FPGA Implementation of Farrow Structure for Time Interleaved Analog to Digital Converter Clock Skew Compensation

Maicon Bruno Hofmann[1], Sibilla B. L. França[2], André Mariano[1,2]
[1] CPGEI - UTFPR
[2] GICS - UFPR
E-mail: hofmannmb@gmail.com

***Abstract*— In modern wireless communication systems, TIADC arises as a good candidate for applications with high sampling rate and moderate power consumption requirements. However the performance of TIADC is impaired due to clock skew between the various ADCs. Clock skew performance degradation can be mitigated by fractional delay filters. Focusing on the clock skew mitigation this paper presents a fractional delay filter implemented in FPGA, based on Farrow structure. The procedure used for TIADC modeling and implementation of digital filter in FPGA is presented in detail.**

I. Introduction

Recent wireless front-end receivers dedicated to mobile applications operate at least two frequency translations before I/Q demodulation. Frequency translation increases the system complexity, introducing several issues associated with the down-conversion mixers (dynamic range limitation, noise injection from the local oscillator, etc.). Herein, the position of the analog-to-digital interface in the receiver chain can play a major role. Shifting the ADC closer to the antenna enables eliminating some analog functions such as amplifiers, filters and mixers (integrated circuits and/or external components). These functions can be easily implemented in the digital domain, reducing the receiver complexity and therefore the power consumption. Nonetheless, the bottlenecks are the ADC required specifications in terms of sampling rates and resolution, which become more severe with the antenna closeness. In order to reach these strict specifications, TIADC (Time Interleaved Analog to Digital Converter) emerges as an interesting solution [1]. A TIADC consisting of M ADCs configured to perform the acquisition of the same analog signal interleaved in time is illustrated in 1.

This configuration allows each ADC to perform conversion at F_s sample rate, obtaining a resulting $F_c = MF_s$ times sample rate. Theoretically resulting TIADC system resolution is the same of their ADCs counterparts, but ADC differences in gain, offset and in the time of acquisition due to clock skew, degrade the resolution of TIADC [2] [1] [3]. However gain and offset errors can be mitigated with orthogonal calibration techniques [4]. Timing mismatches are still in intense research and is not entirely solved [2] [3]. In [1] calibration techniques and delay adjustment in sample and hold converter were used. Another approach that not depends from analog circuitry and

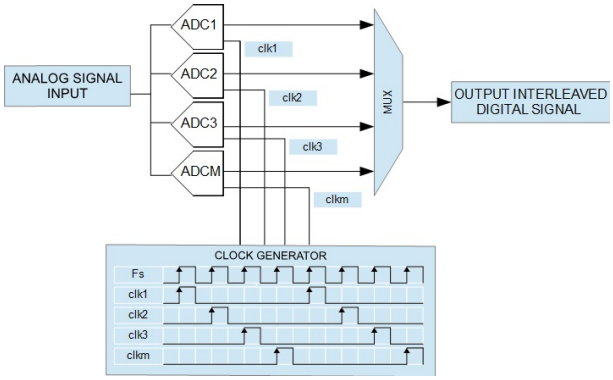


Fig. 1. TIADC system illustration. Bottom: clock control; top: multiplexed TIADC output at Fs sample rate.

can be done in digital domain is the use of fractional delay filters, proposed by [5]. The implementation of digital circuits for analog response improvement is interesting because digital correction does not suffer from the performance issues of analog circuits, like aging and temperature effects. Of particular interest to fraction delay filter implementation in hardware is the Farrow structure, which allows to obtain a fractional delay filter for different delay values by means of a tunable fixed coefficients structure. In addition, Farrow structure does not require recalculation or storage of coefficients in memory. This article presents a procedure for hardware implementation of Farrow structure from a Simulink filter modeling. The paper is organized as follows: fractional delay filters and Farrow structure are reviewed in section II, TIADC and correction filter modeling is presented in section III, implementation of the correction filter modeling and the structure of loop test is show in Section IV and Section V is held for the presentation of results. Finally in section VI space is reserved for discussion of the results and final conclusions.

