A low-complexity self-calibrating adaptive quadrature receiver.

Ediz Cetin[1]
Suleyman S. Demirsoy[2]
Izzet Kale[1,3]
Richard C.S. Morling[1]

[1] School of Informatics, University of Westminster
2 Altera European Technology Centre, United Kingdom
[3] Applied DSP and VLSI Research Centre, Eastern Mediterranean University

A Low-Complexity Self-Calibrating Adaptive Quadrature Receiver

Ediz Cetin[*], Suleyman S. Demirsoy[+], Izzet Kale[*§] and Richard C. S. Morling[*]

[*]*Applied DSP and VLSI Research Group, Department of Electronic Systems, University of Westminster, 115 New Cavendish Street, London, W1W 6UW, United Kingdom*

[+]*Altera European Technology Centre, United Kingdom*

[§]*Applied DSP and VLSI Research Centre, Eastern Mediterranean University, KKTC*

{*e.cetin, kalei, morling*}*@wmin.ac.uk and sdemirso@altera.com*

Abstract

In this paper digital part of a self-calibrating quadrature-receiver is described, containing a digital calibration-engine. The blind source-separation-based calibration-engine eliminates the RF-impairments in real-time hence improving the receiver's performance without the need for test/pilot tones, trimming or use of power-hungry discrete components. Furthermore, an efficient time-multiplexed calibration-engine architecture is proposed and implemented on an FPGA utilising a reduced-range multiplier structure. The use of reduced-range multipliers results in substantial reduction of area as well as power consumption without a compromise in performance when compared with an efficiently designed general purpose multiplier. The performance of the calibration-engine does not depend on the modulation format or the constellation size of the received signal; hence it can be easily integrated into the digital signal processing paths of any receiver

1. Introduction

In industry today a lot of effort is being spent in developing cost and power efficient, highly integrated transceivers for single and multi-carrier systems. One of the key aspects in implementing a radio transceiver is the RF front-end where filtering and frequency translation operations are carried out. *I*/*Q* signal processing is used to downconvert the RF signal to the baseband. However, analog implementations of *I*/*Q* signal processing is vulnerable to phase and gain mismatches between the *I* and *Q* branches of the receiver. This results in imperfect image rejection, which is not sufficient for communications applications leading to severe performance degradation. Given the large signal constellations of *M*-QAM/PSK, even modest *IQ*-imbalances results in detrimental performance degradation. Furthermore, the power consumption is a critical parameter in mobile battery-operated systems. Apart from design and user interface, parameters like operating and stand-by times have the main effect on the customer's choice in buying mobile devices.

In the past, transceiver designs have focused on careful circuit layout and matching in order to achieve maximum performance. However, these solutions were expensive, time consuming and power hungry hindering the levels of integration that could be achieved. Furthermore, for optimal system operation, wideband image-rejection is required for a broad range of operating temperatures, frequencies and signal power levels. This is not achievable using a one-time optimisation technique, such as trimming.

Both analog and digital methods for dealing with *IQ*-imbalances have been reported in the literature [1] – [4]. All of the reported digital approaches are software based and thus not suitable for direct hardware implementation. This paper sets out to design and implement the digital part of an self-calibrating quadrature receiver that is capable of compensating for the RF impairments on the fly during the normal operation of the receiver [5],[6]. Efficient low-complexity FPGA implementation is carried out utilising *Reduced-Range-Multipliers* (RRM) [7]. The use of RRMs reduces power and area consumption considerably when compared with an efficiently designed general purpose multiplier.

The paper is organized as follows: Section 2 outlines the sources and the influence of RF-impairments on the receiver's performance. Section 3

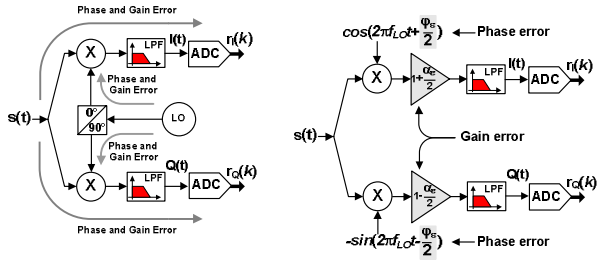
Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06)
0-7695-2614-4/06 $20.00 © 2006 IEEE

deals with the development of the unsupervised adaptive self-calibration algorithm. Section 4 deals with the low-power, efficient architectural design and implementation of the calibration-engine using time-multiplex architecture and RRM, while simulation results and performance analysis as well as area and delay comparisons are given in Section 5. Concluding remarks are given in Section 6.

