

Biquad Implementation of an IIR Filter for IQ Mismatch Correction in an SoC RF Receiver

Karen M.G.V. Gettings, Andrew K. Bolstad, Michael N. Ericson and Xiao Wang

MIT Lincoln Laboratory
Lexington, Massachusetts
{Karen.Gettings, Andrew.Bolstad, Michael.Ericson, Xiao.Wang}@ll.mit.edu

Abstract—This paper presents an IQ mismatch correction design and implementation that is part of a system-on-chip (SoC) that also includes a homodyne RF receiver and a sparse non-linear equalizer. It uses IIR filters to help the RF receiver achieve greater than an 80 dB image rejection ratio. The IIR filters are implemented using biquad structures to minimize power consumption by limiting the number of bits used per tap. The design was implemented in 65 nm CMOS technology and it is estimated to have a power performance of 150 GOPS per watt.

Keywords—IQ mismatch correction; biquad; low power

I. INTRODUCTION

System-on-chip (SoC) implementations are attractive solutions for size, weight, and power-restricted applications, such as mobile devices and unmanned aerial vehicles. The SoC pictured in Fig. 1 is a low-power homodyne receiver that aims for a spur-free dynamic range of more than 80 dB and a power consumption of less than 1 watt. The receiver includes both I and Q channels to double its effective bandwidth.

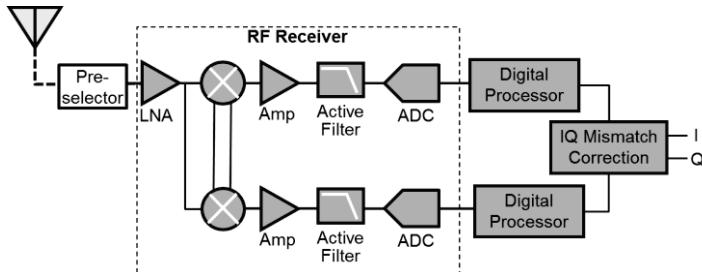


Fig. 1: System on chip block diagram.

RF Receivers that aim for high (>80 dB) image response rejection ratio (IMRR) must minimize IQ imbalances in the system. Because IMRR is sensitive to IQ imbalance, size, weight, and power-restricted applications must address the issue of correcting IQ imbalance in an efficient and low-power manner.

On the analog side of this SoC, each channel includes a mixer, an amplifier, an active anti-aliasing filter [1], and a 14-bit ADC. On the digital side, it includes a low-power sparse non-linear equalization engine [2], called SPEq (Sparse Polynomial Equalization), to achieve an SFDR of over 90 dB per channel. SPEq yields significant power savings by

increasing linearity in the digital domain instead of trying to build a highly linear analog front-end. Furthermore, because the analog front-end is co-designed with the digital equalizer, the digital side more efficiently compensates the analog side.

The IQ mismatch correction engine follows a similar design methodology as the digital equalizer in the sense that it is also co-designed with the front-end to achieve an IMRR of >80 dB while minimizing power consumption. To do this, we use a design similar to what is described in [3]. Fig. 2 pictures a top-level view of our architecture, where I' and Q' are the uncompensated channels, and I and Q are the compensated channels.

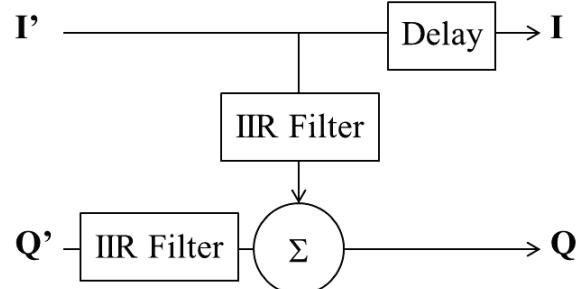


Fig. 2: IQ Mismatch Compensation Architecture

In the following section, we give a primer of the mathematics behind IQ mismatch compensation and the reasons why we chose IIR filters for our implementation. In the subsequent section, we present details of the three primary features of our implementation: parallel biquad filters, power-optimized multipliers, and fixed-point arithmetic. Finally, we provide results of the solution that is integrated into our 65 nm SoC design.

