FPGA Based Implementation of a Predictive Floating-Point Analog-to-Digital Converter

Voicu Groza, Boris Dzerdz
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada, K1N 6N5
VGroza@UOttawa.ca, Dzerdz@IEEE.org

Abstract *– Floating Point Analog-to-Digital Converters (FP-ADC's) were developed and used to quantize large dynamic range signals in applications where large signals need not be encoded with a precision greater than that required for small signals. Comparing floating-point with uniform quantization, it was shown that FP-ADC requires much smaller silicon area for the same dynamic range, but at the cost of doubling the conversion time.*

To improve the resolution and speed of conversion of such an FP-ADC, a higher precision predictive floating-point architecture was conceived (PFP-ADC). The PFP-ADC consists of two parallel uniform A/D converters, a D/A converter, a fixed-gain amplifier and a subtraction circuit. The current subtrahend of the subtraction circuit is based on the previous sample acquisition, while the current minuend is the measured signal itself. Determination of mantissa and exponent occurs in parallel.

This paper presents the principle used to improve the resolution of FP-ADC quantized signals, and its proof-of-concept FPGA based implementation. The resulting improved SNR that was achieved by using the proposed FP-ADC is better than that of other FP-ADCs, while the conversion time is shorter due to the use of prediction techniques and statistical characteristics of the measured signals.

Keywords: -- Quantization, floating point arithmetic

I. INTRODUCTION

In recent years, floating-point analog-to-digital converters have been conceived and designed for quantizing large dynamic signals, mainly in telecommunications [1] and high-energy physics instrumentation [2].

The classic two-cycle FP-ADC architecture employs a uniform quantization ADC connected to the acquired signal through a programmable gain amplifier (PGA) [3]. Initially, the ADC performs a coarse conversion cycle to detect the exponent and then, setting the PGA gain correspondently, it executes a second finer conversion to determine the mantissa. Since a reduced resolution is needed for exponent acquisition, the architecture presented in [4] shortens the total conversion time by overlapping the fine quantization cycle over the last part of the coarse quantization. To further speed-up conversion process, the architecture with two ADC's (one for coarse quantization and the other for fine quantization) was proposed in [5]. Still, having the two quantizers connected in cascade, the two quantization cycles are disjunctive and speed improvement is mainly obtained from using a variant of flash ADC for the exponent acquisition.

The parallel FP-ADC architecture [6] minimizes the conversion time by employing two ADCs that work simultaneously: one determines the exponent, while the other one, connected to the quantizer input over PGA, finds the mantissa. The PGA gain is based on the prediction of the input signal. If the acquired signal falls in the same range as its prediction, the conversion result is delivered immediately; if not, the mantissa is acquired again with the PGA gain reset to the most recently acquired exponent.

All the above FP-ADCs fail to give satisfactory results when acquiring a complex signal formed from the summation of a small amplitude / high frequency signal with a very slow, but large signal. The solution to such applications is the Higher Precision Predictive Floating-Point Analog-to-Digital Converter (PFP-ADC) presented in this paper. The PFP-ADC employs differential quantization to get a higher precision mantissa.

II. STRUCTURE OF THE PROPOSED FLOATING-POINT ANALOG-TO-DIGITAL CONVERTER

The block diagram of the Higher Precision Predictive Floating-Point Analog-to-Digital Converter is presented in Fig.1. It consists of two Analog-to-Digital converters (ADC-M and ADC-E), one Digital-to-Analog converter (DAC), one fixed-gain expansion amplifier (AMP) and a subtraction circuit (SUB). Both ADC-M and ADC-E are uniform quantization analog-to-digital converters (ADC) with the same resolution *m*, where ADC-M measures the mantissa, and ADC-E is used to determine the exponent of the quantized signal.

The two quantization cycles that are carried out in two distinct time intervals in the classic FP-ADC method are executed in parallel in this circuit.

