Design and low-power implementation of an adaptive image rejection receiver.

Ediz Cetin[1]
Sinan Topcu[2]
Izzet Kale[1]
Richard Morling[1]

[1] School of Informatics, University of Westminster
[2] STMicroelectronics, Istanbul, Turkey

Design and Low-power Implementation of an Adaptive Image Rejection Receiver

Ediz Cetin*, Sinan Topcu+, Izzet Kale* and Richard C. S. Morling*
*Applied DSP and VLSI Research Group, Department of Electronic, Communication and Software Engineering
University of Westminster, London, United Kingdom
+ STMicroelectronics, Istanbul, Turkey.
{e.cetin, kalei, morling}@wmin.ac.uk and sinan.topcu@st.com

***Abstract*— This paper deals with and details the design and implementation of a low-power; hardware-efficient adaptive self-calibrating image rejection receiver based on blind-source-separation that alleviates the RF analog front-end impairments. Hybrid strength-reduced and re-scheduled data-flow, low-power implementation of the adaptive self-calibration algorithm is developed and its efficiency is demonstrated through simulation case studies. A behavioral and structural model is developed in Matlab as well as a low-level architectural design in VHDL providing valuable test benches for the performance measures undertaken on the detailed algorithms and structures.**

I. INTRODUCTION

Image rejection receivers utilize In-phase and Quadrature (*I*/*Q*) signal processing in dealing with bandpass signals. However, analog implementations of *I*/*Q* signal processing is vulnerable to RF-impairments [1]-[11], resulting in imperfect image rejection, which is not sufficient for communications applications. With large signal constellations of M-QAM/PSK even modest RF-impairments result in detrimental performance degradation. Therefore, digital techniques which will enhance this image rejection and alleviate the *I* and *Q* channel mismatches play an important role in simplifying the analog front-ends for future high performance highly-integrated single-chip wireless transceivers.

Conventional image rejection architectures are implemented by analog circuit techniques [9]–[11]. However, hybrid and digital solutions have also been reported in the literature which attempts to improve IRR [1] - [8]. An unsupervised adaptive self-calibrating image rejection receiver was proposed and its performance evaluated in [8] utilising the *Digital Image Rejection Processor* (DIRP). This paper deals with efficient low-complexity, low-power implementation of this adaptive self-calibrating image rejection receiver. A key contribution of this paper is the application of the strength reduction transformation at the algorithmic level to obtain low-power implementation of the adaptive self-calibrating image rejection receiver. Furthermore, clever scheduling and pipelining of the algorithm for low-power implementation has been undertaken.

The paper is organized as follows: Section II gives a brief description of the adaptive image rejection receiver. Section III details the application of the strength reduction at the algorithmic level along with *Time-Division-Multiplexed* (TDM) architectural design, while concluding remarks are given in Section IV.

II. ADAPTIVE IMAGE REJECTION RECEIVER

The adaptive self-calibrating Image rejection receiver is composed of a modified Weaver image rejection mixer and a DIRP. With this architecture the I/Q errors are eliminated without using any off-chip discrete components, in the DSP domain at the baseband. Fig. 1 depicts the image rejection receiver incorporating the DIRP.

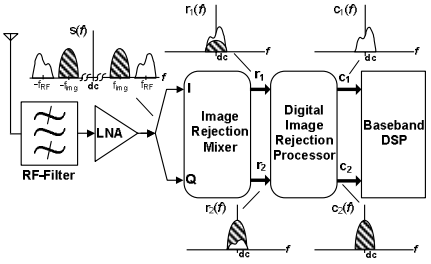

Figure 1. Image rejection receiver system with DIRP

The incoming signal, $s(t)$, consists of the wanted signal $u(t)$ at f_{RF} and unwanted image signal $i(t)$ at f_{IMG} where $f_{IMG} = f_{RF} - 2f_{IF}$. Hence, the incoming signal $s(t)$ can be expressed as:

$$s(t) = \Re\left\{ u(t)e^{j2\pi f_{RF}t} \right\} + \Re\left\{ i(t)e^{j2\pi f_{IMG}t} \right\} \qquad (1)$$