II. IIR FILTER FOR IQ MISMATCH CORRECTION

Mismatches in the amplitude and phase responses of the I and Q receiver channels combined with those of the mixer lead to degradation of IMRR (see [4], [5]). Without loss of generality, mismatches can be assumed to occur only in the Q channel. Under this assumption, the I and Q local oscillator (LO) signals can be written as $\cos(\omega_{LO}t)$ and $\sin(\omega_{LO}t + \theta)$, where θ represents phase mismatch. The frequency response of the baseband I channel is assumed to be that of an ideal

* This work is sponsored by the Assistant Secretary of Defense for Research & Engineering under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

lowpass filter, while the frequency response of the Q channel is a non-ideal lowpass filter which models frequency dependent amplitude and phase mismatches (including amplitude mismatch in the LOs).

Consider the response of the receiver to a tone at frequency $\omega_{LO} + \omega$:

$$\begin{aligned}x_I^+(t) &= h_I * (2 \cos((\omega_{LO} + \omega)t) \cos(\omega_{LO}t)) \\&= \cos(\omega t) \\x_Q^+(t) &= -h_Q(t) * (2 \cos((\omega_{LO} + \omega)t) \sin(\omega_{LO}t + \theta)) \\&= |H_Q(\omega)| \sin(\omega t - \theta + \angle H_Q(\omega))\end{aligned}$$

We have assumed no nonlinear distortion in the receiver. Similarly, the response to a tone at frequency $\omega_{LO} - \omega$ is:

$$\begin{aligned}x_I^-(t) &= h_I(t) * (2 \cos((\omega_{LO} - \omega)t) \cos(\omega_{LO}t)) \\&= \cos(\omega t) \\x_Q^-(t) &= -h_Q(t) * (2 \cos((\omega_{LO} - \omega)t) \sin(\omega_{LO}t + \theta)) \\&= -|H_Q(\omega)| \sin(\omega t + \theta + \angle H_Q(\omega))\end{aligned}$$

In the case of no mismatch, $H_Q(\omega) = 1$ and $\theta = 0$. Thus, the complex received signals would be:

$$\begin{aligned}x^+(t) &= x_I^+(t) + jx_Q^+(t) = e^{j\omega t} \\x^-(t) &= x_I^-(t) + jx_Q^-(t) = e^{-j\omega t}\end{aligned}$$

Under the assumption that mismatches occur in the Q channel, the goal of digital IQ correction is to modify the Q channel output such that all mismatches are removed. To this end, the Q channel output is replaced by the sum of filtered Q channel output and filtered I channel output:

$$\begin{aligned}\tilde{x}_I(t) &= x_I(t) \\ \tilde{x}_Q(t) &= g_I(t) * x_I(t) + g_Q(t) * x_Q(t)\end{aligned}$$

The desired filter responses of g_I and g_Q can be found via:

$$\begin{aligned}\begin{bmatrix}X_I^+(\omega) \\ X_I^-(\omega)\end{bmatrix} &= -j \begin{bmatrix}X_I^+(\omega) & X_Q^+(\omega) \\ -X_I^-(\omega) & -X_Q^-(\omega)\end{bmatrix} \begin{bmatrix}G_I(\omega) \\ G_Q(\omega)\end{bmatrix} \\&= -j \begin{bmatrix}\tilde{X}_Q^+(\omega) \\ -\tilde{X}_Q^-(\omega)\end{bmatrix}\end{aligned}$$

Given the desired frequency responses of filters G_I and G_Q , digital hardware must implement these filters as either finite impulse response (FIR) or infinite impulse response (IIR) filters. IIR filters are capable of modeling nonlinear phase responses and typically require fewer filter taps than FIR filters; however, care must be taken to avoid unstable IIR filters, particularly when finite precision arithmetic is used. We chose to implement IIR filters for IQ compensation based on the analysis of IQ mismatch in our test chips and the goal to minimize power consumption.