II. Fractional Delay Filters and Farrow structure

Delay values that are integer multipliers of acquisition sample time are easily obtained with memory elements. However, to obtain delay values that are fractions of acquisition sample time there is a need to use digital filters structures. Ideally this

filter should keep the signal amplitude across the spectrum in frequency and move the signal of a fractional delay D, i.e., the filter should have linear phase and unity gain [6]. The frequency response of this filter should be of the form:

$$H(e^{jw}) = e^{-jwD}, |w| \leq \pi \quad (1)$$

This filter is not feasible, since the requirement of a constant amplitude across the spectrum makes it non causal. The filter designed will be an approximation of the ideal one. The design of this approximation filter can be done by different design methods, like the least square error, maximally flat and minimax [6] design. One of the simplest methods for fractional delay filter design is the use of Lagrange interpolator, where the coefficients can be obtained with equation 2.

$$h(n) = \prod_{k=0, k\neq n}^{N} \frac{d-k}{n-k} \quad n = 0, 1, 2, ...N. \quad (2)$$

Where $h(n)$ are the filter fixed coefficients, d is the fractional delay and N is the filter order.

In several applications of fractional delay filters different delay values are needed, which requires on-line design of filter coefficients or the use of a large amount of memory to store the coefficients of k desired filters. An alternative to this approach were proposed in [7] by Farrow. Consider the design of M fractional delay filters with different delay values in the range [0, 1]. Now, the same order $h_m(n)$ tap coefficients of M filters designed can be approximated by a polynomial of order $M-1$ on fractional delay d as

$$h_m(n) = \sum_{m=0}^{M} c_m(n) d^m \quad n = 0, 1, 2, ..., N. \quad (3)$$

Where c_m are the real coefficients polynomial approximation of the fractional delay filters. Farrow structure is used to obtain an approximated fractional delay filter in delay value d. Equation 3 can be implemented as the sum of the product of M fixed filters in parallel multiplied by a fractional delay factor d, as shown in Fig. 2. Because Farrow structure can be implemented as shown in Fig. 2 it is interesting for hardware implementation, allowing the use of different fractional delay filters only controlling delay d. In this article is presented an implementation of two approaches for obtaining Farrow coefficients: *FarrowLagrange* [8] and *least square polynomial fit* (LSP) [6].

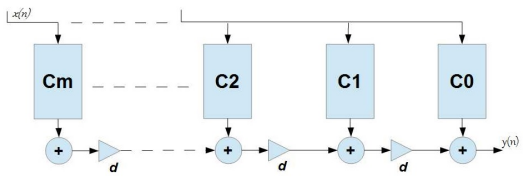


Fig. 2. Farrow Structure.

III. TIADC AND FILTER MODELING

In this section TIADC and filters used for clock skew compensation modeling are presented. Simulink software was chosen for modeling all elements in this article. The overall procedure to modeling, test and analyse TIADC and to validate the filter are illustrated in Fig. 3. Four main blocks are shown: *Simulink*, *Quartus*, *FPGA* and *Computer*. In *Simulink* block TIADC acquisition clock skew errors are simulated for an input sinusoidal signal (block *TIADC model*). This data is the input for filter (block *Digital filter model*) and to a *.mif* file converter (block *mif*). The output data from the filter (block *Filter output data*) is converted to a *.mif* file and goes to *Computer data analysis*. The filter model is converted to a VHDL code and is used as a individual component of the project, together with converted input and output data in *Quartus* block (follow indicative arrow's flow). *Quartus* block integrates all project elements to be loaded in FPGA with a control and validation logic (block *Validation logic*). FPGA block represents the loaded Quartus project in FPGA board and the analysis tool SignalTap [9]. SignalTap reads the filter, validation logic and simulated data and show it on computer. Validation logic will confirm if simulated filter output data is equal to the data from tested filter in FPGA (with the same input data), validating FPGA filter implementation.