2. RF-Impairments

Receiver architectures that utilize *IQ*-signal processing are vulnerable to mismatches between the *I* and *Q* channels. Sources of *IQ*-imbalances in the receiver are: the RF splitter used to divide the incoming RF signal equally between the *I* and *Q* paths which may introduce phase and gain differences as well as the differences in the length of the two RF paths can result in phase imbalance. The quadrature 90° phase-splitter used to generate the *I* and *Q Local-Oscillator* (LO) signals that drive the *I* and *Q* channel mixers may not be exactly 90°. Furthermore, there might be differences in conversion losses between the output ports of the *I* and *Q* channel mixers. In addition to these, filters and ADCs in the *I* and *Q* paths are not perfectly matched. The effects of these impairments on the receiver's performance can be detrimental. This section sets out to establish the influence of these impairments on the receiver's performance.

A model of a quadrature downconverter with the I/Q-phase and gain mismatch contributions by various stages is shown in Figure 1(a), whereas Figure 1(b) shows the analytical model used with all the phase and gain mismatches accumulated and represented by the erroneous LO signals.



(a) Theoretical Model (b) Analytical Model

Figure 1 RF Impairment contribution by various stages

The *IQ*-impairments can be characterized by two parameters: the amplitude mismatch, α_ε and the phase orthogonality mismatch, φ_ε between the *I* and *Q* branches. The amplitude-imbalance, β in decibels is obtained from the amplitude mismatch, α_ε as:

$$\beta = 20\log_{10}\left[(1+0.5\alpha_\varepsilon)/(1-0.5\alpha_\varepsilon)\right] \quad (1)$$

The following is an outline analysis to show the effects of RF-impairments on the receiver's performance. The incoming signal $s(t)$ can be expressed as:

$$s(t) = \Re\left\{u(t)e^{j2\pi f_{RF}t}\right\} \quad (2)$$

where $u(t)$ is the complex envelope of the received signal at f_{RF}. The erroneous complex LO signal, $x_{LO}(t)= I_{LO}+jQ_{LO}$, is given as:

$$x_{LO}(t) = e^{j2\pi f_{LO}t}(g_1 e^{j\frac{\varphi_\varepsilon}{2}} - g_2 e^{-j\frac{\varphi_\varepsilon}{2}}) + e^{-j2\pi f_{LO}t}(g_1 e^{-j\frac{\varphi_\varepsilon}{2}} + g_2 e^{j\frac{\varphi_\varepsilon}{2}}) \quad (3)$$

where $g_1=(1+0.5\alpha_\varepsilon)$, $g_2=(1-0.5\alpha_\varepsilon)$. The received signal $s(t)$ is quadrature mixed with the non-ideal LO signal, x_{LO}, and low-pass filtered resulting in received baseband signal $r_{IQ}(k)$. The complex baseband equation for the *IQ*-imbalance effects on the ideal received signal $r_{IQ}(k)$ is given as:

$$\begin{aligned} r_{IQ}(k) &= g_1[u_I(k)\cos(\varphi_\varepsilon/2) + u_Q(k)\sin(\varphi_\varepsilon/2)] \\ &\quad + jg_2[u_I(k)\sin(\varphi_\varepsilon/2) + u_Q(k)\cos(\varphi_\varepsilon/2)] \\ &= \tfrac{1}{2}[(2\cos\tfrac{\varphi_\varepsilon}{2} - j\alpha_\varepsilon\sin\tfrac{\varphi_\varepsilon}{2})u(t) + \\ &\quad (\alpha_\varepsilon\cos\tfrac{\varphi_\varepsilon}{2} + j2\sin\tfrac{\varphi_\varepsilon}{2})u^*(t)] \\ &= h_1 u(t) + h_2 u^*(t) \end{aligned} \quad (4)$$

where ( )* is the complex conjugate. To examine the performance of the quadrature mixer, we define the *Image-Rejection Ratio* (IRR) as the ratio between the image signal to desired signal (h_2/h_1). This as a function of phase and gain errors (α_ε, φ_ε) and is given in decibels as [5]:

$$IRR(\alpha_\varepsilon,\varphi_\varepsilon) = 10\log\left(\frac{2-2\cos\varphi_\varepsilon+0.5\alpha_\varepsilon^2(1+\cos\varphi_\varepsilon)}{2+2\cos\varphi_\varepsilon+0.5\alpha_\varepsilon^2(1-\cos\varphi_\varepsilon)}\right) \quad (5)$$