The RF front end is characterized by obtaining its frequency responses for I and Q channels through a frequency sweep. This is initially done without digital compensation and then analyzed in Matlab, where we determine the number of effective IIR taps needed and calculate the coefficients to achieve the desired compensation. We call this stage “training.”

Our approach to IQ compensation makes no assumptions on the input signal. As a result, we rely on training signals for calibration before deployment. Alternatively, adaptive compensators may be used provided the input signal satisfies certain properties. For example, the adaptive compensator of [6] relies on proper input signals, i.e. signals whose complementary autocorrelation function is equal to zero everywhere. While many communications signals are proper, real-valued waveforms such as binary phase-shift keying are not. The use of training signals for calibration allows us to operate on a wider class of input signals.

III. IIR FILTER IMPLEMENTATION

After choosing IIR filters to compensate the IQ mismatch, we proceeded with designing a direct transpose version of an IIR filter, as shown in Fig. 3.

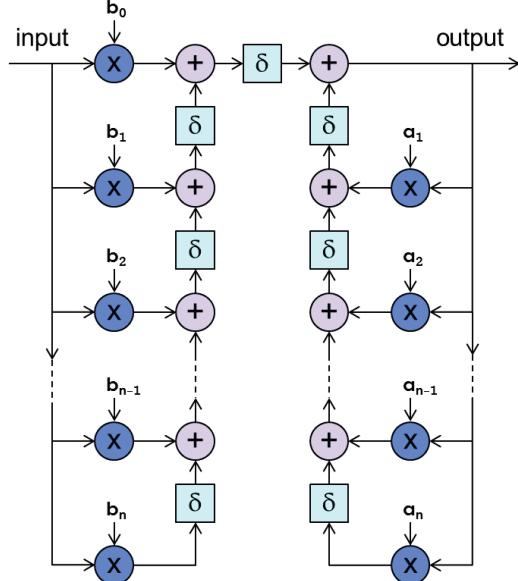


Fig. 3: Direct Transpose IIR Filter

The direct transpose implementation of the IIR filter was quite challenging for our design given the large number of taps we needed for appropriate IQ mismatch correction, which were calculated to be 24 feedforward and feedback taps when analyzing the front end data through Matlab. As the order of the IIR filter increases, so does the complexity of the implementation, as it increases range and precision demands. Twenty-four backward taps imply having a very large fanout in the feedback loop, which made the static timing requirements difficult to meet. In an attempt to meet static timing requirements with this large fanout, the resulting power

consumption estimate was higher than targeted. Furthermore, and most importantly, so many taps in an IIR filter make it difficult to keep the filter stable, as described in [7], because the system would need a large number of bits for stability and precision. This is counter to our goal of minimizing the number of bits used in our system to maximize power savings.

A. Parallel Biquad Filter vs. Direct Transpose IIR Filter

A parallel biquad filter is an attractive form of hardware implementation for IIR filters [8] because its structure limits the range of feedback coefficients to be from -2 to 2. It also minimizes the fanout in the critical static timing verification path to only 2 taps instead of the full set of taps as previously shown in the direct transpose IIR filter diagram. A typical biquad structure is pictured in Fig. 4.

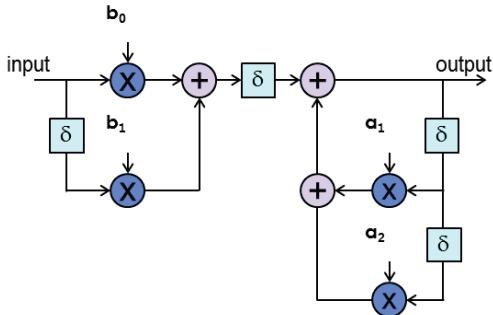


Fig. 4: Biquad Structure

To implement the 24 taps needed for each IIR filter, we included 12 biquad structures in parallel, as partially represented in Fig. 5. Parallelizing them allowed us to limit the latency through the system and save power, as then we do not need to pipeline the I channel as much to match the delay of the corrected Q channel (see delay block in figure Fig. 2). Each biquad can be turned on and off independently, depending on the number of taps needed during training.