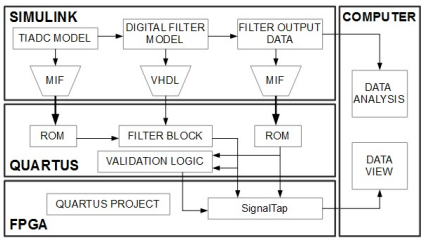


Fig. 3. Overall procedure block diagram.

Fig. 4 illustrates the modeling for a TIADC containing two 8-bit ADCs in parallel, a source of sinusoidal signal, a filter with Farrow structure to compensate clock skew and an input for the fractional delay value d (block *CorrectionDelay*).

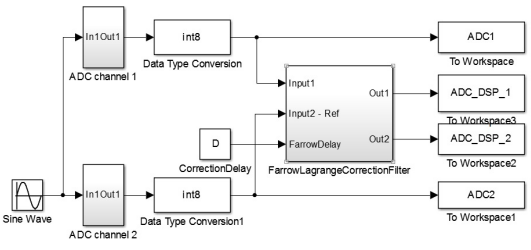


Fig. 4. TIADC modeling using *FarrowLagrange* correction filter.

In the proposed modeling the outputs ADC1 and ADC2 are the signals without correction, while ADC DSP 1 and ADC DSP 2 are the signals after correction by the filter. The output signal from the multiplexer is not implemented in this model for simplicity, but multiplexing is further performed using a Matlab script. In modeling, both ADCs are controlled by the same sampling frequency, with a delay of at least half sampling

period. Fig. 5 presents the structural modeling of individual ADCs.

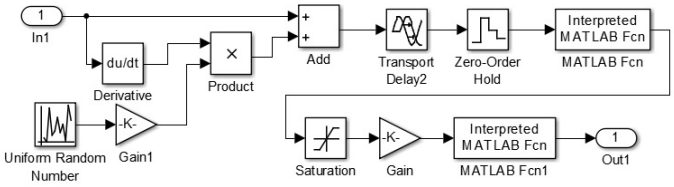


Fig. 5. ADC generic model used in simulations.

At each ADC input is added a white noise signal. The clock skew effect is simulated by means of *TransportDelay* block. In this modeling the sampling time of ADC2 suffers from a fractional delay value d. So, in correction filter implementation the signal from ADC1 is delayed by the same fraction d in Farrow filter. The signal from ADC2 is delayed by an integer amount $Dint$ to compensate the integer delay value added by Farrow filter. The modeling used for *FarrowLagrange* is shown in Fig. 6 and Farrow filter with *LSP* is shown in Fig. 7.

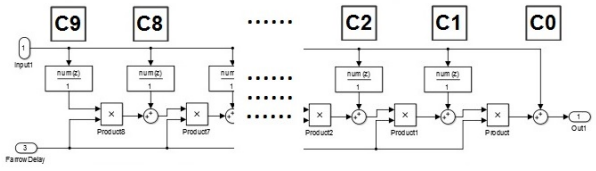


Fig. 6. Simulink model for *FarrowLagrange* reconstruction filter. FarrowLagrange with C9 to C0 FIR filters.

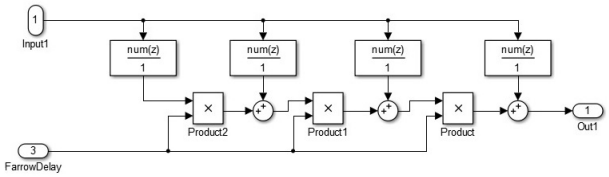
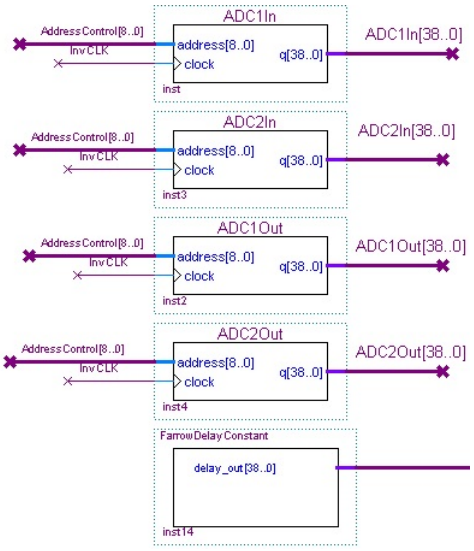


Fig. 7. Simulink model for *Farrow* with third order polynomial reconstruction filter.