How the IRR varies with phase and gain errors is visually depicted in Figure 2. It can be observed that, in order to achieve an IRR of 60 dB, phase and gain errors must be 0.01 dB and 0.1° respectively, revealing very stringent, matching requirements. In practice, analog mismatches limit the IRR to 25 – 40 dB [8].

The effect of *IQ*-impairments on the constellation diagrams of 32-PSK and 256-QAM modulated signals with phase error of 15° and amplitude imbalance of 3 dB is depicted in Figure 3.

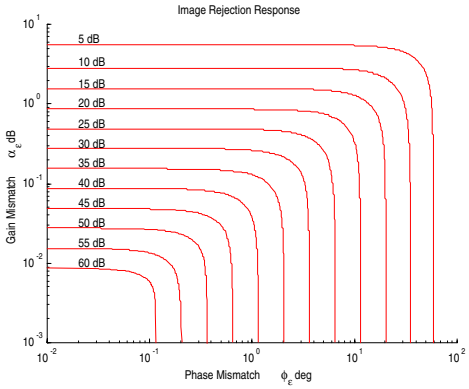
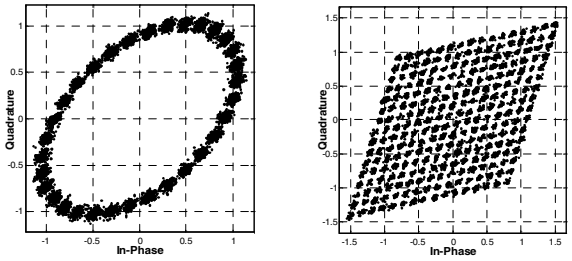


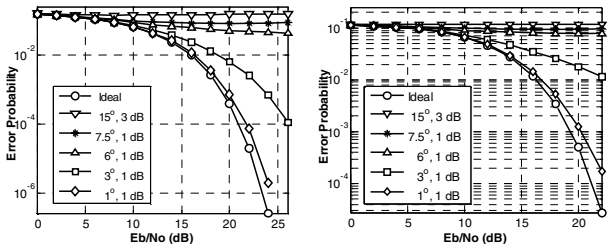
Figure 2 IRR as a function of phase and gain errors



(a) **(b)**

Figure 3 Constellation Diagrams for (a) 32-PSK and (b) 256-QAM - [φ_ε=15°, β=3 dB]

As shown in Figure 3, *IQ*-impairments rotate and scale the constellation points moving them outside and/or close to the borders of their respective detection regions. Hence, small amount of noise will suffice to move them into the wrong detection region hence increasing the *Bit-Error-Rate* (BER). Figure 4 demonstrates the effects of varying the *IQ* phase and gain mismatches on the raw BER performances of the systems using these 32-PSK and 256-QAM modulation formats.



(a) **(b)**

Figure 4 The effects of *IQ*-imbalances on BER of (a) 32-PSK and (b) 256-QAM modulated signals

As can be observed from Figure 4(a) and (b), *IQ*-imbalances degrade the systems BER performance greatly. This degradation in performance is surely not desirable and must be compensated. Section 3 outlines an digital self-calibrating adaptive algorithm developed for compensating for these impairments.

3. Self-Calibration

This section is a brief summary of [5] and [6] which introduces the *Blind-Source-Separation* (BSS) based adaptive calibration scheme. Our approach to the problem is to develop an adaptive BSS based *Calibration-Engine* (CE) that can operate at the baseband without pilot/test tones, by simply processing the received signals. The adaptive technique uses unsupervised/blind digital signal processing techniques to estimate nonlinearities and compensate for them in real-time during the normal operation of the receiver. The receiver can adapt its configuration to the environment, a major factor for maintaining performance under almost all circumstances.

The only assumption we make is that the *I* and *Q* components of the received signal, $r_I(k)$ and $r_Q(k)$, in the absence of impairments are orthogonal and not correlated witch each other.