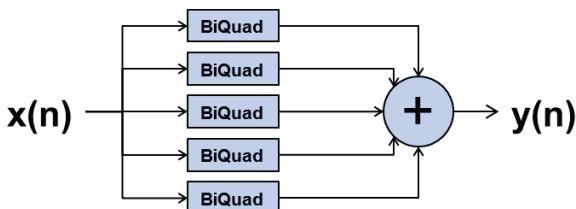


Fig. 5: Parallel Biquad Structure

To determine the amount of precision needed in the system, we worked backwards from the 16-bit 2's complement output of the IQ compensation, which we wanted to match closely to the 16-bit output from Matlab when rounding its floating point values.

When adding a total of 24 biquad outputs together, the aggregated output may have up to 4 bits of rounding error (4 bits can represent all 24 potential rounding errors of up to 0.5). To avoid this issue, we need to save at least 4 additional internal bits to represent the fractional value of each biquad

output. Fig. 6 shows how, for the regular biquad structure, we keep 8 bits of precision for the fractional portion of the output and 28 bits for the integer, for a total of 36 bits per output of each biquad. By keeping 8 fractional bits, we reduce the chance that the integer output is different than the rounded floating point result to be less than 1%, i.e., the 8-bit fractional value rounds in the same direction as the Matlab floating point value more than 99% of the time.

We keep more than 16 bits for the integer portion of the result to allow the combining of larger values, which may be positive or negative, to create the final 16-bit result. If the final combined value is greater than or equal to 2^{15} , the 16-bit output clips to $2^{15} - 1$; if the final combined value is less than -2^{15} , the 16-bit output clips to -2^{15} .

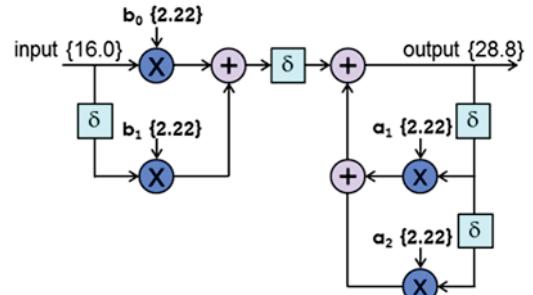


Fig. 6: Coefficients Biquad Structure

Two bits are sufficient for the integer parts of the feedback coefficients since they are limited to -2 to 2 on the feedback side. However, coefficients on the feedforward side are not restricted; in fact, simulations of our system show that the integer parts of the feedforward coefficients for some of the filter taps are comparatively very large. To fit this requirement, each IIR filter includes 4 “super-biquad” structures. These structures, as shown in Fig. 7 keep 24 bits of precision for the feedback coefficients and 32 bits for the feedforward coefficients while providing 44 bits on the output. The structures consume more power than the regular biquad structures because of the increased number of bits, but, because the feedforward side has a shorter critical path, we are able to use the low power multipliers that are described in the next subsection.

To help save power, each tap of each biquad in our implementation can be turned off independently if it is not needed to achieve the desired compensation that is calculated during training.

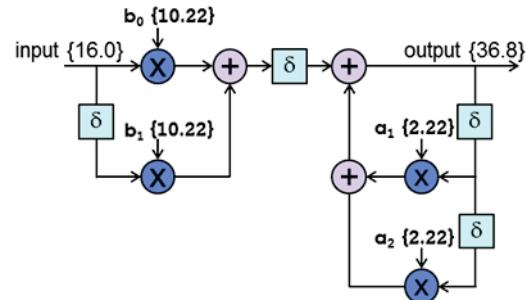


Fig. 7: Super Biquad Structure

B. Power-Optimized Multipliers vs. Delay-Optimized Multipliers

We are able to take advantage of using structured, power-optimized multipliers because we have a moderate clock rate of 200 MHz. These are Baugh-Wooley two's complement multipliers with a configurable number of pipeline stages to minimize power consumption, loosely based on [9]. In order to use them everywhere in our design, we had to carefully examine all critical timing paths because these multipliers are slower than their higher-power, delay-optimized counterparts, which are commonly chosen by logic synthesis tools in order to more easily meet their timing requirements.

