POLITECNICO DI TORINO
Repository ISTITUZIONALE

Baseband analog front-end and digital back-end for reconfigurable multi-standard terminals

Availability:
This version is available at: 11583/1401934 since:

Publisher:

Published
DOI:

Terms of use:

Publisher copyright

(Article begins on next page)

14 June 2024

Baseband Analog Front-End and Digital Back-End for Reconfigurable Multi-standard Terminals

Andrea Baschirotto, *Senior Member, IEEE,* Fabio Campi, Rinaldo Castello, *Fellow, IEEE,*

Giovanni Cesura, Roberto Guerrieri, Luciano Lavagno, *Member, IEEE,* Andrea Lodi,

Piero Malcovati, *Senior Member, IEEE,* Mario Toma

Abstract

Multimedia applications are driving wireless network operators to add high-speed data services such as Edge (E-GPRS), WCDMA (UMTS) and WLAN (IEEE 802.11a,b,g) to the existing GSM network. This creates the need for multi-mode cellular handsets that support a wide range of communication standards, each with a different RF frequency, signal bandwidth, modulation scheme etc. This in turn generates several design challenges for the analog and digital building blocks of the physical layer. In addition to the above-mentioned protocols, mobile devices often include Bluetooth, GPS, FM-radio and TV services that can work concurrently with data and voice communication. Multi-mode, multi-band, and multi-standard mobile terminals must satisfy all these different requirements. Sharing and/or switching transceiver building blocks in these handsets is mandatory in order to extend battery life and/or reduce cost. Only adaptive circuits that are able to reconfigure themselves within the handover time can meet the design requirements of a single receiver or transmitter covering all the different standards while ensuring seamless inter-interoperability. This paper presents analog and digital base-band circuits that are able to support GSM (with Edge), WCDMA (UMTS), WLAN and Bluetooth using reconfigurable building blocks. The blocks can trade off power consumption for performance on the fly, depending on the standard to be supported and the required QoS (Quality of Service) level.

I. INTRODUCTION

The growing economic and social impact of mobile telecommunication devices, together with the evolution of protocols and interoperability requirements among different standards for voice and data, is currently driving

A. Baschirotto is with Department of Innovation Engineering, University of Lecce, Italy, andrea.baschirotto@unile.it

R. Castello is with, Department of Electronics, University of Pavia, Italy, rinaldo.castello@unipv.it

F. Campi, G. Cesura, and M. Toma are with STMicroelectronics, Italy, fabio.campi@st.com, giovanni.cesura@st.com, mario.toma@st.com

R. Guerrieri and A. Lodi are with Advanced Research Center on Electronic Systems, University of Bologna, Italy, rguerrieri@deis.unibo.it, andrea.lodi@deis.unibo.it

L. Lavagno is with Department of Electronics, Politecnico di Torino, Italy, luciano.lavagno@polito.it

P. Malcovati is with, Department of Electrical Engineering, University of Pavia, Italy, piero.malcovati@unipv.it

worldwide research towards the implementation of fully-integrated multi-standard transceivers. The most advanced fully integrated solutions in the scientific literature and on the market do not cover the four most important telecommunication standards, namely GSM, WCDMA, Bluetooth, and wireless LANs (WLANs). In order to allow the user to switch seamlessly among different standards, achieving so-called "global roaming", for both voice and data applications, all these standards have to be supported by an integrated transceiver. GSM and WCDMA (UMTS) are the dominant standards for voice and mixed voice/data mobile services, while WLANs based on the IEEE 802.11a/b/g protocol are the most important standards for high data-rate wireless internet access. Finally, Bluetooth enables the terminal to be wirelessly connected with other devices at low data rates over a short distance. Implementation of an integrated multi-standard transceiver that is competitive with solutions based on separate devices for the different standards must take various points into account. First of all, both silicon area and static power consumption must be minimized, thus requiring the maximum possible hardware sharing among the transceivers for the different standards. In order to reach this goal, one must define which standards can be used at the same time. We assumed that only two standards among the supported ones can operate concurrently at a given time (e.g WLAN with Bluetooth or voice with Bluetooth or voice with WLAN) and that no handover is supported for Bluetooth.
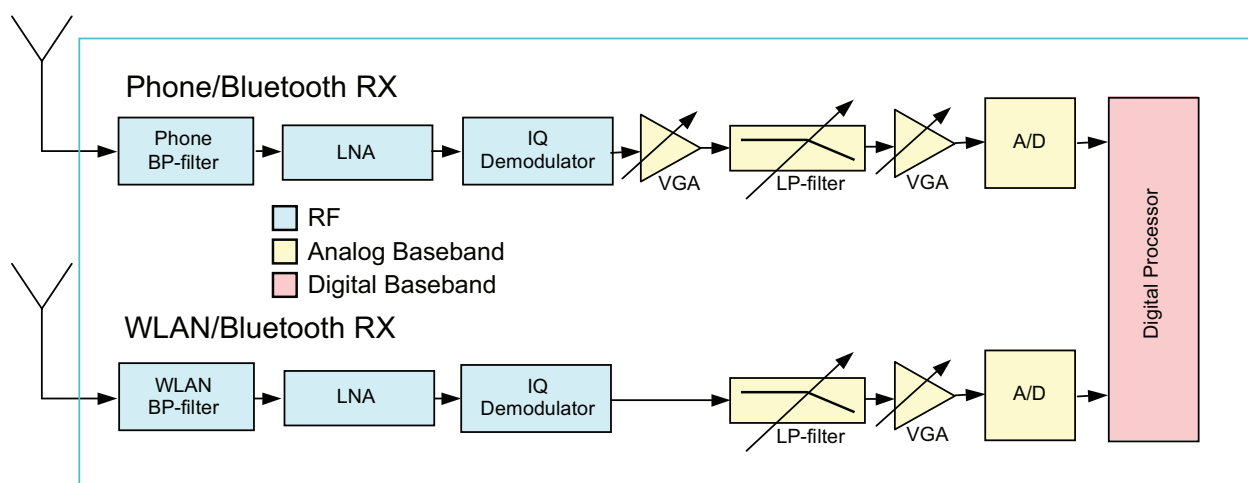