where $u(t)$ and $i(t)$ are the complex envelopes of the wanted and image signals respectively. The incoming signal is downconverted to an IF frequency via the image-rejection-mixer with RF-impairments. Signals are then digitised and digitally downconverted to the baseband to yield two baseband signals $r_1(k)$ and $r_2(k)$ which can be expressed as:

$$r_1(k) = u(t)(\overbrace{g_1 e^{-j\frac{\varphi_\varepsilon}{2}} + g_2 e^{j\frac{\varphi_\varepsilon}{2}}}^{h_1}) + i^*(t)(g_1 e^{j\frac{\varphi_\varepsilon}{2}} - g_2 e^{-j\frac{\varphi_\varepsilon}{2}})$$
$$r_2(k) = u(t)(\overbrace{g_1 e^{-j\frac{\varphi_\varepsilon}{2}} - g_2 e^{j\frac{\varphi_\varepsilon}{2}}}^{h_2}) + i^*(t)(g_1 e^{j\frac{\varphi_\varepsilon}{2}} - g_2 e^{-j\frac{\varphi_\varepsilon}{2}})] \quad (2)$$

where $g_1=(1+0.5\alpha_\varepsilon)$, $g_2=(1-0.5\alpha_\varepsilon)$ and φ_ε is the phase and α_ε is the gain mismatch between the I and Q channels. The desired signal corrupted by the image signal scaled by h_1 is contained in $r_1(k)$, and $r_2(k)$ contains the image signal corrupted by the desired signal scaled by h_2 due to the phase and gain errors. This is demonstrated in the frequency domain in Fig. 1. The mixing coefficients h_1 and h_2 can be expressed as:

$$h_1 = h_2 = (g_1 e^{j\frac{\varphi_\varepsilon}{2}} - g_2 e^{-j\frac{\varphi_\varepsilon}{2}}) \quad (3)$$

Signals $r_1(k)$ and $r_2(k)$ form the two inputs of the DIRP with $c_1(k)$ and $c_2(k)$ representing the corrected desired channel and the adjacent channel respectively. These can be expressed as [8]:

$$\begin{aligned} c_1(k) &= r_1(k) - \overbrace{r_2(k)w_1(k)}^{y_1} \\ c_2(k) &= r_2(k) - \overbrace{r_1(k)w_2(k)}^{y_2} \\ w_1(k+1) &= w_1(k) + 2\mu_1 c_1(k) r_2(k) \\ w_2(k+1) &= w_2(k) + 2\mu_2 c_2(k) r_1(k) \end{aligned} \quad (4)$$

The idea behind the DIRP is that in the absence of RF-impairments the desired and image signals are not correlated with each other. However, this is not the case when RF-impairments exist. The DIRP acts as a decorrelator separating the desired channel and the image channel. Detailed design and performance analysis of this is covered in [8].

III. Architectural Design

This section details the implement of the low-power, reduced complexity DIRP via the application of strength reduction as well as clever rescheduling of the algorithm along with efficient pipelining techniques. We start with the parallel brute force implementation of the DIRP, followed by the description and the application of the strength reduction transformation. This is followed by clever rescheduling of the DIRP algorithm for low-power and 100% resource utilisation and its pipelined implementation.

A. Algorithmic Level Power Reduction Techniques

Parallel brute force implementation of the DIRP is depicted in Fig. 2. Fig. 2(a) shows the filter section, whereas Fig. 2(b) details the adaptive weight-update section which makes up the DIRP [8]. Algorithmic transformations are an important class of architectural level transformations, which have been proposed for high speed and low-power [12]. These transformations rely on the fact that most linear DSP algorithms can be expressed in terms of multiply-add operations. In particular, the strength reduction transformation trades off high-complexity multiply operations with low-complexity add operations thus achieving low-power [12].