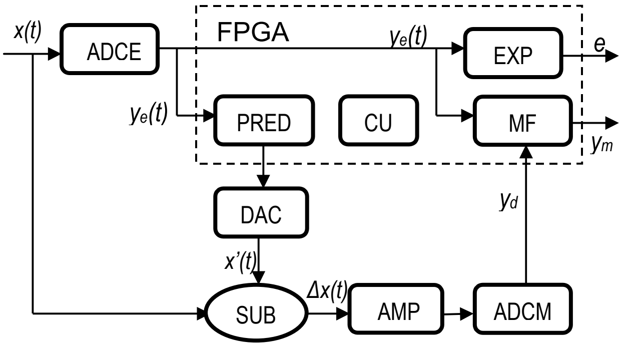

Figure 1: Block Diagram of PFP-ADC Concept

Availability of the measurement output is dependent only on the propagation delay of the circuit and statistical nature of the measured signal. The output is available in the form of an IEEE standard floating-point representation. The output word formatting is performed in the FPGA based control unit, which also controls the DAC and PGA setting.

The coarse quantization, accomplished by the ADC-E, provides a rough value of the current value of the input signal, which is directly fed to the FPGA for encoding the exponent, as well as being inputted to the predictive circuitry (PRED) in the FPGA to forecast the future value of the quantized signal to being applied to the DAC.

The quantization function of such an ADC is given by the analytical expression (1), that extends the uniform quantizer characteristic 10[g] in the *overload region* ($|x| > V_{FS}/2$):

$$y_e = \begin{cases} -(2^m)\cdot\frac{\Delta}{2} & \text{for } x < -(2^m)\cdot\frac{\Delta}{2} \\ \left\lfloor \frac{x+\frac{\Delta}{2}}{\Delta} \right\rfloor \cdot \Delta & \text{for } x \in \left[-(2^{\mathrm{m}})\cdot\frac{\Delta}{2}, (2^{\mathrm{m}}-1)\cdot\frac{\Delta}{2} \right] \\ (2^m-1)\cdot\frac{\Delta}{2} & \text{for } x > (2^m-1)\cdot\frac{\Delta}{2} \end{cases} \tag{1}$$

where $\Delta = \frac{V_{FS}}{2^m}$

and $\lfloor a \rfloor$ represents the greatest integer less than or equal to a. The exponent e can be expressed as a function of y_e as shown by equation (2)

$$e = f(y_e) = \begin{cases} 0 & \text{if } x \in \left[-\frac{\Delta}{2}, \frac{\Delta}{2} \right) \\ \left\lfloor \log_2 \left| \frac{y_e}{\Delta} \right| \right\rfloor + 1 & \text{if } x \notin \left[-\frac{\Delta}{2}, \frac{\Delta}{2} \right) \end{cases} \tag{2}$$

This equation (2) is implemented by EXP, which calculates the exponent e for PFP-ADC. To express the exponent as an E-bit number, the ADC has a resolution $m \geq 2^E$-1, with m and E natural numbers ($m \in \boldsymbol{N}$, $E \in \boldsymbol{N}$). Powers of 2 are chosen for m (i.e., $m = 2^E$) to take full advantage of the m-bit resolution in expressing the exponent values. The exponent e performs a partition of the range of the quantized signal x in *measurement domains* $\{D_e, e \in [0, 2^{\mathrm{E}}) \cap \boldsymbol{N}\}$, as given by (3)

$$e = \begin{cases} 0 & \text{if } x \in D_0 = \left[-\frac{\Delta}{2}, \frac{\Delta}{2} \right) \\ \left\lfloor \log_2 \left| \frac{y_e}{\Delta} \right| \right\rfloor + 1 & \text{if } x \in D_e = \left[-\left(2^{e+1}-1\right)\frac{\Delta}{2}, -\left(2^e-1\right)\frac{\Delta}{2} \right) \cup \\ & \left[\left(2^e-1\right)\frac{\Delta}{2}, \left(2^{e+1}-1\right)\frac{\Delta}{2} \right) \\ 2^E-1 & \text{if } x \in D_{2^E-1} = \left(-\infty, -\left(2^{2^E-1}-1\right)\frac{\Delta}{2} \right) \cup \\ & \left[\left(2^{2^E-1}-1\right)\frac{\Delta}{2}, \infty \right) \end{cases} \tag{3}$$

The Digital to Analog Converter (DAC), on the other hand, converts this binary value $y_e(t)$ back to the analog domain, for subtraction from the original input signal. This results in the difference signal (*Δx(t) = x(t)-x'(t)*) being produced and applied to the input of the fixed-gain amplifier AMP. Since *Δx(t)* < 2^m, the AMP gain is set to 2^m to scale this difference to the full input range of the ADC-M.