Fig. 1.   Receiver channels

From these considerations, we defined the receiver and transmitter architectures shown in Fig. 1 and Fig. 2, respectively. These architectures reflect the following basic ideas:

- two parallel receiver (RX) chains based on direct conversion architecture are implemented, one supporting all cellular standards and Bluetooth, and the other supporting all WLAN standards and Bluetooth;
- two parallel transmitter (TX) chains are implemented, one based on direct modulation for GSM, Bluetooth and possibly WCDMA (UMTS), and the other, based on direct conversion architecture, for all WLAN standards and Bluetooth;
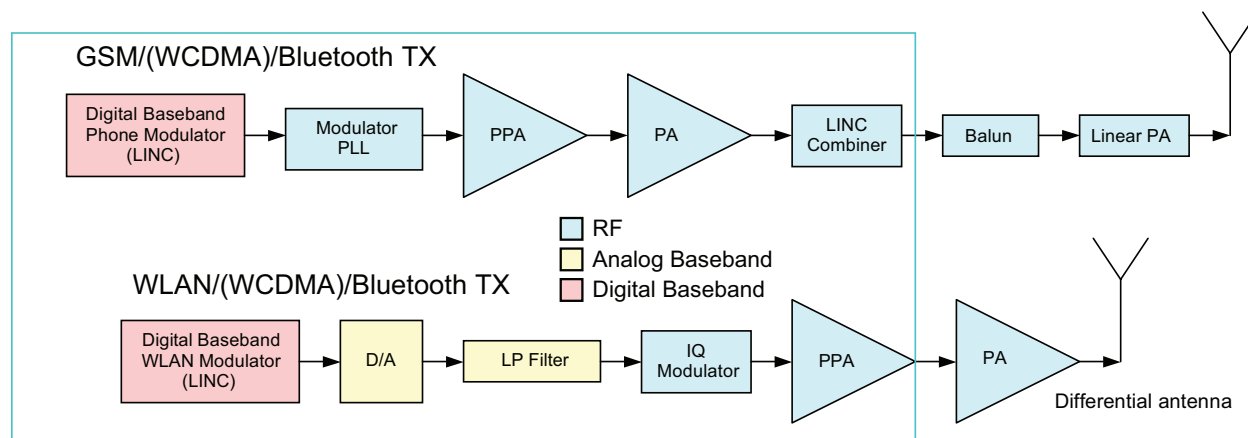
Fig. 2.   Transmitter channels

- the RX and TX chains covering the cellular standards can reconfigure themselves in a short time (less than 200μs), thus allowing vertical handover between GSM and WCDMA, which do not need to operate concurrently;

- vertical handover between cellular and WLAN standards, which can operate concurrently, is based on the use of two different transceivers;

The project "Enabling technologies for reconfigurable wireless terminals", funded by the Italian National Project FIRB, is a first step toward the above mentioned multi-standard integrated transceiver. In particular, five different chips, which represent a preliminary step towards the final device, are presently under testing: (1) receiver and (2) transmitter for DCS1800, UMTS, and Bluetooth, (3) receiver and (4) transmitter for WLAN at 2.4GHz and 5GHz, and Bluetooth as well as (5) the digital processor for all standards. This paper presents the baseband section (both analog and digital) for all five chips, discussing the most important design aspects and the achieved experimental results. RF circuits are presented in a companion paper.