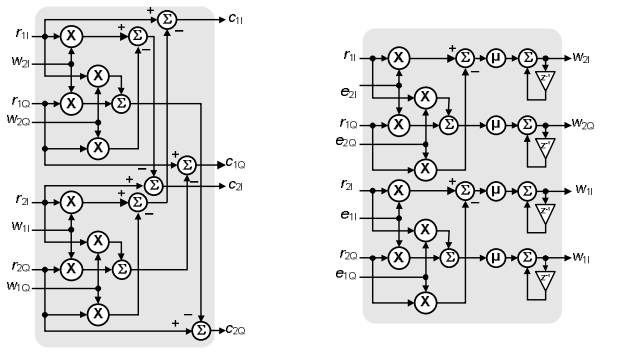

Figure 2. Structure of brute force implementation of DIRP (a)Filter and (b) weight-update sections

The algorithmic transformation of multiplying two complex numbers, $(a+jb)$ and $(c+jd)$ is given as:

$$(a+jb).(c+jd) = (ac-bd) + j(ad+bc) \quad (5)$$

As can be observed from (5) a total of four real multiplies and two real additions are needed for computing the complex multiplication. Equation (5) can be strength reduced and reformulated as:

$$\begin{aligned} (a-b).d + a.(c-d) &= ac - bd \\ (a-b).d + b.(c+d) &= ad + bc \end{aligned} \quad (6)$$

As can be observed from (6) the number of real multiplications is three and the number of real additions is five i.e. one multiplier is replaced with three adders. We will now apply the strength reduction technique to the DIRP algorithm. The outputs of the filter block of the DIRP are given as:

$$\begin{aligned} y_1(k) &= r_2(k) w_1(k) \\ y_2(k) &= r_1(k) w_2(k) \end{aligned} \quad (7)$$

where $w_{(1,2)}(k)=w_{I(1,2)}(k)+jw_{Q(1,2)}(k)$, and $r_{(1,2)}(k)=r_{I(1,2)}(k)+jr_{Q(1,2)}(k)$. Putting these into (7) we have:

$$\begin{aligned} y_1(k) &= w_1(k) r_2(k) \\ &= [w_{1I}(k) + jw_{1Q}(k)][r_{2I}(k) + jr_{2Q}(k)] \\ &= [w_{1I}(k) r_{2I}(k) + w_{1Q}(k) r_{2Q}(k)] + j[w_{1I}(k) r_{2Q}(k) - w_{1Q}(k) r_{2I}(k)] \\ y_2(k) &= [w_{2I}(k) + jw_{2Q}(k)][r_{1I}(k) + jr_{1Q}(k)] \\ &= [w_{2I}(k) r_{1I}(k) + w_{2Q}(k) r_{1Q}(k)] + j[w_{2I}(k) r_{1Q}(k) - w_{2Q}(k) r_{1I}(k)] \end{aligned} \quad (8)$$

The adaptive coefficient updates can be expressed as:

$$\begin{aligned} w_1(k+1) &= w_1(k) + 2\mu_1 c_1(k) r_2(k) \\ &= w_{1I}(k) + 2\mu_1[c_{1I}(k) r_{2I}(k) - c_{1Q}(k) r_{2Q}(k)] + \\ &\quad jw_{1Q}(k) + 2\mu_1[c_{1I}(k) r_{2Q}(k) + c_{1Q}(k) r_{2I}(k)] \\ w_2(k+1) &= w_2(k) + 2\mu_2 c_2(k) r_1(k) \\ &= w_{2I}(k) + 2\mu_1[c_{2I}(k) r_{1I}(k) - c_{2Q}(k) r_{1Q}(k)] + \\ &\quad jw_{2Q}(k) + 2\mu_1[c_{2I}(k) r_{1Q}(k) + c_{2Q}(k) r_{1I}(k)] \end{aligned} \quad (9)$$

At this stage we can apply the strength-reduction transformation to the filter output. For the filter output equation given in (7) the transformation that follows is (only $y_1(k)$ is shown to prevent repetition):

$$
\begin{aligned}
y_1(k) &= r_2(k)w_1(k) \\
&= \left[r_{2I}(k) + jr_{2Q}(k)\right]\left[w_{1I}(k) + jw_{1Q}(k)\right] \\
&= \left[y_A(k) + y_C(k)\right] + j\left[y_B(n) + y_C(n)\right]
\end{aligned} \tag{10}
$$

where:

$$
\begin{aligned}
y_A(k) &= \left[w_{1I}(k) + w_{1Q}(k)\right].r_{2I}(k) \\
y_B(k) &= \left[w_{1I}(k) - w_{1Q}(k)\right].r_{2Q}(k) \\
y_C(k) &= -w_{1Q}(k).\left[r_{2I}(k) - r_{2Q}(k)\right]
\end{aligned} \tag{11}
$$