Overall structure of the proposed approach is depicted in Figure 5, with *IQ*-imbalances modeled as unknown scalar mixing matrix.

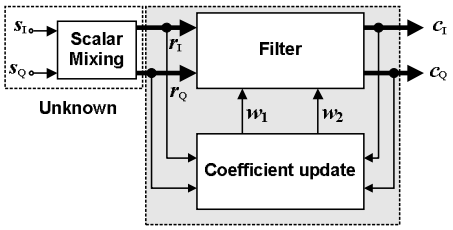


Figure 5 Structure for BSS based adaptive calibration-engine

In the proposed approach the filter block consists of 2-taps, w_1 and w_2. Output signals c_I and c_Q can be expressed as a function of transmitted signals s_I and s_Q as:

$$\begin{aligned} c_I(k) &= (1-w_1h_2)s_I(k)+(h_1-w_1)s_Q(k) \\ c_Q(z) &= (h_2-w_2)s_I(k)+(1-w_2h_1)s_Q(k) \end{aligned} \quad (6)$$

where h_1 and h_2 are given in (4). When the filters converge, i.e. w_1=h_1 and w_2=h_2 then the source estimates become:

$$\begin{aligned} c_I(k) &= (1-h_1h_2)s_I(k) \\ c_Q(k) &= (1-h_2h_1)s_Q(k) \end{aligned} \quad (7)$$

As can be observed from (7) the influence of the *IQ*-imbalances have been removed. However, both I and Q channels are scaled by $(1-h_1h_2) \approx 1$ which can be safely ignored. The coefficient update can be done with any algorithm depending on the desired performance. *Least-Mean-Square* (LMS) and *Recursive-Least-Squares* (RLS) algorithms being the most obvious ones resulting in different convergence speeds and computational complexities. The LMS [9] algorithm is used in this paper due to its low-complexity which makes it suitable for real-time systems and practical for integration into the receiver signal processing chains.

4. Architectural Design

In a portable battery powered device it is desirable to keep the size and power consumption as low as possible. Designing for lower power has become a critical pre-requisite for technical and commercial success. In Section 3 we have described an approach that improves the performance of the receiver with some hardware overhead. It is our aim in this section to reduce the hardware complexity and power-consumption of the calibration-engine as much as possible. Figure 6 depicts the overall processing structure for the BSS-based CE.

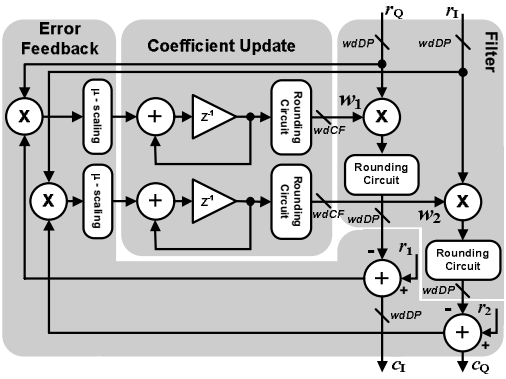


Figure 6 Overall structure for the calibration-engine

For our application the number of bits used to represent the data (r_I, r_Q), *wdDP*, is 16-bits two's complement. The number of bits used to represent the coefficients (w_1, w_2), *wdCF,* is 8-bits fractions. The value used for the LMS step-size is $\mu=2^{-13}$. This value was specifically chosen to be a power of two because it can be implemented in hardware as a simple right shift by 13 bits through hardwiring as opposed to an actual multiply hence eliminating the need for an extra multiplier. This is shown as μ-scaling block in Figure 6.

One strategy for implementing the CE is the direct mapping approach, where there is a one to one correspondence between the operations on the signal flow graph and operators in the final implementation. Such architecture is conceptually simple to implement, however, this is not the efficient way to implement digital algorithms. If the silicon area is not the main constrain but the speed is, then this parallel / direct-mapped approach is the obvious choice. On the other hand, if area is an important constrain and if very high-throughput is not the goal then time-multiplexed architectures are utilized in which multiple operations on a signal flow-graph can be mapped onto the same functional hardware unit. In our proposed method, since the entire signal processing operations associated with the CE takes place at the base-band not the IF or RF, the time-multiplexed approach operating at the base-band data rate is utilized. Figure 7 depicts the time-multiplexed architecture for implementing the CE.