II. ANALOG BASEBAND SECTION

The challenges in designing the analog baseband section of a reconfigurable transceiver are mainly related to the very different specifications of the different standards. In particular, bandwidth, gain, noise, resolution and linearity requirements are quite different from one standard to another. One "brute force" approach to design could be to select the most stringent requirement for each parameter, thus deriving a set of specifications valid for all standards. This approach, however, is definitely not efficient, especially in terms of power consumption. A more reasonable approach, which has been adopted for the design of the circuits reported in this paper, is to adapt the circuit performance, and hence the power consumption, to the standard considered. This adaptation of an analog device is performed through a digital control which either adjusts the biasing conditions of the active building blocks (e.g. operational amplifiers) or turns on or off entire stages (e.g. in an analog-digital converter or in a programmable-gain

amplifier) or reconfigures the interconnections among the blocks of the circuit. The details of architectural choices and circuit design for the receiver and transmitter chains are reported in the next Sections.

A. Receiver Analog Baseband Channel

The input spectrum of the receiver baseband block typically includes adjacent channels, in-band and out-of-band blockers, that can dominate (by up to 40-60dB) the signal to be processed. For this reason analog baseband blocks are required to exhibit not only a target in-band dynamic range, but excellent linearity for out-of-band signals. This is because a non-linear behavior with out-band-signals would result in an intermodulation whose product components would fall in the signal band, corrupting the signal quality. The analog baseband block of a receiver is composed of a series of Voltage Gain Amplifiers (VGA) and Low-Pass Filters (LPF). The VGAs increase signal amplitude, while the LPF reduces the amount of the out-of-band signal in order to increase the signal dynamics available for the useful signal. This functionality is shown in Fig. 3.
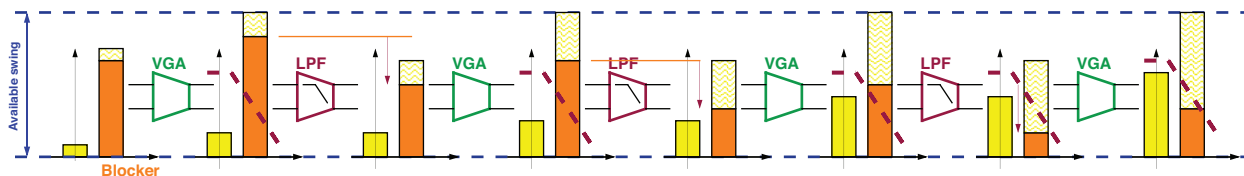

Fig. 3.   Receiver baseband analog signal processing

The design of the receiver baseband channel implies a trade-off between LPF selectivity (higher filter selectivity would result in a lower number of stages) and circuit complexity. In the design considered in this paper, we used a structure with two VGAs and one LPF, as shown in Fig. 4.
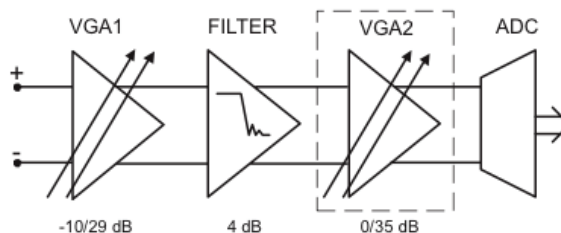

Fig. 4.   Block diagram of the developed analog baseband signal processing channel

For the channel devoted to cellular application, the signal is amplified by 59dB. This is because the input signal can be very low. The VGA1 (with a gain programmability in the 0dB-29dB range) then requires a reduced linear range, while it must have a very low Input Referred Noise ($IRN \approx 5$nV/$\sqrt{\text{Hz}}$). This constraint was satisfied by using an open loop approach implemented by a resistively-degenerated and resistively-loaded differential stage, whose dc-gain is fixed by a resistive ratio. In this scheme we used an open-loop architecture to reduce the power

consumption. For small input signals, a large gain is required. This is achieved by minimizing the degeneration resistance, which also reduces the *IRN*, as required by the low input signal amplitude. On the other hand, for large input signals, a reduced gain is required. This is achieved by maximizing the degeneration resistance, which also increases the linear range, as required by the large input signal amplitude. Several solutions for the filter are proposed in literature. Active-*RC* structures exhibit excellent linearity at the cost of high power consumption [1]. On the other hand, g_m-*C* filters feature a reduced linear range but with low-power consumption [2]. In this design we developed a novel structure that is the merging of the two solutions above and is called "active-g_m-*RC*". Fig. 5 shows the 2nd order low-pass active-g_m-*RC* cell structure in its single-ended form.
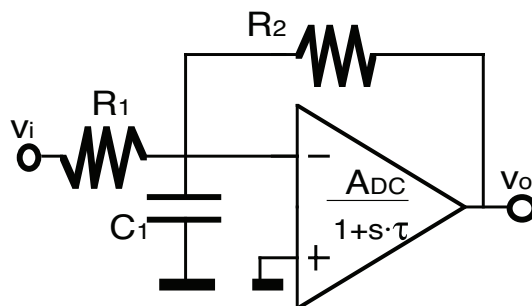