For the DIRP case the strength reduced form of (7), following the derivations of equations (10) (11), is given by:

$$
y_{1,2}(n) = [c_A(k) + c_C(k)] + j[c_B(k) + c_C(k)] \tag{12}
$$

where:

$$
\begin{aligned}
c_A(k) &= w_{I(1,2)}(k).r_{I(1,2)} \\
c_B(k) &= w_{Q(1,2)}(k).r_{Q(1,2)} \\
c_C(k) &= -w_{Q(1,2)}(k).[r_{I(1,2)} - r_{Q(1,2)}]
\end{aligned} \tag{13}
$$

Following a similar approach and applying the strength reduction technique to (9) we end up:

$$
\begin{aligned}
&w_{(1,2)}(k+1) = w_{(1,2)}(k) + \mu[2c_{I(1,2)}(k)r_{Q(1,2)}(n) + \\
&[(c_{I(1,2)}(k) - c_{Q(1,2)}(k)).(r_{I(1,2)}(k) - r_{Q(1,2)}(k))] + \\
&j[2c_{Q(1,2)}(k)r_{I(1,2)}(k) + (c_{I(1,2)}(k) - c_{Q(1,2)}(k)). \\
&(r_{I(1,2)}(k) - r_{Q(1,2)}(k))]]
\end{aligned} \tag{14}
$$

From (12)-(14), we can now construct the structure of the strength reduced DIRP. Fig. 3 (a) depicts the "filter section" of the strength reduced DIRP, whereas Fig. 3(b) depicts the "weight-update section".

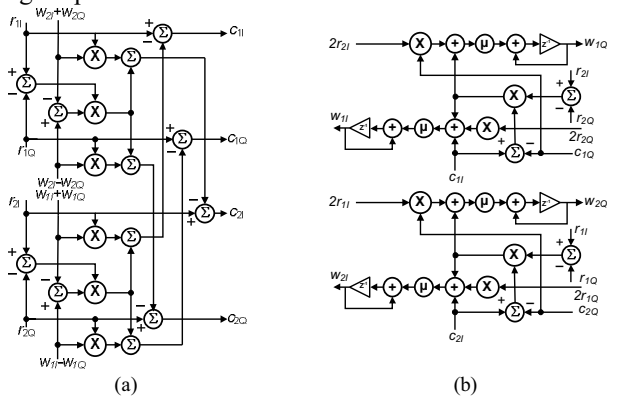

Figure 3. Structure of strength reduced DIRP (a)Filter and (b) weight-update sections

As can be observed from Figs. 2 and 3, application of the strength reduction technique has resulted in the reduction of number of multipliers at the expense of adders. This finding tabulated in Table I.

TABLE I. STRENGTH REDUCTION SAVINGS

	Number of multiplies	*Number of Adders*
Before	16	16
After	12	24

If we assume that effective capacitance of a two-operand multiplier is K_c times that of a two-operand adder [12], it can be seen that application of the strength reduction to the implementation of the DIRP results in a power saving factor, PS, given by:

$$
PS = \frac{P_{D,o} - P_{D,sr}}{P_{D,o}} = \frac{4K_c - 8}{16k_c + 16} \tag{15}
$$

where $P_{D,o}$ and $P_{D,sr}$ are the dynamic power dissipation of the original and strength-reduced DIRP algorithms. Fig. 4 depicts the PS as a function of K_c. As can be observed, the power saving can be made for K_c>2 for the DIRP application. Asymptotically, the power savings approach 25% as K_c increases. For a typical K_c value of 8 [12], the power saving is 16.67%.

