing the positive and negative outcomes, respectively. Therefore, an XNOR operation, which is the union of AND and NOR operations, realizes the absolute value of the result in a time subtraction operation on an input signal and a fixed delay amount. By monitoring the events of AND and NOR operations during the period of interest, the sign of the result is easily deduced. The sign of the result is used in the conversion algorithm which is explained in the next sub-section.

The third case, which is not shown, where the outcome is 0, i.e., no output pulse, occurs when the falling edge of the input signal is aligned with the rising edge of the delayed signal, meaning the delay value matches the input pulse-width. From this observation, it is concluded that when no signal is generated at the output of XNOR operation, and when no events are generated at the outputs of AND and NOR operations, the outcome of the time subtraction operation is 0.

B. Absolute Value Based Algorithmic Conversion

In the proposed pipelined TDC architecture, a pulse value representing the bit-weight of the stage is subtracted from the input signal and a new pulse signal representing the residue is generated, similar to a standard pipelined ADC [17], [18]. In a standard pipelined ADC, amplification of the residue is required to keep the reference values and the input range for each stage fixed to reduce both the design overhead and verification efforts. However, no signal amplification is required in the proposed TDC as the proposed architecture can operate with changing reference and input range values without any overhead, as will be shown next.

The proposed algorithm is based on operating on the absolute value and the sign bit of the residue generated. During each conversion step, for an N-bit accurate conversion, a value of 2^{N-k} is subtracted from the value to be converted, where k is the conversion step number, which varies from 1 to N, and the first step ($k = 1$) converts the most significant bit (MSB). In this section **bold-faced** values represents the operation of the current stage and *italic* values belong to the preceding stage. During the conversion, the stage output bit is **1** if i) the output bit of the previous stage is *1* and the sign of the residue is **positive**, or ii) the stage output bit of the previous stage is *0* and the residue of the current stage is **negative**, or iii) regardless of the result of the previous stage, if no residue is generated, i.e., result of subtraction is 0, an output bit of **1** is generated. This last case, i.e., iii), is where a completion detection mechanism for the conversion can be implemented. All other conversion cases result in a stage output bit of **0**.
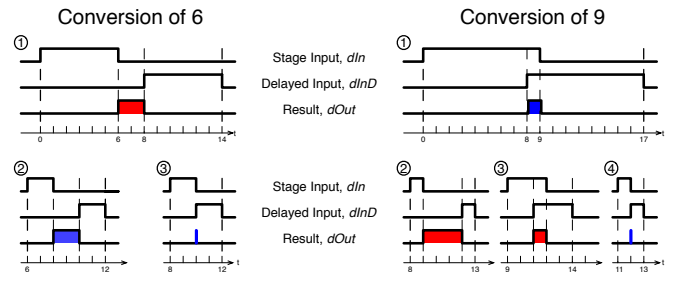

Figure 3. 4-bit pipelined, time-domain conversion examples based on absolute valued residue and time subtraction. Conversion steps are numbered above each sub-figure.

C. Application of Time Subtraction to Time-to-Digital Conversion

In this sub-section, application of time-domain subtraction to the absolute value based digital conversion is presented. Two 4-bit conversions utilizing time-domain subtraction are presented in Figure 3. The conversion steps are in order from top to bottom and left to right. It should also be noted that the time axis on the figures is continuous for the duration of the conversion, i.e., the conversion steps overlap.

As presented in Section II-A, the residue (*dOut*) is generated by XNORing the input signal *dIn* with its delayed version, which is represented with *dInD* from now on, during a pre-defined time window. In the figures, the red color represents a resolved value of 0, and blue represents a resolved value of 1.

During the current stage's conversion, if the resolved bit of the previous stage's is *1*, an event on the AND operation of **dIn** and **dInD** will result in a resolved output bit of **1**. An event on the NOR operation of the same signals will result in an output bit of **0**. Likewise, if the output bit of the previous stage is *0*, meaning the residue's sign is negative, the output bit according to the results of AND and NOR operations is reversed, i.e., a **1** is output for an event on NOR and a **0** is output for an event on AND. Regardless of the output bit of the previous conversion, if neither NOR nor AND operations create an event, it is concluded that the final value is being converted in the current step, an output bit of **1** is generated and the overall conversion is finished, effectively realizing completion detection.