Fig. 5.   The active-g_m-*RC* biquadratic cell

The operational amplifier (op-amp) has a single-pole transfer function (in the frequency range of interest) that is taken into account in the transfer function synthesis. An Adjusting Circuit controls the op-amp frequency response in order to track the time constant of the passive components (*R* and *C*). This has the effect to transform the dependence of the filter frequency response on the transistor parameters into a dependence only on the passive component values (*R*'s and *C*'s). The active-g_m-*RC* cell exhibits the following features, which make it preferable for the implementation of the baseband filter of portable multi-standard terminals:

- *low power consumption* (a key objective for portable terminals): one op-amp is used to synthesize a 2nd order transfer function, halving the power consumption compared with standard two-op-amp active-*RC* biquadratic cells. In addition, the op-amp frequency response is used to synthesize the filter frequency response. Thus the op-amp unity-gain-bandwidth is comparable with the filter pole. This reduces its power consumption with respect to other closed-loop structures (active-*RC* or MOSFET-*C*), in which the op-amp unity-gain bandwidth $f_u > 50 \div 100 \cdot f_{LP}$ is used, requiring a large power consumption;

- *high-linearity*: a very large linear range is achieved due to its closed-loop structure. Moreover, out-of-band signals are first filtered by the very linear R_1-C_1 low pass filter at the input. This gives a very high out-of-band *IP*3 (3rd order intercept point), which is particularly interesting in telecom systems where the higher amplitude of out-of-band blockers requires a large out-of-band linearity;

- *frequency response accuracy*: the Adjusting Circuit makes the op-amp frequency response depend on the passive

component values (R and C) spread, which is the only spread to be compensated.

The 4th-order UMTS/WLAN reconfigurable filter is realized by the cascade of two active-g_m-RC biquadratic cells. The filter can be reconfigured to adjust the bandwidth to the selected standard (2.11MHz and 11MHz for UMTS and WLAN, respectively) by a single bit that controls the values of the resistors (this keeps the overall noise constant). In addition, in the UMTS case the power consumption is reduced by controlling the input stage device sizes and their current level. For both standards, the capacitors are grounded in order to seen by the common-mode signal as well. Otherwise, a high frequency resonance for the common-mode signals would be present. This implies that the capacitance dominates the overall filter area. However, sharing the capacitors for the two standard configurations thus minimizes the area occupation. The capacitor values are finally adjusted by the tuning circuit to compensate for technology variations. A key feature of this structure is the limited power consumption due to the use of low f_u op-amps. In fact, the f_u/f_{LP} ratio is less than two.

The full filter design has been optimized in order to minimize the power consumption using a specifically developed automatic design toolbox, which for a given set of constraints (noise, linearity, transfer function) and device models, directly defines all the device sizes in order to minimize power. Finally, the amplitude of the input signal of the second VGA is very large (after the previous amplification), so that for this stage linearity is more important than noise performance. For this reason we used a closed loop architecture, with two 17.5dB gain-stages and a 2.5dB gain resolution. This block also implements an additional 1st order LPF. Finally since the offset may be significant at this stage, due to the large amplification of the previous stages, it includes an offset compensation circuitry. Tab. I reports the overall simulated baseband channel features.

TABLE I

SUMMARY OF THE SIMULATED RECEIVER ANALOG BASEBAND CHANNEL FEATURES

Parameter	Value
Filter order	5th
Power supply	2.5V
Power consumption (UMTS/WLAN)	51.7mW/55mW
VGA1	19mW
LPF (UMTS/WLAN)	22.7mW/45mW
VGA2	10mW
Gain range (UMTS/WLAN)	−6dB÷68dB/4dB÷39dB
In-band IIP3 (UMTS/WLAN) @ Max gain	5dBm/1dBm
Out-of-band IIP3 (UMTS/WLAN) @ Max gain	30dBm/26dBm
IRN (UMTS/WLAN) @ Min gain	$9.6\mu V_{RMS}/51\mu V_{RMS}$

B. Analog-to-Digital Converter

The architecture of the ADC stems from the observation that while Delta-Sigma is the topology of choice for low speed high resolution conversion, pipelined topologies are well suited to medium-high speed and medium-low resolution. Thus the combination of the two can cover a wide portion of the speed-resolution space [3]–[6]. Furthermore, both topologies include the same building blocks, such as op-amps, comparators, switches and capacitors. The difference between them is the network that interconnects the blocks. Thus, a converter made of those building blocks and a reconfigurable interconnection network can implement different topologies and work at the different bandwidth and resolution levels needed for the various standards. In addition to supporting reconfiguration among different standards, the ADC can adapt its architecture, performance and power consumption to its environment. A dynamic configuration manager, considering the required Quality of Service (*QoS*), can exploit the reconfigurability of the ADC in order to save power and extend the battery life-time [7], [8].