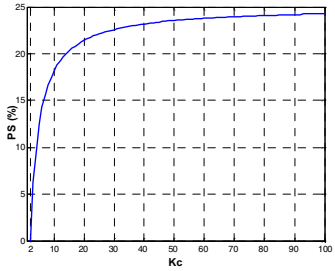

Figure 4. Power Saving as a function of Effective Capacitance(K_c)

B. Algorithmic Rescheduling and Pipelining for Low Power

The architectural design aims to have 100% utilisation of each element. This is achieved by clever use of process rescheduling and pipelining stages to incorporate the different sections of the design. In this section a detailed analysis of the architecture, process schedule and process cycles will be carried out and the most favourable architecture and data flow will be established. The TDM based architecture is preferred for implementation as it utilises the least hardware. The first step in designing the TDM based architecture is to decide on the data flow over the structure diagram. The overall structure consists of the repetitive use of three distinctive sub-structures. These are: *Complex Multiplication Block* (CMB), *Filter Output Block* (FOB), and the *Weight Update Block* (WUB). With the parallel implementation the data flow is straight forward. In the first clock cycle the filter outputs, c_1 and c_2, are calculated in a parallel manner. In the following clock cycle the filter outputs are fed back to the adaptive coefficient update section for the calculation of the weight factors ($w_1(k+1)$ and $w_2(k+1)$) of the next iteration. The TDM based hybrid model on the other hand has to follow a different data flow structure. The most suitable dataflow structure where each sub-block is utilized 100% at all time is shown in Fig. 5.

The data flow starts with the use of CMB, where inputs r_{1I} and r_{1Q} are multiplied with w_{2I} and w_{2Q} (Step 1). The next step is to use the calculated intermediate result in obtaining the outputs c_{2I} and c_{2Q} in the FOB (Step 2a). However, at this stage the CMB stands idle. So we utilize it by running this block at the same time with the FOB, (Step 2b). While c_{2I} and c_{2Q} outputs are calculated at the FOB, the multiplications of r_{2I} and r_{2Q} with w_{1I} and w_{1Q} are carried out in the CMB. When both CMB and FOB finish their operations the next parallel usage of these blocks starts. This is done as follows: calculated c_{2I} and c_{2Q} values from the FOB can be used in the CMB to be multiplied with r_{2I} and r_{2Q} inputs (Step 3a). At the same time the intermediate results from the previous use of the CMB can be used in the FOB to calculate c_{1I} and c_{1Q} (Step 3b). Once the multiplication operations in Step 3a are finished, the resulting values can be used in the WUB (Step

4a). While the μ-scaling and weight updates are being processed (Step 4a), the CMB can process the results from FOB from Step 3b (Step 4b). Once the calculations in WUB are finished in Step 4a, we have the weight factor of $w_2(k+1)$. We can use this new weight factor to begin the same cycle of operations stated so far by utilizing the CMB (Step 5a). This is possible since the multiplication block will complete its processing of Step 4b. At the same time the results of Step 4b can further be used to calculate $w_1(k+1)$ in the WUB (Step 5b). So in a total of five steps all the outputs will be calculated as well as the weight updates for the next iteration. Clever usage of the sub-blocks and the parallel processing scheme helped us to drop the number of steps required to calculate the outputs and weight factors from eight steps to five steps.

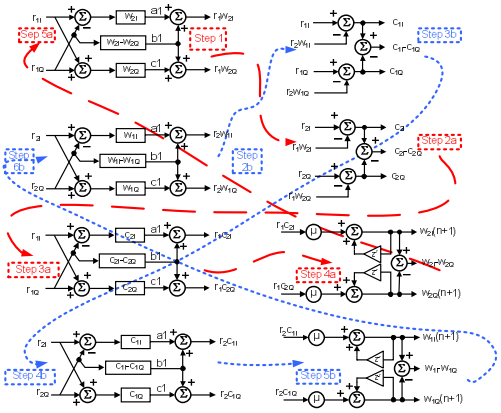

Figure 5. Most suitable data-flow

C. Architectural Design

According to the data flow proposed in the previous section the architecture design of the hybrid models is undertaken. Fig. 6 depicts the proposed architecture of the TDM based DIRP architecture.