Power savings come from cascading structured low-power architectures of multipliers and accumulators when those operations happen within a single clock cycle. We found that the synthesis tool would choose faster, but also higher-power, architectures of these mathematical functions and, while it would still meet our timing requirements by doing this, it was unnecessary since a ripple-carry adder adds only a small delay to a structured Baugh-Wooley multiplier, which outputs the least significant bits before the most significant bits. Since our clock period was long enough to contain the propagation delays through Baugh-Wooley multipliers, which use less power than other multiplier architectures, we took advantage of the earlier availability of the least significant bits to also use ripple-carry adders, which use less power than other adder architectures.

C. Fixed Point Arithmetic vs. Floating Point or Block Floating Point

In order to minimize power consumption, our design is fully fixed point. We can represent fractions with fixed-point arithmetic without incurring 3–4x area and 2–3x delay penalties of using floating-point [10]. The delays translate to higher power consumption to meet the desired timing target. The drawback of using fixed point is that the dynamic range is more limited than using floating point. We had to carefully select the precision needed in each stage of our filters to meet the desired dynamic range.

Block floating point arithmetic is an attractive solution for filters [11], in part because it is also power efficient. The problem with using block floating point in our design is that within the same calculation we needed different precision from different stages of our filter. To make all the precision the same to use block floating point would have meant adding up to 43 bits to an operation, which would have cost significant power consumption.

IV. RESULTS

We implemented 2 IIR filters, each with 12 biquad structures, 4 of which are super-biquads and 8 of which are regular biquads. The filters were implemented in IBM 65 nm CMOS technology, and estimated to consume approximately 450 mW of power after place and route, or roughly a quarter of a watt per 24-tap feedforward and feedback filter. The power estimation is from the Cadence Encounter place and route tool.

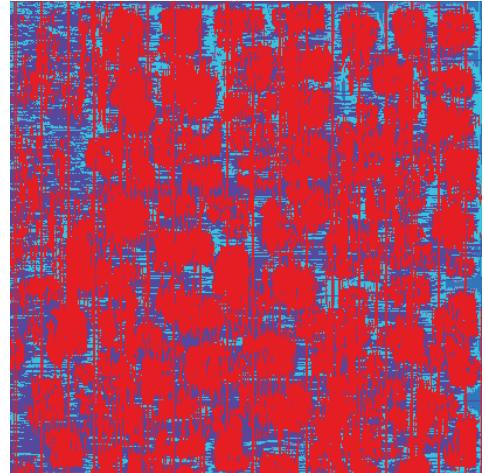


Fig. 8: IQ Compensation Implementation in 65 nm CMOS

The filter processes 200 million samples per second (MSPS), and, with 7 operations per biquad and 24 operations to combine the biquad outputs and round the final result, the IQ mismatch correction circuit has a performance of approximately 150 GOPS/Watt (billion operations per second per watt).

The layout of the IQ compensation architecture is presented in Fig. 8, with a size of roughly 6 mm².

Matlab simulations show that, with the 24 tap filters, our front end should see an image rejection ratio of up to 85 dB, as shown in Fig. 9. This figure shows a Monte Carlo simulation of the RF front end with a local oscillator (LO) frequency of 3.5 GHz, whose output has been compensated with the Matlab model of our IQ compensation scheme with equal numbers of taps in the feed-forward and feed-back paths. The results show good equalization results up to 35 taps, after which degradation occurs because there is not enough data to fit the taps. The best improvement in the simulation is obtained with 5 taps; however, this is a single Monte Carlo simulation, and we added up to 24 taps to cover potential discrepancies between the simulated and the measured data of the physical circuit. Each tap can be independently turned off to save on power consumption depending on the needs of the actual chip.