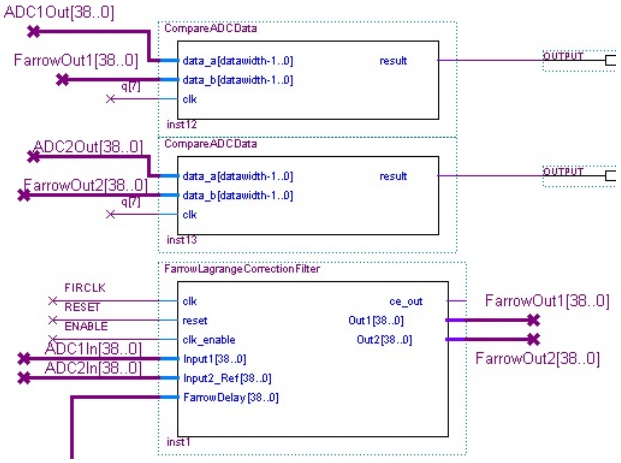
Besides the different methods to calculate filter coefficients, the basis filter order and type are the same in both cases: ninth order Lagrange filter ($N = 10$ in equations 3 and 2). Filters were implemented with fixed point coefficients. For *FarrowLagrange* were used 39 bits of word length (sign included) and 28 bits dedicated for fractional number part. For *Farrow* approach equivalent results were obtained with third order polynomial Farrow structure with 12 bits word length and 10 bits of fractional part. Word length selection criteria is the minimum value for SFDR performance degradation equal or less than 2 dB from floating point simulation results. The performance results and implementation in FPGA are discussed in next sections.

IV. FPGA FILTER IMPLEMENTATION

FPGA implementation of filters modeling were done with Matlab HDLCoder tool. As explained in section III, input and



(a)



(b)

Fig. 8. Schematic created in Quartus for filters test. (a) ROM memories. (b) Filter and comparator.

output simulated data were converted to *.mif* files and filters modeling were converted for a VHDL code. For each filter design was created a project in Quartus II Web Edition 10.0 with the structure shown in Fig. 8. For tests were used one *Terasic DE2-115* development board, that contains an Altera Cyclone IV E FPGA (part-number *EP4CE115F29C7*).

The filter test structure consists of four ROMs, two comparators, an address counter, two separated switches debounce circuits and one prescaler. The prescaler is used to divide the 50MHz clock frequency to a subset of lower synchronized frequencies to control data load from ROM and to control filter and validation logic processing. Two switches are used to implement reset and enable function. When enabled, the lowest clock frequency from prescaler outputs to address counter. At each positive clock edge address value is incremented. The address counter output is connected on each ROM memory and its contents are loaded on positive clock edge of *InvClk* signal (Fig. 8). On each positive clock edge of *FIRCLK* signal the data at the input of filter is loaded and processed by