Another key aspect of the ADC architecture proposed in this work is the extensive use on the digital side of background self-calibration algorithms [9]–[21]. Those algorithms are used in the background to overcome the effects associated with the limited precision of analog building blocks and with component mismatches that impair performance. This also turns out to be a very efficient way to reduce the power consumption of the ADC. For instance, the capacitances can be sized to meet thermal noise requirements rather than matching (which is compensated digitally), which often results in a smaller capacitor size and therefore less op-amp current for the same gain-bandwidth product. The complete ADC architecture, including the digital blocks used for background self-calibration, is highlighted in Fig. 6. The ADC core is made of six equally sized stages, each one resolving 1.5 bits, followed by a 2-bit quantizer. In order to further reduce the power dissipation, every pair of stages shares one op-amp; this block therefore requires only three op-amps. The maximum achievable resolution is 8 bits, and a 6 bit resolution can be digitally selected. In this case the first two stages are switched off and the input signal is fed directly to the third stage. To increase the overall ADC resolution a 2/3 bit reconfigurable stage has been added before the core ADC block, extending the overall resolution to 10 or 11 bits. In $\Delta\Sigma$ mode, the hardware is reused to implement a second order 2-bit modulator which achieves 74dB *DR* over a 200kHz signal bandwidth.

Component matching and a finite op-amp gain-bandwidth product affect the linearity of the ADC, thus reducing the *IP*3 of the system. In order to overcome this potential problem, two digital algorithms have been implemented in the chip. The first one, called DAC Noise Cancellation (DNC) [21], estimates and corrects the non-linearity error introduced by the multi-bit (2/3 bit) unit element DAC mismatch of the first conversion stage. The second one, called Gain Error Correction (GEC) [15], [19], estimates and corrects gain errors in the first, second and third stages. In both techniques the analog error is modulated by a pseudo-random noise sequence and then the digital output is processed in order to extract the modulated information and digitally enhance the ADC performance. It is worth noting that those techniques require few modifications in the analog part of the chip, while more complexity is added in the digital part. As CMOS technology scales, more digital signal processing (DSP) becomes available for the same area at a reduced power consumption; hence those techniques are very attractive when the reduction of
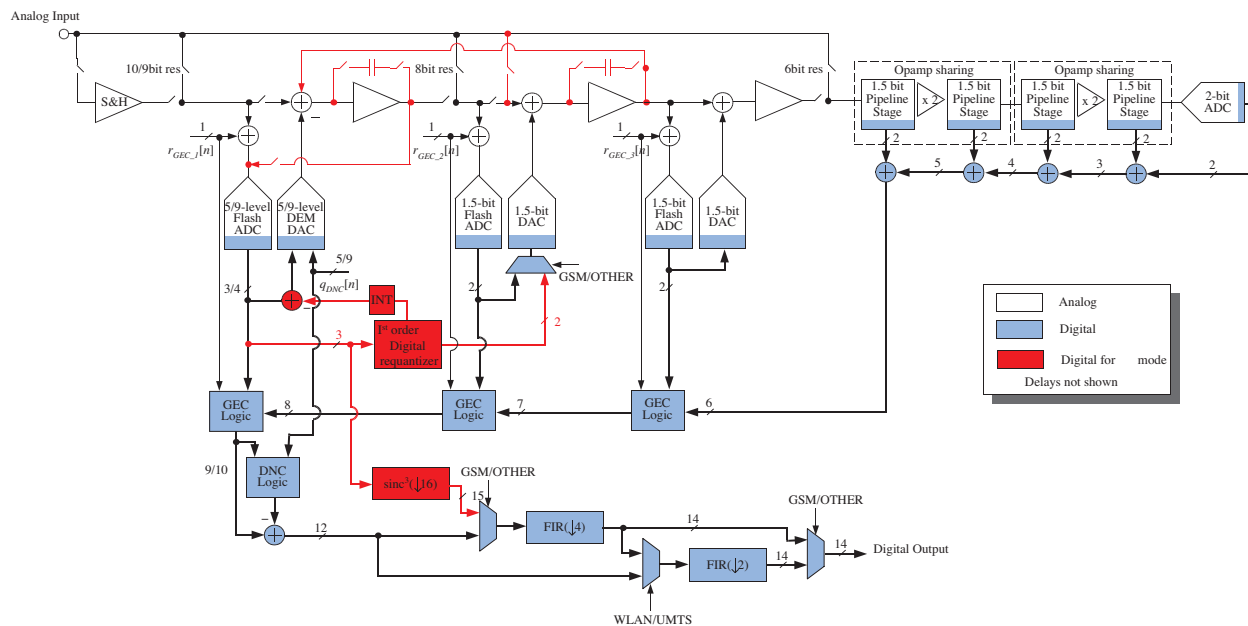
Fig. 6.   Block diagram of the reconfigurable ADC

the overall power consumption is of major importance, as in the case of portable terminals. Combining those digital techniques with the selective power-down of unused blocks can minimize the power consumption. For example, the single ADC in WLAN-mode can convert the input signal (10MHz bandwidth) with a 9 bit resolution consuming 6.8mA; if the resolution is switched to 6 bits the current used is only 3.2mA. In UMTS-mode, the required resolution is 11 bits and the input signal (2MHz bandwidth) is converted using 8mA. When a GSM-EDGE signal has to be converted, the architecture is reconfigured in $\Delta\Sigma$-mode; in this case only the first two stages are used and the current consumption is 4mA.