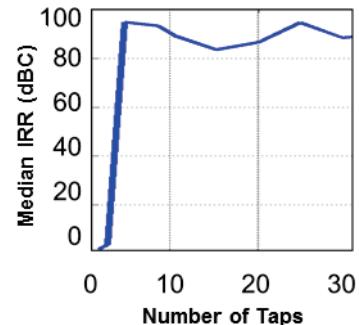


Fig. 9: Image Rejection Ratio per Number of Taps for IQ Mismatch Correction Circuit.

The chip has been taped out and is currently being fabricated.

V. CONCLUSIONS

We implemented an IQ correction circuit capable of improving the IMRR of a wideband RF front-end to 85 dB while consuming estimated power of less than half a watt. To minimize power consumption, we used biquad structures to limit the precision needed for implementing a high order filter with fixed point arithmetic. The architecture was taped out in IBM 65 nm technology and future work includes testing the filter and characterizing it with the analog front end that feeds it in our power-optimized SoC.

ACKNOWLEDGMENTS

The authors thank the analog design team led by H. Kim, who helped co-design the IQ compensation architecture. We also thank B. Miller for his valuable input on choosing an IQ mismatch correction architecture.

REFERENCES

- [1] Kim, H.; Green, M.; Miller, B.; Bolstad, A.; Santiago, D., "An active filter achieving 43.6dBm OIP3," Radio Frequency Integrated Circuits Symposium (RFIC), 2011 IEEE , vol., no., pp.1,4, 5-7 June 2011.
- [2] Gettings, K.; Bolstad, A.; Chen, S.-Y.S.; Ericson, M.; Miller, B.A.; Vai, M., "Low Power Sparse Polynomial Equalizer (SPEQ) for Nonlinear Digital Compensation of an Active Anti-Alias Filter," Signal Processing Systems (SiPS), 2012 IEEE Workshop on , vol., no., pp.249,253, 17-19 Oct. 2012.
- [3] Inamori, M.; Bostamam, A.M.; Sanada, Y.; Minami, H., "IQ Imbalance Compensation Scheme in the Presence of Frequency Offset and Dynamic DC Offset for a Direct Conversion Receiver," Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th , vol., no., pp.1,5, 26-29 April 2009.
- [4] M. Mailand, R. Richter, and H.-J. Jentschel, "IQ-Imbalance and its compensation for non-ideal analog receivers comprising frequency-selective components," Advances in Radio Science, vol. 4, pp. 189—195, 2006.
- [5] M. Valkama, M. Renfors, and V. Koivunen, "Compensation of Frequency-Selective I/Q Imbalances in Wideband Receivers: Models and Algorithms," 3rd IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC '01), Taiwan, 42—45, 2001.
- [6] Anttila, L.; Valkama, M.; Renfors, M., "Circularity-Based I/Q Imbalance Compensation in Wideband Direct-Conversion Receivers," IEEE Transactions on Vehicular Technology, vol.57, no.4, pp.2099-2113, July 2008.
- [7] Malik, J.S.; Hemani, A., "On the design of Doppler filters for next generation radio channel simulators," Signals, Circuits and Systems (SCS), 2009 3rd International Conference on , vol., no., pp.1,6, 6-8 Nov. 2009.
- [8] Francis, M., "Infinite Impulse Response Filter Structures in Xilinx FPGAs," Xilinx White Paper, August 2009.
- [9] Lan-Da Van; Jin-Hao Tu, "Power-efficient pipelined reconfigurable fixed-width Baugh-Wooley multipliers," Computers, IEEE Transactions on , vol.58, no.10, pp.1346,1355, Oct. 2009.
- [10] Andrew Rushton, VHDL for Logic Synthesis, 2nd ed. West Sussex, England: John Wiley & Sons, Ltd, 1998.
- [11] Oppenheim, A.V., "Realization of digital filters using block-floating-point arithmetic," Audio and Electroacoustics, IEEE Transactions on , vol.18, no.2, pp.130,136, Jun 1970.