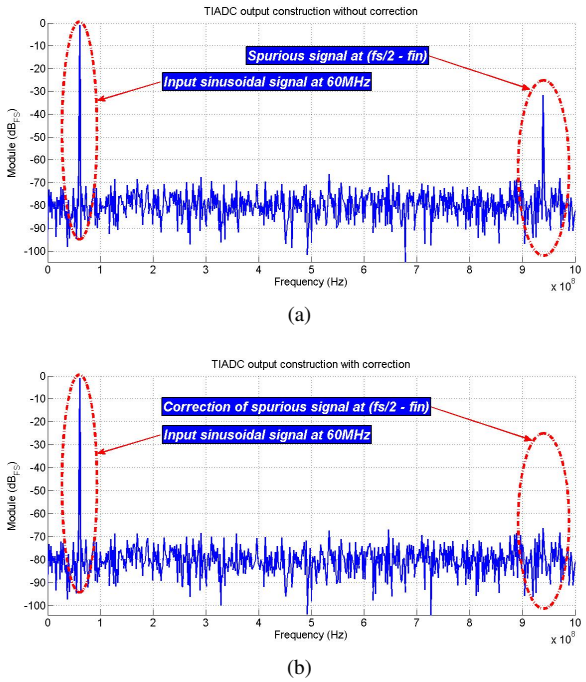


Fig. 9. TIADC spectrum. (a) without filter. (b) with Farrow filter.

the implemented Farrow structure. Filtering process is made in just one clock cycle, because the Farrow structure filter was implemented in a full parallel form. The output data from the Farrow filter is then compared with the simulated data stored in ROM memories (*ADC1Out* and *ADC2Out* in Fig. 8) to verify if the expected response from the implemented filter is obtained. Comparators output, ROM memories values and Farrow filter output were monitored using Quartus II SignalTap tool. The resource allocation for each Farrow filter implemented and the verification results are presented in section V.

V. Results

Tests were performed in FPGA filter design with generated TIADC modeling data. For each test was chosen a sinusoidal input frequency signal and a delay value to be added as clock skew in sampling time of ADC2. The complete spectrum of TIADC operating at a sampling frequency (*fs*) of 2GHz with an input signal (*fi*) of 60MHz before and after *FarrowLagrange* filter correction is shown on Fig. 9.

It should be noted the effectiveness of Farrow filter to mitigate the spurious signal in the right of the spectra, at $fs/2 - fin$ frequency in Fig. 9. The elimination of the spurious signal implies in an improvement on *Signal to Noise Ratio* (SNR) and *Spurious Free Dynamic Range* (SFDR) of TIADC structure. A comparison of SFDR performance at one delay value between TIADC without compensation and the two implementation approaches for Farrow filter is shown in Fig. 10. The improvement in SFDR performance were higher than 30dB between 100MHz to 250MHz.

The returned screen from SignalTap tool for FPGA tests realized is shown in Fig. 11. Note that the reference points for signal analysis (in bold boxes), while Fig. 11.b depicts

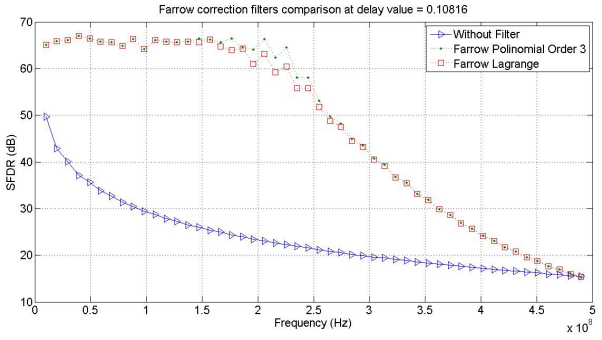
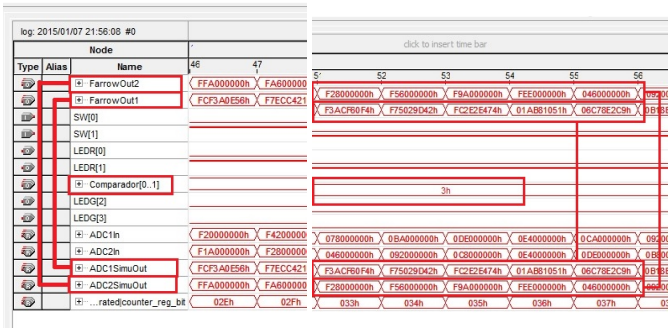


Fig. 10. SFDR comparison between two Farrow structure design approaches.

part of SignalTap screen test. As can be seen the output of both comparators is in high level (value *0x3h*, marked by a bold box) for the entire verified window and the data output from Farrow filter (*FarrowOut1* and *FarrowOut2* in Fig. 11.b) is equal with stored data in ROM memories (*ADC1SimuOut* and *ADC2SimuOut*). This result validates the implementation, showing that the results obtained in FPGA are equivalent to results obtained in software simulations.