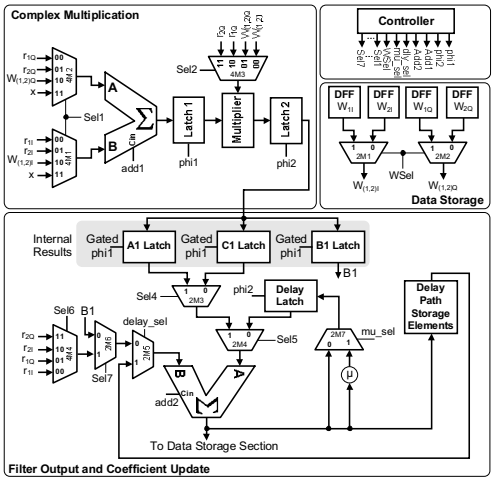

Figure 6. Hybrid DIRP Architecture

The architecture consists of four major parts: CMB, merged FOB and WUB section, Controller and the storage block where the calculated output and weight values are stored. The controller is implemented as a 4-bit ring counter and generates control signals sel1, … sel7, add1, mu_sel etc.

Performance of the proposed architecture was evaluated using 32-PSK modulated signals. Simulation results are shown in Fig.7 for varying phase and gain errors demonstrating the effective operation of the architecture.

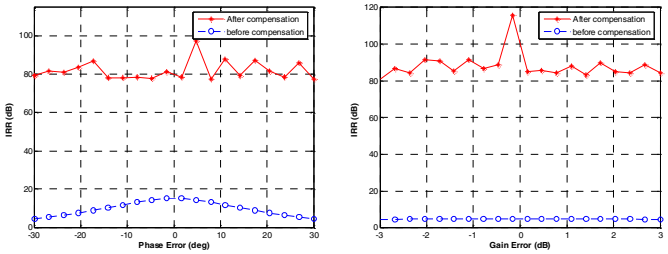

Figure 7. IRR performance for (a) varying phase error, (b) varying gain error

IV. Concluding Remarks

Design and implementation of a low-power image-rejection receiver incorporating DIRP to alleviate RF-impairments and improve IRR has been undertaken. Strength reduction, data re-scheduling and pipelining approaches were used to reduce the power consumption. It has been shown that the application of strength reduction at the algorithmic level results in a power saving of 16.67%. Complexity of the algorithm is reduced by four real multipliers at the expense of eight real adders. The algorithm is also amenable for software DSP implementation requiring small processing overhead.

References

[1] Cetin, E., Kale, I. and Morling, R.C.S., "Living and Dealing with RF Impairments in Communication Transceivers", IEEE International Symposium on Circuits and Systems, (ISCAS), pp. 21-24, May 2007

[2] F. Ehbahani, Y. Ishigami, J. Leete and A.A. Abidi, "CMOS mixers and polyphase filters for large image rejection" *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 873 -887, June 2001.

[3] L. Der, B. Razavi, "2-GHz CMOS image-reject receiver with LMS calibration" *IEEE Journal of Solid-State Circuits*, , vol. 38 pp. 167 -175, Feb. 2003.

[4] C. C. Chen, 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.

[5] 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.

[6] Windisch, M. and G. Fettweis, "Performance analysis for blind I/Q imbalance compensation in low-IF receivers", International Symp. on Control, Communications and Signal Processing, pp: 323 – 326, 2004

[7] Crols J. and M.J. Steyaert, "Low-IF Topology for High Performance Analog Front Ends of Fully Integrated Receivers", IEEE Transactions on Circuits and systems- II, vol. 45, no.3, pp. 269-282 March 1998.

[8] Cetin, E., I. Kale and R. C. S. Morling, "Adaptive Self-Calibrating Image Rejection Receiver", IEEE International Conference on Communications (ICC 2004), vol. 5, pp. 2731-2735, June 2004

[9] M. D. McDonald, "A 2.5GHz BiCMOS image-reject front-end," ISSCC Dig. of Tech. Papers, pp. 144–145, Feb. 1993.

[10] W. Baumberger, "A single-chip image rejecting receiver for the 2.44 GHz band using commercial GaAs-MESFET-technology" IEEE Jou. of Solid-State Circts., , vol. 29 issue. 10, pp. 1244 -1249, Oct. 1994.

[11] Maligeorgos, J.P.; Long, J.R., "A low-voltage 5.1-5.8-GHz image-reject receiver with wide dynamic range"; IEEE Journal of Solid-State Circuits, vol. 35, issue: 12 , pp. 1917 -1926, Dec 2000.

[12] Shanbhag, N. R., Goel, M., "Low-Power Adaptive Filter Architectures via Strength Reduction", Proc. Int. Symp. on Low power Electronics and Design, pp. 217-220, Aug. 1996.