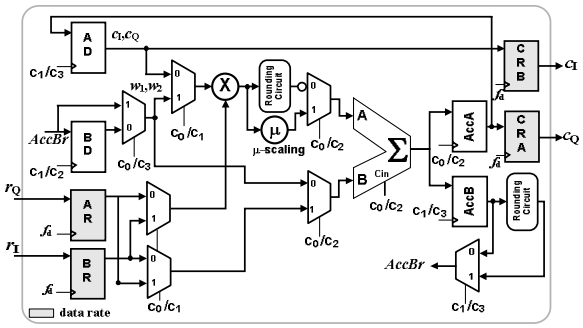


Figure 7 Time-multiplexed BSS based calibration-engine implementation

With this architecture a single 16x16 multiplier is utilized to calculate 8x16 (*wdCF*, *wdDP*) and 16x16 (*wdDP*, *wdDP*) multiplications required by the algorithm. The CE takes four cycles to compute c_I and c_Q. Since CE operates four times the data rate, it can operate in real-time producing outputs at the base-band data rate. Table 1 defines the register names whereas Table 2 depicts the schedule of the CE.

Table 1 Register Names

Name	Function
AD,BD	Multiplicand input registers
AR, BR	Multiplier input registers
AccA, AccB	Accumulator A and B registers for storing c_I, c_Q and w_1, w_2 values respectively.
CRA,CRB	c_I and c_Q output registers

Registers AR, AB, CRA and CRB operate at data rate and all remaining internal registers are clocked at four times the data rate.

Table 2 4-cycle processor schedule

Cycle	Multiplier o/p	Accumulator O/P
C0	$r_Q w_1$	$c_I = r_I - r_Q w_1$
C1	$c_1 r_Q \mu$	$w_1(k+1) = w_1(k) + c_1 r_Q \mu$
C2	$r_I w_2$	$c_Q = r_Q - r_I w_2$
C3	$c_Q r_I \mu$	$w_2(k+1) = w_2(k) + c_Q r_I \mu$

At the first half of CO cycle, $r_Q w_1$ is computed and $c_I = r_I - r_Q w_1$ is computed in the second half of C0. This is followed by w_1 computation for the next stage and c_Q for the current stage. This is followed by the computation of w_2 for the next stage in cycle C3 and so forth. Outputs are generated at the end of C0 and C2 cycles. These are fed into c_I and c_Q output registers CRA and CRB which are clocked at the data rate. As can be observed 100% utilization is achieved for the multiplier.

Further reductions in terms of the area and power can be made by considering the way we implement the 16x16 multiplier and the operating frequency of the CE. In order to implement 16x16 multiplication we can either make use of a dedicated 18x18 multiplier if available in the FPGA, or the multiplier needs to be implemented on the configurable logic cells. Furthermore, since one of the multiplications required is 8x16, and the CE operates at the baseband (relatively low-frequency), we can further split the 16x16 multiplication into two 8x16 multiplications. This will influence cycles C1 and C3 whereby, each of these cycles will now require two cycles to complete (C1a, C1b and C3a, C3b). Multiplication by 8x16 which takes place in Cycles C0 and C2 is not effected.

Figure 8 shows the improved CE architecture incorporating the improvements described above. The adder/subtractor, shift-by-8 and the *Partial-Product-Register* (PPR) associated with it after the multiplier handles the two-part multiplication by both positive and negative numbers. Depending on the *Most-Significant-Bit* (MSB) of the r_I or r_Q either addition or subtraction is performed. It must be pointed out that this add/subtract operation is only performed during cycles C1b and C3b where MSB-bits of the multiplication result are computed. The rest of the system is not affected and stays the same as before.