The quantized difference y_d produced in the finer quantization cycle by ADCM is added to the (correspondingly weighted) coarse quantization result y_e (generated by ADCE) in the Mantissa Formatting Module (MF) and shifted to the left to get the normalized form of the quantized signal. Since the most significant bit of the normalized mantissa represents the minimum value of the *measurement domain* D_e, it represents a redundant information that is already implied by the exponent value and should not be sent further. If N bits are used to express the mantissa ($N < 2m$) the quantization function of the PFP-ADC, $\hat{y} = Q(x)$, is given by equation (4):

$$Q(x) = \left(y_e + \frac{y_m}{2^m} \right) \cdot 2^{N-m+e} \cdot \left(\frac{1}{2^{2^E-1}} \right) \tag{4}$$

The binary variable $y_e(t)$, which is output of the first A/D conversion step is being held locked by the predictive circuitry in the FPGA as long as the input signal fluctuates within one single ADC-E quantization step, thus keeping the much of the system in its static state and allowing for the statistical properties of the measured signal to dominate.

If the ADC-E measurement produces a binary value that doesn't coincide with the currently held one (i.e., $y_e(t) \neq y_e(t\text{-}1)$), to get the correct mantissa, the FPGA based Control Unit (CU) resets the held value to the prediction based on the most recently acquired exponent and the ADC-M effectuates a new conversion to get the correct mantissa. Thus, in general, this system will require $1 < n < 2$ measurements on average. How close it gets to a single measurement per sample will depend on the statistical properties of the measured signal itself.

III. DESCRIPTION OF THE FPGA BASED CONCEPT SYSTEM

The FPGA based system design features two 10-bit Analog Devices Inc. A/D converters (AD9203), with the maximum sampling rate of 40MSPS, a Texas Instruments 10-bit, 100 MSPS D/A converter, and an Altera APEX EP20K200E-1X device as its main elements. Input voltage swing of measured signals has been limited to 1V peak to peak to keep it within the most linear portion of ADC's and DAC's characteristics.

The logic implemented within the FPGA provides following three functions: it implements the main sample acquisition control unit, it formats the output binary word and it implements the prediction algorithms for DAC settings. Figure 2 provides a flow chart of the sample acquisition state machine.

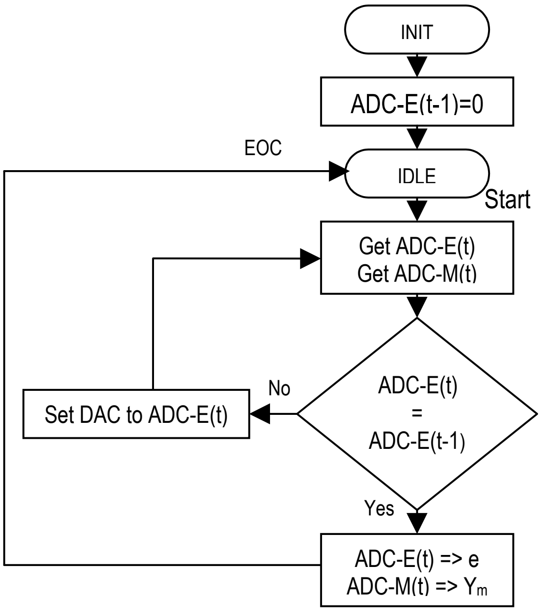

Figure 2: Sample Acquisition Flow Chart

Initially, the value of acquired exponent ADC-E can be set to any value. Statistical mean of the input signal distribution is a good starting value. This value is applied to the input of DAC. The two concurrent processes (to acquire exponent and to acquire mantissa) start the conversion synchronously with the *start* signal (clock pulse). In the first pass, design acquires in two parallel processes both exponent and mantissa as ADC-E(t) and ADC-M(t). It then proceeds to compare the exponent with the predicted value that is based on initially assumed, or the value acquired in a previous cycle, relying on the history of the acquired signal.