III. PIPELINED ASYNCHRONOUS TDC

The conversion algorithm presented in the previous section, and the implementation that is presented in this section improve the conversion algorithm and the design presented in [11]. The implementation in [11] is synchronous, and requires a clocking signal , which results in higher power consumption and chip area. The operation principle of the circuit depends on the matching of multiple delay elements, causing problems in practical implementations. Another disadvantage of the synchronous implementation is feed-forward and feed-back signals between stages and complex timing requirements for correct operation. Furthermore, a custom clocked comparator is required in the implementation possibly introducing metastability into the system. The aforementioned problems are solved by adopting an asynchronous compatible algorithm and by asynchronous system implementation. Moreover, the employed algorithm realizes completion detection during conversion, effectively increasing the conversion speed.

Based on the absolute value and time-domain subtraction based conversion presented in Section II, an AP-TDC was designed using
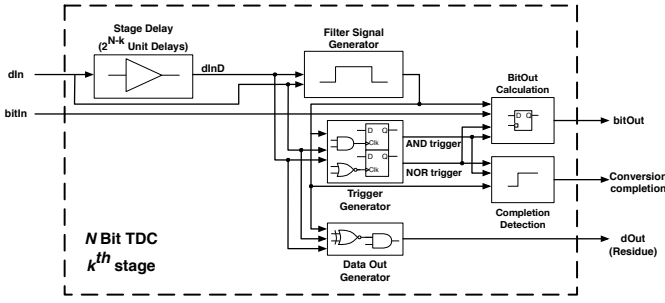
Figure 4. Block diagram of a stage of the designed asynchronous TDC.

a standard CMOS 65 nm process. Low-power high threshold voltage transistors were used for the delay elements and low-power standard threshold voltage standard cells were used for the rest of the circuitry. The delay elements are designed and characterized for 1V supply voltage operation. The architecture of the AP-TDC and implementation details are as follows:

A. Architecture

A block level diagram of a stage of the AP-TDC is shown in Figure 4. In the figure, *dIn* represents the pulse to be converted, the *bitIn* is the conversion result from the previous stage, *bitOut* is the converted value, and *dOut* is the residue to the next stage. A stage of the TDC consist of a delay element for delaying the input signal of the stage, a generator for the filter signal to mark the period of interest as shown in Figure 2, a trigger generator for keeping track of the events on AND and NOR operations, a residue generator for creating the output for the next stage, stage output bit calculator and completion detector. Except the delay element, all the circuitry is the same for all the stages. Furthermore, three points that are not illustrated in the figures should be noted: i) *bitIn* connection to the first stage is logic high as the value to be converted by the TDC is positive, ii) when an input less than the LSB is applied, all the *dOut* signals stay high in the system without switching to a low value. Hence a completion circuit just for the case of input less than 1 LSB is also implemented and ORed with the completion signal generated by the stages, and iii) all the sub-blocks are designed to be reset on the falling edge of the reset signal.

B. Sub-Blocks and Implementation Details

Residue generator: The logical function of the residue generator is to create a residue signal employing an XNOR gate and filtering the generated residue with a filter signal so that when both inputs are low, an event is not created at the output. This filtering operation is easily realized by ANDing the output of an XNOR gate with the filter pulse. However, during the design of the residue generator, it was found that the pulse-width of the signal generated by XNORing *dIn* and *dInD* deviated from the expected value, resulting in errors in the generated residue. Furthermore, the time-period where the difference between the edges of the signals *dIn* and *dInD* are small but existent, no output pulse is generated, resulting in a dead zone. This dead zone
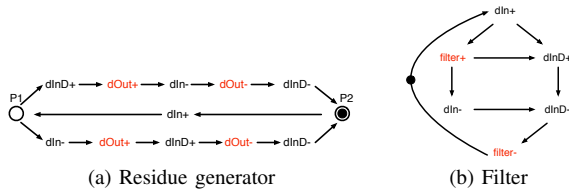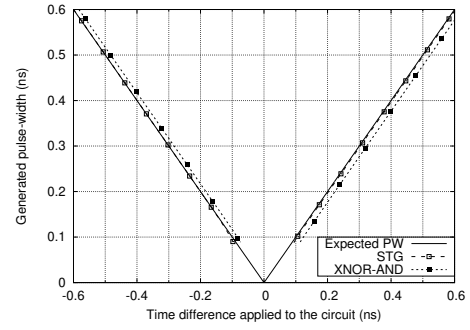

Figure 6. Simulation results of both residue generator implementations. Output response is shown for varying input time differences between the falling edge of *dIn* and rising edge of *dInD*.