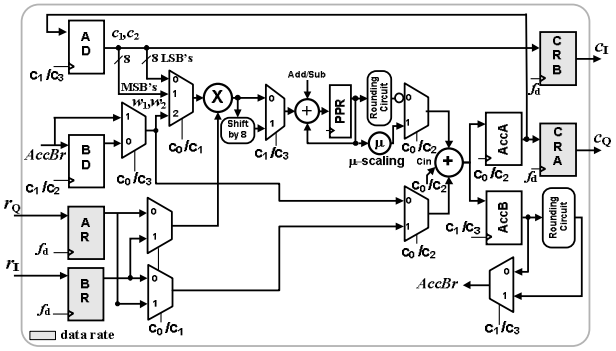


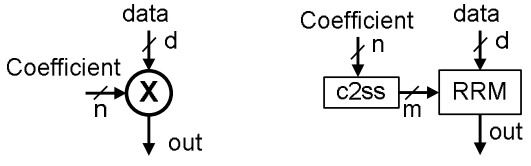
Figure 8 Improved time-multiplexed BSS based calibration-engine implementation

The total power per iteration of the LMS algorithm is determined by the power dissipation of shift, add, multiply, memory load, and memory store operations. Out of these, the power consumption of the multiplier is a key factor. Hence, if one can reduce the area and power consumption of the multiplier circuit the overall power and area consumption will be reduced greatly. This can be achieved by the use of clever multiplier structures. The hardware design strategy proposed in this paper for achieving this further reduction in area and power is the use of reduced-complexity multiplication through RRM [7].

4.1 Reduced Range Multipliers

The RRM has been developed in [7] to utilize the fixed resource environment of the FPGAs. It is implemented by making use of the reconfigurable arithmetic structures proposed in [10]. By utilizing these reconfigurable arithmetic structures to their full extend, it is possible to have reconfigurable multipliers that can replace *General Purpose Multipliers* (GPM) in adaptive filters.

Replacing a GPM with an RRM is straight forward process. This is depicted in Figure 9. The data input of the multiplier is common and unchanged in both approaches. However, the *n*-bit coefficient is then used to generate the select signals required by the configurable cells. The *Coefficient-to-Select-Signals* (c2ss) block consists of combinational logic and it handles not only the select line generation but also the mapping of the coefficient values.



(a) GPM **(b) RRM**

Figure 9 Replacing GPM with RRM

RRMs are particularly useful for adaptive filter implementations where not all parts of the dynamic range are needed for coefficient multiplications. Figure 10 shows dynamic ranges of several RRM for 8-bit coefficient values along with the dynamic range of a GPM. Areas represented by "■" correspond to the range coverage of the RRM.

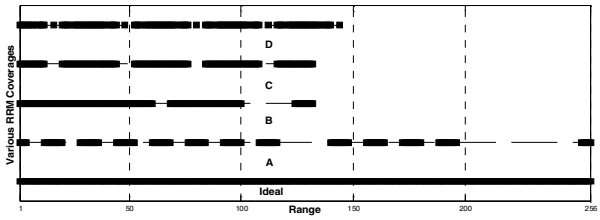


Figure 10 Dynamic ranges for various 8-bit RRM designs

As can be observed from Figure 10, there are various alternative coverage of coefficients that can be provided by the RRM, this property can be exploited in a variety of applications with specific requirements. Furthermore, it must be pointed out that hardware complexities of the RRM structures (A-D) are all the same. The RRM structure that was employed in our design is shown in Figure 11 with the dynamic range as displayed in Figure 10 (A).

The basic structure consists of four reconfigurable stages. The possible products out of each intermediate structure are shown in set brackets. The numbers '2', '8' and '16' next to the signals in the RRM diagram shows that those signals are left-shifted by 2-bits, 3-bits, and 4-bits respectively before they are connected to the next stage. S1 and S0 represent the select lines to choose one of the operations that are shown on the structures and are generated by the c2ss block.

The RRM block introduced in [7] to replace a GPM for 8x16 multiplication and shown in Figure 11, can be employed here as a low complexity solution. It was shown to reduce the multiplier area by 40% without significant performance loss. However, because it has a reduced range of coefficients, the 16x16 multiplier will also be a reduced range implementation. We can minimize the effect of using RRM by choosing a range that covers the 8-bit most regularly. For this reason, we chose to use option D in Figure 10.

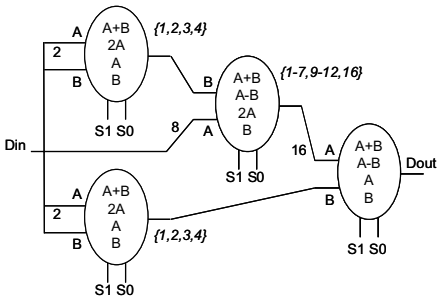


Figure 11 RRM structure used in our design

5. Simulation & Performance Comparison

This section contains detailed simulation results to compare the effects on the performance and area of using RRM instead of GPM in CE implementation. The performance measures used are IRR, *Modelling-Error* (ME) and BER. Furthermore, area in terms of *Look-Up-Tables* (LUT) are given. 256-QAM and 32-PSK modulated signals were used along with varying phase and gain mismatches. The communication channel was assumed to be AWGN.