(a) (b)

Fig. 11. SignalTap screen showing the filter output data and comparison with simulated stored data. (a) Organization of data setup. (b) Midle-point of data indicating comparators result

Table I compares the two Farrow structures design approach and the allocated FPGA resources for each approach. Fixed coefficients are considered different from variable ones because is possible to obtain further improvements with SOPOT techniques [10]. In table I the resource usage of FarrowLagrange is more than ten times the resource usage of Farrow with (LSP) approach. As FarrowLagrange coefficients used are three times

TABLE I
Farrow structure design approach and FPGA resources (percentage in parenthesis) comparison

Filter Approach	FarrowLagrange	Farrow
Fixed coefficients multipliers	91	31
Variable multipliers	9	3
Adders	100	33
LUT (%)	9777 (9%)	1711 (1%)
REGISTERS (%)	104 (<1 %)	104 (<1 %)
9-bit Multipliers (%)	420 (79 %)	16 (3 %)

the Farrow LSP, it is expected to have more resource usage, specially because Matlab considers the worst-case number of bits at multiplier output.

VI. Conclusion

Modern wireless communication systems call for high resolution and moderate power consumption ADCs. TIADCs appear as an interesting solution, despite the dynamic range limitation due to clock skew effects. To overcome these effects, a FPGA implementation of Farrow structure from Simulink modeling for clock skew compensation was presented. Both Farrow structures mitigate the effects of clock skew error with nearly the same results. However, the design of Farrow structure using a LSP approximation is clearly more advantageous in terms of hardware resources usage than FarrowLagrange approach. The implementation of Farrow structure in FPGA allows to validate the feasibility of the proposed clock skew compensation method. In addition, this design procedure is an interesting way to evaluate the performance of the filter in a short time and simplified development cycle. Future works will be able to consider different fractional delay filters design with Farrow structure and resource usage optimization.

Acknowledgements

The authors wish to acknowledge Dr. Dallet by the first version of TIADC modeling.

References

[1] B. Razavi, "Design Considerations for Interleaved ADCs," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 8, pp. 1806–1817, August 2013.

[2] C. Vogel and H. Johansson, "Time-Interleaved Analog-To-Digital Converters:Status and Future Directions," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 2006.

[3] C. Vogel, "Modeling, identification and compensation of channel mismatch errors in time interleaved analog-to-digital converters," Ph.D. dissertation, Graz University of Technology, Austria, July 2005.

[4] G. Ferré, B. L. Gal, L. Bossuet, M. Jridi, and D. Dallet, "Orthogonal Correction Implementation for Time-Interleaved Analog-To-Digital Converters: Realtime application," in *In Proceedings of the 16th EURASIP European Conference on Signal Processing, August 2008.G.*, 2008.

[5] H. Johansson, "A polynomial-based time-varying filter structure for the compensation of frequency-response mismatch errors in time-interleaved ADCs," *IEEE J. of Selected Topics in Signal Processing*, vol. 3, no. 3, pp. 384–396, June 2009.

[6] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay - tools for fractional delay filter design," *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60, January 1996.

[7] C. W. Farrow, "A continuously variable digital delay element," in - *Circuits and Systems, 1988., IEEE International Symposium on*, 1988.

[8] V. Valimaki, "A new filter implementation strategy for Lagrange interpolation," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS-95)*, vol. 1, 1995, pp. 361–364.

[9] Quartus II Software and SignalTap tool. [Online]. Available: www.altera.com/downloads/download-center.html

[10] W. Wang, J. Yan, D. Shi, and Z. He, "Optimization and Design of Fractional-delay Filters Based on Farrow Structure," *Journal of Computational Information Systems*, vol. 8, no. 24, pp. 10 089–10 095, Dec 2012.