Table I
PERFORMANCE COMPARISON OF RESIDUE GENERATOR IMPLEMENTATIONS.

Implementation	error @ lower end	dead zone range (ps)	error @ upper end
STG	-8%	-97,7 - 92	-7%
XNOR-AND	20%	-72 - 112	-22%

is basically the main limitation on the resolution of the TDC and reducing the dead zone and getting an accurate residue signal pulse-width wise is of utmost importance for increasing the time resolution. To improve the performance of the residue generator, asynchronous circuit design techniques were employed. The signal transition graph (STG) of the implemented residue generator is shown in Figure 5(a). The STG was synthesized and mapped to standard cell libraries using *Petrify* [19]. When compared to a more straightforward XNOR-AND implementation, STG based implementation has both better accuracy and a more symmetric dead zone. The transistor level simulation results for varying input time differences is shown in Figure 6, and the dead zone and accuracy results at the edges of the dead zone are presented in Table I. It can be seen from the simulation results that even though the STG implementation has a slightly longer dead zone, the accuracy at the edges of the dead zone is much better and the shape of the dead zone is more symmetric when compared to the XNOR-AND implementation.

Delay elements: In the proposed TDC implementation, the LSB value of the TDC is limited and set by either the minimum value of the unit delay element or the width of the dead zone of the residue generator, whichever is higher. Based on the simulation results of the residue generator, the STG implementation has a dead zone of 189.7 ps. Therefore, realizing a delay element with a delay shorter than this value is not feasible and the unit-delay element of the system was designed to be 200 ps, as delays shorter than the dead zone width would not improve the resolution of the TDC. For ease of implementation and presenting the system as a proof of concept, a CMOS buffer based unit delay element was designed, and binary weighted delay elements were created using the unit delay cell. Even though the number of delay elements is similar to a standard Flash-TDC, if the delay elements are designed in an analog fashion, the number of delay elements in the implementation is greatly reduced, i.e. from $2^N - 1$ to N.

In addition to the absolute value of the delay elements designed, the variation of the delays of each stage with respect to process variations is also important. As the variation of a delay element directly affects the output residue, low variation in the first stages of the pipeline is required. For example, for an *N*-bit AP-TDC, the delay variation

(a) Residue generator   (b) Filter

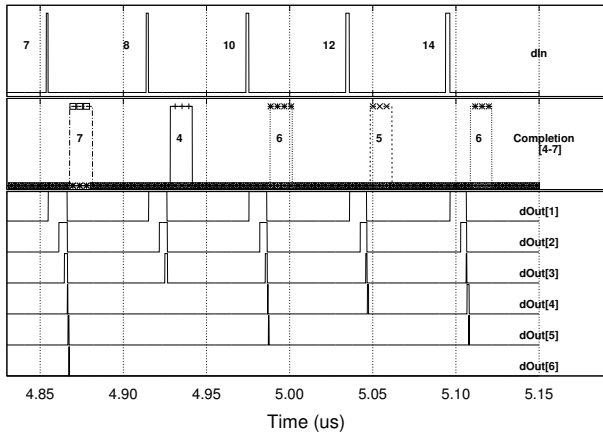Figure 5. Signal transition graphs of the synthesized asynchronous circuits

Figure 7. Transient simulation results of the AP-TDC. The panels from the top to the bottom are: Input data to the AP-TDC, completion signals for stages 4 to 7, and residues generated by each stage.

Figure 8. Power spectral density of the AP-TDC.

of the *kth* stage should be less than a certain amount such that the delay is $N - k + 1$ bit accurate and the following stage's resolution is not reduced. The effects of process variation on the delay elements were simulated using 1000 point Monte Carlo simulations and it was verified that the designed delay elements satisfy the accuracy requirements under 1-σ delay variation.

Filter signal generator: The filtering signal, which shows the period of interest in Figure 2, is a pulse signal that marks the time slot between the rising edge of the input signal and the falling edge of the delayed input signal. Such a circuit is easily implemented by the design of an STG and synthesizing the circuit using *Petrify*. The STG of the designed circuit is shown in Figure 5(b). The resulting filter signal is ANDed with the generated residue to create the *dOut* signal that is fed into the subsequent stage.