In the first simulation study, both phase and gain errors are varied simultaneously in order to establish better understanding of the IRR performance. Phase error is varied from 0 to 30° and gain error is varied from 1 to 3 dB. Result of this is depicted in Figure 12 as a three-dimensional surface plot.

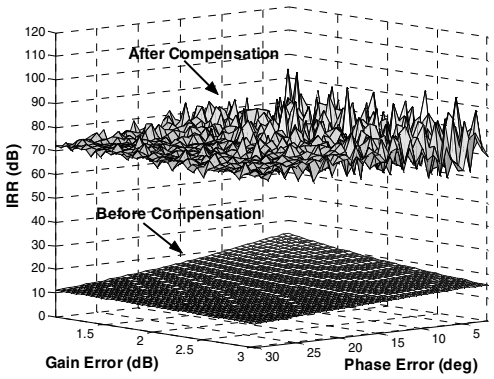
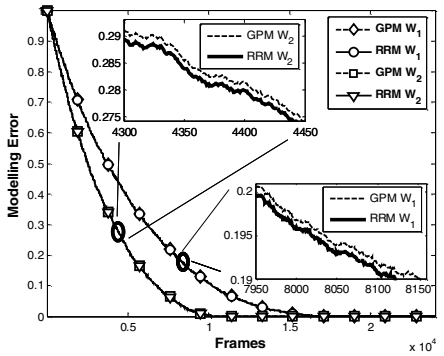


Figure 12 IRR before and after compensation for varying *IQ*-mismatches using RRM

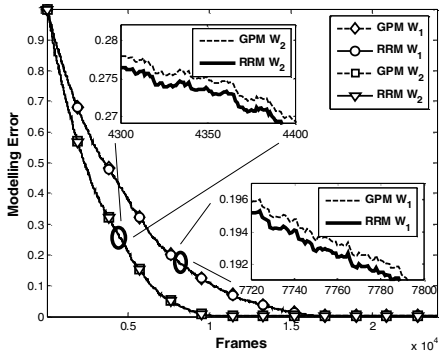
The ME [5] gives a global figure for the quality of the identification of the unknown mixing coefficients h_1 and h_2 by w_1 and w_2. Furthermore, it provides useful information about the convergence rate of the calibration-engine. ME is defined as the squared norm of the difference of the values between the original coefficients used in the scalar mixture and the

estimated coefficients, relative to the squared norm of the mixture coefficients.

ME plots are given in Figure 13. As can be observed the de-mixing coefficients w_1 and w_2 matches the mixing coefficients h_1 and h_2 as the ME approaches zero. Furthermore, we have zoomed in to certain parts of the ME plots to show how closely RRM follows the GPM for both w_1 and w_2. Depending on the RRM dynamic range, the GPM curve and RRM curve may differ at certain parts of the modelling error graphs.



(a)



(b)

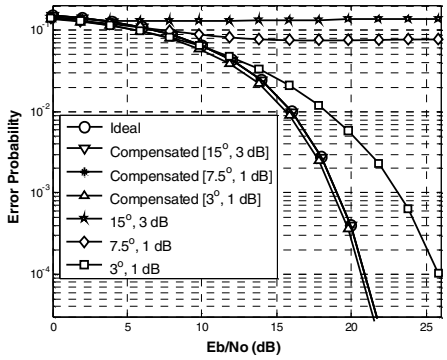
Figure 13 Modelling Errors (a) 32-PSK and (b) 256-QAM

Figure 14 on the other hand, depicts the BER before and after compensation for (a) 32-PSK and (b) 256-QAM with various phase and gain errors. As can be observed after compensation the BER closely matches the ideal case e.g. RF-impairments have been eliminated.