A key aspect for a multi-mode terminal is the ability to switch from mode to mode in a seamless way. This in turn sets a requirement on the time allotted for so-called vertical handover (in some cases this time is specified by the standards). The same is true for a change in the QoS, which should be transparent to the user. This ADC can change mode in less than 100μs, while a switch from 10 bit to 6 bit resolution takes only 600ns. Fig. 7 shows the measured power spectral density of a full scale 9.6MHz input sine wave before and after the background correction in WLAN mode. The digital algorithms improve the SNR by more than 10dB and the $SFDR$ by more than 19dB. The digital algorithms dramatically reduce the second and the third harmonic to a level of −80dBc and −77dBc. Fig. 8 shows the measured relationship between the analog power consumption and the $SNDR$ of the converter. This reconfigurable converter efficiently exploits the trade-off between resolution and power consumption highlighted in this plot.
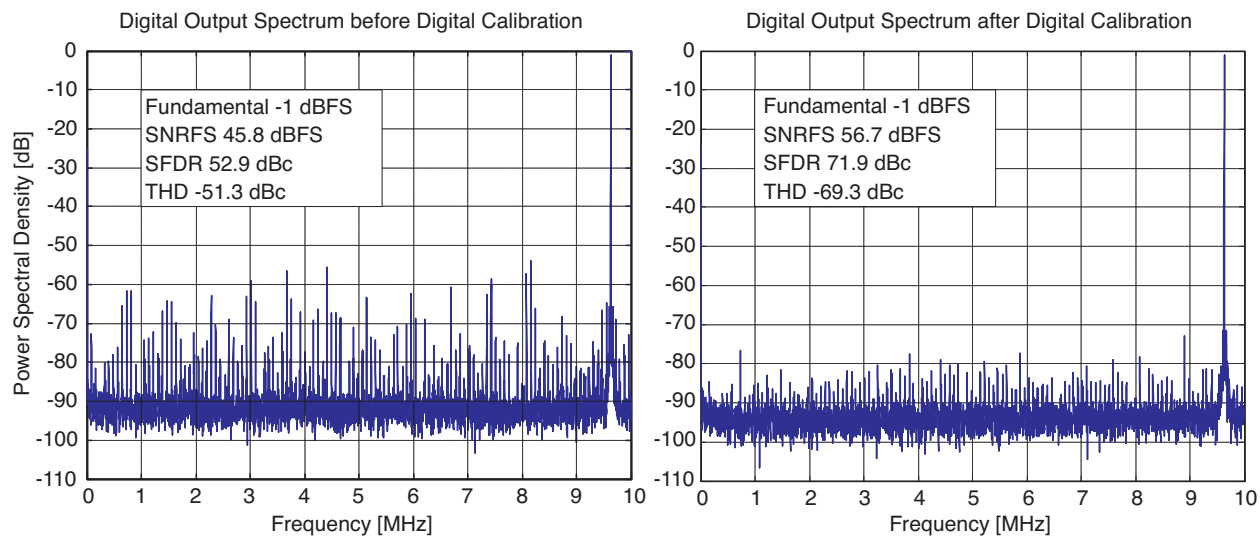
Fig. 7.   Measured power spectral densities at the ADC output before and after background calibration
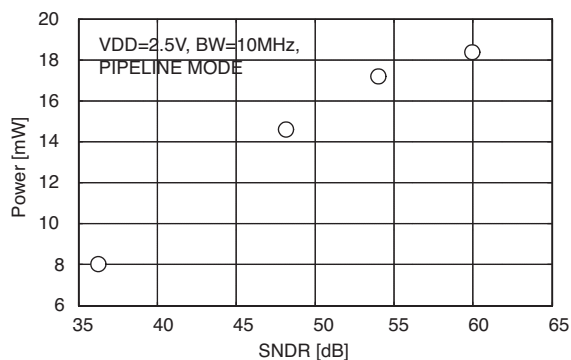

Fig. 8.   Measured variation of the analog ADC power consumption as a function of the *SNDR*

C. Transmitter Analog Baseband Channel

The transmitter analog baseband channel has to transform the digital data stream produced by the digital processor (described in Section III) into an analog signal, which is then delivered to the RF section. The analog baseband channel developed for the reconfigurable terminal described in this paper presents the following challenging key features:

- it operates at low voltage (i.e. at 1.2V);
- it is reconfigurable in terms of bandwidth, resolution and data-rate in order to satisfy the different standards;
- its power consumption changes depending on the selected standard, in order to maximize the efficiency.