Trigger Generator: This block is used for keeping track of firing of either the AND or the NOR gate that provides the information whether *dIn* and *dInD* signals are overlapping or not. The implementation is shown in Figure 4. D-type flip-flops are triggered by a rising edge on their clock input and the D inputs of the flip-flops are connected to logic high, effectively storing the event that triggered.

BitOut Calculator: The output bit of the stage is calculated and stored in a D-latch. The D-latch is enabled by the filter signal, hence following the generated value throughout the period of interest, and storing the value with the falling edge of the filter signal.

Completion Detector: The completion signal is generated when neither AND nor NOR events are triggered during a conversion. This block is triggered when both edges are in the dead zone of the residue generator, meaning the time difference between the edges is less than the resolution of the TDC, hence signaling the completion of the overall conversion process.

IV. SIMULATION RESULTS

The proposed 7-bit asynchronous pipelined TDC has been designed in a standard 65 nm process using low-power and standard threshold voltage process options for a 1 V supply. Transistor level simulations were run using HSPICE and a transient simulation of the TDC is shown in Figure 7. During the presented simulation window, 5 distinct values, which are shown on the top panel, are converted. It is easily observed that completion signals are generated by different stages during the conversion process. In the middle panel, the stages generating the completion signal are marked, 7 being the LSB stage. Furthermore, generated residues throughout the pipeline are shown
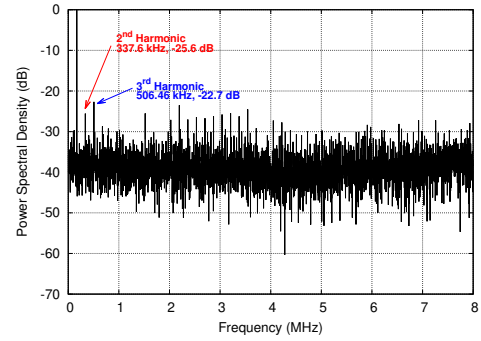
in the bottom panel of the figure. From the simulation results it is calculated that, the AP-TDC completes the conversion process 24.6% faster using the proposed algorithm with completion detection, as compared to a conventional Flash TDC.

The performance of the TDC was evaluated with respect to both static metrics such as differential non-linearity (DNL) and integral non-linearity (INL) and dynamic metrics such as signal-to-noise ratio through transient simulations. For the static case, INL and DNL were calculated using the histogram method. The pulse-width of the input signal to the TDC was increased by 1 ps during each conversion cycle and the converted value was sampled with the completion signal. AP-TDC DNL varies between 0.43 and -0.77 LSB, and INL varies between -0.05 and -2.11 LSB.

The dynamic performance was evaluated using a signal with varying pulse-widths based on a sampled sine wave. The sampling frequency and the sine wave frequency was set to 16.66 MHz and 168.82 kHz, respectively. The converted digital word was sampled with each completion signal. A 8192-point FFT was applied to the recorded data and power spectral density was calculated, as plotted in Figure 8. Dynamic simulation results show an SNR of 39.2 dB, and an effective accuracy of 6.22 bits. Based on the layout generated, the AP-TDC has a total area of $1275\,\mu m^2$, of which half of it is the area due to delay elements. While operating from a 1 V supply voltage, the TDC consumes $38\mu A$. For comparison, a 7-bit flash TDC implemented in the same technology using the same unit-delay element has a total area of $1862\,\mu m^2$, and has a current consumption of $53\,\mu A$, both numbers excluding the thermometer to binary converter of the Flash TDC. Based on the figure-of-merit (FOM) defined in [6], the AP-TDC has a FOM of 9.9-fJ/conversion step.

V. CONCLUSIONS

This paper presents the design of an asynchronous pipelined TDC. An absolute value based conversion algorithm is modified to include completion detection and later converted to a form that employs time-domain subtraction and is suitable for asynchronous implementation. AP-TDC design, when compared to a Flash TDC design using the same delay elements, reduces the total area and power consumption by 31.5% and 28.3%, respectively. Furthermore, the conversion process is 24.6% faster owing to completion detection. Further power and area reduction may be possible by implementing the delay elements in an analog fashion.

ACKNOWLEDGEMENT

REFERENCES

[1] F. Yuan, "Cmos time-to-digital converters for mixed-mode signal processing," *The Journal of Engineering*, January 2014. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/joe.2014.0044

[2] D. Miyashita, R. Yamaki, K. Hashiyoshi, H. Kobayashi, S. Kousai, Y. Oowaki, and Y. Unekawa, "An LDPC decoder with time-domain analog and digital mixed-signal processing," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 1, pp. 73–83, Jan 2014.