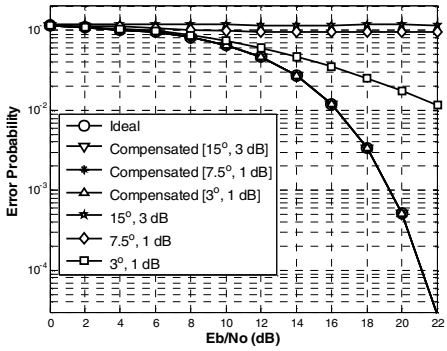
In terms of hardware real-estate, if we replace the GPM by the RRM, we save around 40% on the multiplier area as shown in Table 3 in terms of LUT count. Our design is implemented on a Xilinx Virtex FPGA. The synthesis was carried out using LeonardoSpectrum, for Virtex FPGA XV300BG432-5. The GPM was designed by the Core Generator from Xilinx. The critical path delay values are provided by the synthesizer and do not include the I/O buffer delays

Table 3 Area and delay comparison

	GPM (16x16)	RRM [A - D]
Multiplier Area	140 LUT	85 LUT
Delay	12.11 ns	10.91 ns



(a)



(b)

Figure 14 BER Curves before and after compensation for (a) 32-PSK and (b) 256-QAM

6. Concluding Remarks

In this paper we have presented an efficient FPGA implementation of a blind-source-separation based self-calibrating quadrature receiver using reduced range multipliers. Furthermore, a time-multiplexed architecture is designed and implemented. Use of RRM instead of GPM results in 40% reduction in the hardware complexity of the multiplier and subsequent reduction in power consumption. Through fixed-point simulations we have investigated the affects of replacing the GPM with RRM. Our results show that the RRM based design provides us with excellent performance.

The application of the adaptive self-calibrating architectures to wireless transceivers will eliminate the need for discrete off-chip components resulting in simpler, lower cost and low power transceivers with enhanced performance. These will subsequently manifest themselves in simpler RF front-ends and relaxed ADC analog circuit requirements and resulting in a major step towards true integration of low-power single-chip radio transceivers.

7. References

[1] Churchill F.E., Ogar G.W. and B.J. Thompson, "The Correction of I and Q Errors in a Coherent Processor", *IEEE Trans. on Aerospace and Electronic Systems*, vol. AES-17, no.1, pp. 131-137, January 1981.

[2] Lohtia, A., Goud, P. and C. Englefield, "An adaptive digital technique for compensating for analog quadrature modulator/demodulator impairments", *IEEE Pacific Rim Conference on Communications, Computers and Sig. Proc.*, vol. 2, pp. 447-50, 1993.

[3] Li Yu; Snelgrove W.M., "A novel adaptive mismatch cancellation system for quadrature IF radio receivers" *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46 issue 6, pp. 789 –801, June 1999.

[4] Valkama M., M. Renfors, and V. Koivunen, "Advanced methods for I/Q imbalance compensation in communication receivers," *IEEE Trans. Signal Processing*, vol. 49, pp. 2335–2344, Oct. 2001.

[5] Cetin, E., Kale, I. and R.C.S. Morling, "Adaptive Digital Receivers for Analog Front-end Mismatch Correction", *IEEE VTS 54th Vehicular Technology Conference (VTC 2001 Fall)*, vol: 4, pp. 2519 –2522, 2001.

[6] Cetin, E., I. Kale and R. C. S. Morling, "On The Structure, Convergence and Performance of an Adaptive I/Q Mismatch Corrector", *IEEE VTS 56th Vehicular Technology Conference (VTC 2002 Fall)*, vol: 4, pp. 2288 –2292, 2002.

[7] Cetin, E., S. S. Demirsoy, I. Kale and R.C.S. Morling, "Efficient FPGA Implementation of an Adaptive *IQ*-Imbalance Corrector for Communication Receivers Using Reduced Range Multipliers" *Proc. European Signal Processing Conference (EUSIPCO)*, September 2005.

[8] Chen, C. C., C. C. Huang, "On the architecture and performance of a hybrid image rejection receiver", *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1029 -1040, June 2001.

[9] Widrow B. and S.D. Stearns, "Adaptive Signal Processing", Prentice Hall, 1985 ISBN: 0-13-004029-0.

[10] Turner R. H., R. F. Woods, "Highly efficient, limited range multipliers for LUT-based FPGA architectures", *IEEE Trans. on VLSI Systems*, vol.12, no.10, pp. 1113-17, October 2004.