The overall DAC+filter architecture, implemented with a fully-differential topology, is shown in Fig. 9. An 8 bit Current-steering DAC drives a resistive load ($R_L = 600\Omega$) and the resulting output voltage is directly applied to a

4th-order low pass analog filter. A number of design choices, described in the rest of this section, were made in order to minimize the power consumption, while achieving a reconfigurable device.
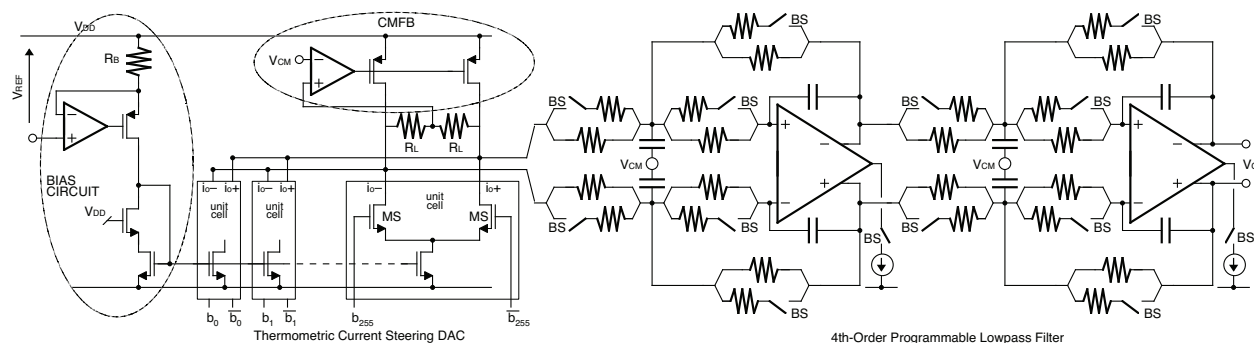

Fig. 9.   DAC+filter structure

Regarding the DAC structure, a current-steering approach has been preferred to a R-$2R$ ladder DAC, since it avoids the use of input and output reference voltage buffers, which would increase power consumption. Regarding the coupling between DAC and filter, use of DAC load resistors R_L (instead of forcing the DAC current directly into the virtual ground of the first filter op-amp) allows us to decouple the DAC output current from the filter op-amp output current, which can be designed to be much smaller than the former. As a consequence the desired output dynamic is achieved by using large resistances in the filter, and by making the filter input impedance much higher than R_L. This power consumption reduction is obtained at the cost of an increased thermal noise which, however, is still negligible with respect to the quantization noise. The DAC structure has been designed to achieve the worst-case linearity and dynamic range target specifications even in the presence of worst case technology mismatches and parameter variations [22].

The same analysis suggested that, due to the 8 bit resolution, the area penalty of a fully thermometric implementation is negligible with respect to the linearity improvement. The unit current source area is designed to satisfy the matching requirements. A maximum relative standard deviation σ_{rel} of 2% results in an Integral Non-Linearity (*INL*) yield of 99%, with 0.5LSB as the upper limit [23]. Thus the minimum area ($W \times L$) of each current source is obtained from the Pelgrom model of the mismatch [24]. The choice of the unit current source overdrive (V_{ov}) is a trade-off between low sensitivity to threshold voltage mismatches (which would require a large V_{ov}) and the headroom available from the 1.2V supply (which limits the maximum value of V_{ov}). As a consequence, we chose the value $V_{ov} \approx 70\text{mV}$, which requires an area of $36\mu\text{m}^2$ for the unit current source. Finally, the unit current level (I_{UNIT}) is designed to minimize the glitches introduced by the charge injection of the switches *MS* (Fig. 9). The glitch amplitude is reduced by setting minimum device sizes for the differential switches and by driving them with minimum swing signals ($V_{low} = 300\text{mV}$, $V_{high} = 800\text{mV}$). The resulting glitch is about 1μA. To make this contribution negligible a I_{UNIT} of 5μA is chosen, which implies $W = 6\mu\text{m}$ and $L = 6\mu\text{m}$ for the unit current cell. The coupling between different unit sources is reduced by using a driver circuit for each of them to provide the

desired voltage levels for the switches. The current I_{UNIT} is generated with the bias circuit shown in Fig. 9, which makes $I_{UNIT} = V_{REF}/R_B$. Resistance R_B is matched with the load resistance (R_L) in order to make the R_L/R_B ratio constant. This reduces the dependence of the DAC output voltage amplitude on the technology spread.