If the comparison of the newly acquired exponent value with the stored one produces equality, this means that the input signal in this time interval has maintained its absolute value within the single quantization step. Analog value produced by DAC and applied to the subtraction circuit was the correct one, and the value of the acquired mantissa ADC-M(t) is correct as well. This results in current ADC-E and ADC-M values being assigned to Y_e and Y_m respectively and signifies the end of cycle upon output formatting to IEEE floating point format in the FPGA.

In the case that newly acquired ADC-E does not compare with the currently held value, the input to DAC is updated with the value predicted on the basis of history of the signal and the newly acquired value and the end of first cycle is reached. The next cycle is then initiated with the following clock pulse (start signal) and the parallel exponent and mantissa acquisition process is repeated.

The correct acquisition is accomplished in one conversion cycle if the voltage at the input of the ADC-E is within single coarse quantization step from the previous sample, or in a series of two conversion cycles if a readjustment of the DAC setting is required. In either case, formatting of mantissa and exponent values to Y_e and Y_m signifies the end of sample conversion.

The mechanism presented here effectively increases the sample acquisition rate as compared to the classic two- cycle FP-ADC conversion scheme, for a factor that is dependant upon statistical properties of the measured signal. It also improves the resolution of the acquired mantissa, since only the expanded difference value is quantized for it. For the FPGA based system at hand, effective resolution obtained with this architecture, will be equal to resolution available from a hypothetical 20-bit linear ADC. This is because the first ADC-E breaks down the input signal range into 2^{10} segments, where each of them is further split into additional 2^{10} segments after amplification and expansion of the difference into full ADC-M acquisition range.

IV. PFP-ADC PREDICTION METHODS

The major factor that affects the prediction speed and accuracy is the complexity of the algorithm used. We have used in analysis here the polynomial regressive extrapolation. Given a discrete-time series $\{x_k = x(t_k), t_k - t_{k-1} = h, \text{k} = [0, \text{n}]\}$ the regressive difference equations are used to define the polynomial prediction equation:

$$x(t_n + t \bullet h) = (1 - \nabla)^{-t} \bullet x_n \qquad (5)$$

For $h = t_k - t_{k-1} = 1$ and $t_0 = 0$, the prediction is given by:

$$\hat{x}_{n+1} = x(n+1) = \left(\sum_{k=0}^{n} \nabla^k\right) x_n \qquad (6)$$

In reality we don't have access to the exact value samples (x_n), but rather to their quantized $\bar{x}$ values. Here we have the expressions of the predicted value given by this extrapolation algorithm for the first three simplest approximation equations.

$$\hat{x}_{n+1} = \begin{cases} \bar{x}_n & \text{if p = 0} \\ 2 \cdot \bar{x}_n - \bar{x}_{n-1} & \text{if p = 1} \\ 3 \cdot \bar{x}_n - 3 \cdot \bar{x}_{n-1} + \bar{x}_{n-2} & \text{if p = 2} \end{cases} \qquad (7)$$

The quantization Signal to Noise Ratio is defined as:

$$SNR = \frac{M(x^2)}{M\left[\left(x - \bar{x}\right)^2\right]} \qquad (8)$$

Then the prediction error (σ_{ep}) in terms of the signal standard deviation (σ_X) and the autocorrelation factors (ρ_1, ρ_2 and ρ_3) will be:

$$\begin{cases} 2 \cdot \sigma_X^2 \cdot \left(1 - \rho_1 + \frac{1}{2 \cdot SNR}\right) & \text{if p = 0} \\ 6 \cdot \sigma_X^2 \cdot \left(1 - \frac{4}{3} \cdot \rho_1 + \frac{1}{3} \cdot \rho_2 + \frac{1}{6 \cdot SNR}\right) & \text{if p = 1} \\ 20\sigma_X^2 \left(1 - \frac{15}{10}\rho_1 + \frac{6}{10}\rho_2 - \frac{11}{10}\rho_3 + \frac{1}{20 \cdot SNR}\right) & \text{if p = 2} \end{cases} \qquad (9)$$

This prediction method operates efficiently for any LTI class that is characterized by autocorrelation factors that drive the parenthesis of the above equations towards zero. If SNR is large enough to neglect the terms that contain it, then the autocorrelation domains for the ideal case of zero prediction error can be estimated (see Figures 3 and 4).