[3] Z. Chen and J. Gu, "Analysis and design of energy efficient time domain signal processing," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*. ACM, 2016, pp. 100–105.

[4] V. Ravinuthula, V. Garg, J. G. Harris, and J. A. B. Fortes, "Time-mode circuits for analog computation," *International Journal of Circuit Theory and Applications*, vol. 37, no. 5, pp. 631–659, 2009. [Online]. Available: http://dx.doi.org/10.1002/cta.488

[5] Y. Cao, W. D. Cock, M. Steyaert, and P. Leroux, "Design and assessment of a 6 ps-resolution time-to-digital converter with 5 mgy gamma-dose tolerance for lidar application," *IEEE Transactions on Nuclear Science*, vol. 59, no. 4, pp. 1382–1389, Aug 2012.

[6] J. Borremans, K. Vengattaramane, V. Giannini, B. Debaillie, W. V. Thillo, and J. Craninckx, "A 86 mhz-12 ghz digital-intensive pll for software-defined radios, using a 6 fj/step tdc in 40 nm digital cmos," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 10, pp. 2116–2129, Oct 2010.

[7] D. J. Kinniment, O. V. Maevsky, A. Bystrov, G. Russell, and A. V. Yakovlev, "On-chip structures for timing measurement and test," in *Proceedings Eighth International Symposium on Asynchronous Circuits and Systems*, April 2002, pp. 190–197.

[8] T. J. Yamaguchi, S. Komatsu, M. Abbas, K. Asada, N. N. Mai-Khanh, and J. Tandon, "A cmos flash tdc with 0.84-1.3 ps resolution using standard cells," in *2012 IEEE Radio Frequency Integrated Circuits Symposium*, June 2012, pp. 527–530.

[9] T. Oh, H. Venkatram, and U. K. Moon, "A time-based pipelined adc using both voltage and time domain information," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 961–971, April 2014.

[10] K. Kim, W. Yu, and S. Cho, "A 9 bit, 1.12 ps resolution 2.5 b/stage pipelined time-to-digital converter in 65 nm cmos using time-register," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 1007–1016, April 2014.

[11] K. O. Ragab, H. Mostafa, and A. Eladawy, "Tdc sar algorithm with continuous disassembly (sar-cd) for time-based adcs," in *5th International Conference on Energy Aware Computing Systems Applications*, March 2015, pp. 1–4.

[12] R. Jiang, C. Li, M. Yang, H. Kobayashi, Y. Ozawa, N. Tsukiji, M. Hirano, R. Shiota, and K. Hatayama, "Successive approximation time-to-digital converter with vernier-level resolution," in *2016 IEEE 21st International Mixed-Signal Testing Workshop (IMSTW)*, July 2016, pp. 1–6.

[13] A. Mantyniemi, T. Rahkonen, and J. Kostamovaara, "A cmos time-to-digital converter (tdc) based on a cyclic time domain successive approximation interpolation method," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 11, pp. 3067–3078, 2009.

[14] H. Chung, H. Ishikuro, and T. Kuroda, "A 10-bit 80-ms/s decision-select successive approximation tdc in 65-nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 5, pp. 1232–1241, 2012.

[15] A. Elshazly, S. Rao, B. Young, and P. K. Hanumolu, "A noise-shaping time-to-digital converter using switched-ring oscillators: Analysis, design, and measurement techniques," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 5, pp. 1184–1197, May 2014.

[16] M. Miskowicz, Ed., *Event-Based Control and Signal Processing*. CRC Press LLC, 2017.

[17] T. B. Cho and P. R. Gray, "A 10 b, 20 msample/s, 35 mw pipeline a/d converter," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 3, pp. 166–172, March 1995.

[18] S. H. Lewis, H. S. Fetterman, G. F. Gross, Jr., R. Ramachandran, and T. R. Viswanathan, "A 10-b 20-msample/s analog-to-digital converter," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 3, pp. 351–358, March 1992.

[19] M. Kishinevsky, J. Cortadella, and A. Kondratyev, "Asynchronous interface specification, analysis and synthesis," in *Proceedings 1998 Design and Automation Conference. 35th DAC. (Cat. No.98CH36175)*, June 1998, pp. 2–7.