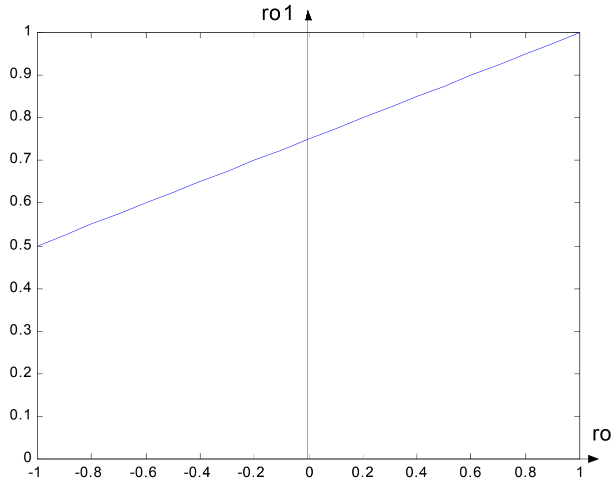

Fig.3 Curve of zero prediction error for p=1

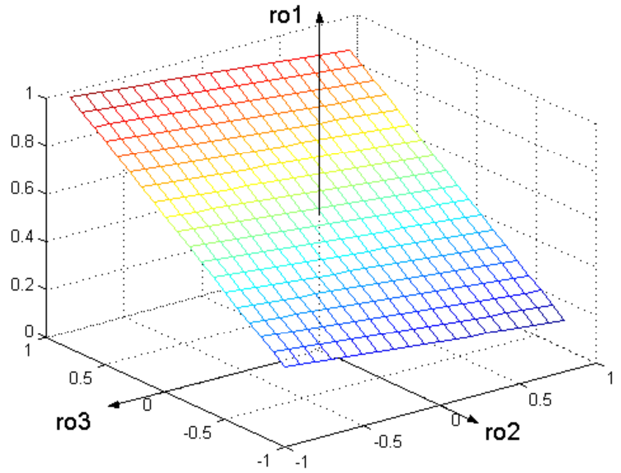

Fig.4 Plane of zero prediction error for p=2

IV. CONCLUSIONS

In order to fully utilize the remarkable speed of Flash ADC's combined with silicon real-estate savings, while preserving the best possible resolution and minimizing the quantization errors, a new Floating Point ADC architecture with predictive exponent determination was presented. The paper presents a concept circuit that implements this structure, along with the FPGA based proof-of-concept prototype, which will be further refined for monolithic implementation in a mixed-signal ASIC. Further research is planned with the specific ASIC implementation details in mind.

REFERENCES

[1] F. Maloberti, "High-speed data converters for communication systems," *IEEE Circuits Systems I*, vol.1, pp. 26–36, Jan. 2001.

[2] G. Haller and D. R. Freytag, "Analog floating-point BiCMOS sampling chip and architecture of the BaBar CsI calorimeter front-end electronics system at the SLAC B-factory," *IEEE Trans. Nucl. Sci.*, pt. 2, vol. 43, pp. 1610–1614, June 1996.

[3] D. U. Thompson, B. A. Wooley*,* "A 15-b Pipelined CMOS Floating-Point A/D Converter," *IEEE Journ. Solid-State Circuits*, Vol.36, No. 2, pp.299-303, 2001

[4] L. Grisoni, et all, "Implementation of a micro power 15-bit 'floating-point' A/D converter," in *Int. Symp. Low Power Electron. Design, 1996*, pp. 247–252.

[5] J. Piper, J. Yuan "Realization of a Floating-Point A/D Converter," *IEEE International Conference on Circuits and Systems (ISCAS)*, Sydney, May 2001

[6] V. Groza "High Resolution Floating Point Analog-to-Digital Converter," *IEEE Transactions on Instrumentation and Measurement*, Vol. 50, No. 6, pp. 1822-1829, December 2001