The output common mode voltage is fixed to analog ground ($V_{DD}/2$) through a common-mode feedback circuit CMFB. The maximum swing on each of the output nodes (controlled by V_{REF}) is fixed to 350mV$_{pp}$ around the analog ground. This is a trade-off between having a significant input signal for the filter following and introducing a negligible signal distortion due to the current source output impedance, which is anyway large thanks to the cascoding action of the current switches. This choice implies a value of Ω for R_L, considering that the full-scale peak-to-peak differential current is equal to 1.275mA. Again in designing the 4th-order Bessel low-pass reconfigurable baseband filter, particular care was taken to reduce power consumption. The filter is the cascade of two identical multi-path active-RC biquadratic cells. Active-RC allows us to achieve the required linear range. The use of a single op-amp to synthesize two poles reduces power consumption. In this structure the op-amp bandwidth has to be about 50-100 times broader than the position of the filter poles.

The op-amps that we used are based on a fully-differential Miller-compensated two-stage topology, which allows for rail-to-rail output swing. In fact, the full-scale peak-to-peak differential output voltage of the block is 1.8V, which means a filter DC gain of 8.2dB. The op-amp bandwidth is reduced for lower pole frequencies (UMTS) by reducing the bias current. As a result, the power consumption is also reduced. The DAC+filter block can be reconfigured for two standards (WLAN and UMTS) as follows. The DAC sampling frequency (F_S) can be changed in order to achieve the required resolution, exploiting the resulting oversampling ratio (OSR) to increase the signal-to-noise ratio (SNR) to above the 8 bit level. In the case of WLAN, the OSR is 4 (using F_S = 80MHz), which leads to a resolution of 9 bits (this implies a design margin of 1 bit with respect to the required SNR). Similarly, in the UMTS case, the OSR is 8 (using F_S = 40MHz) with a 1.5 bit additional resolution. On the other hand, the filter transfer function can be programmed for two bandwidth values (11MHz and 2.11MHz) through a selection bit (BS), which digitally controls the value of the resistors, and the in-band noise floor. The smaller band for the UMTS standard (obtained with larger resistance values) can accept the resulting larger noise floor. The choice of programming the cut-off frequency with the resistance values allowed us to reduce the power consumption for low bandwidth. The selection switches were carefully designed and layouted in order to minimize their parasitic effects. The switches are connected to virtual ground nodes (op-amp inputs, with a limited voltage swing on the parasitic capacitances) or to low impedance nodes (op-amp outputs). In addition the capacitor values can be adjusted by ±35% with a 4 bit digital word, in order to control the technology spread or to allow a fine selection of filter bandwidth. Finally, the filter power consumption is optimized to the standard selected by the signal BS, which selects a bias current level for the op-amp, in order to save power when a smaller bandwidth is programmed.

The DAC+filter block has been fabricated in a standard 0.13μm CMOS technology with six metal layers and MIM capacitors. Fig. 11 shows the micro-photograph of the chip, whose active area is 0.8$_2$. Fig. 10 shows the measured output power spectra of the circuit for a WLAN 802.11a and a UMTS input signal. In both configurations the transmitter baseband channel respects the transmission masks for the standards. The measured features of the
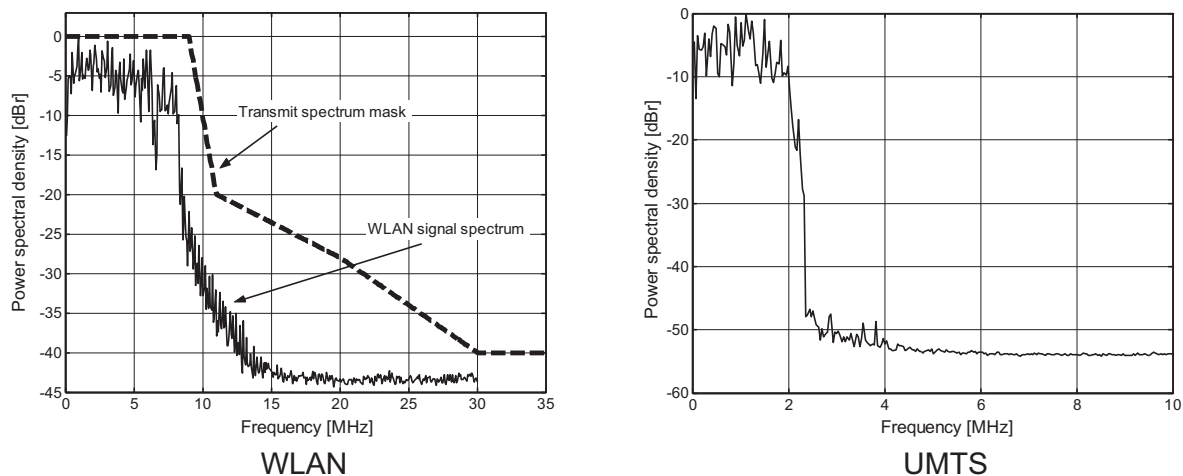
complete circuit are summarized in Tab. II.

Fig. 10. Measured output power spectra of the transmitter baseband channel for a WLAN 802.11a and an UMTS input signal

III. DIGITAL BASEBAND SECTION

Digital signal processing systems aimed at the wireless consumer market must handle a variety of high performance real-time tasks, for which traditional general-purpose processors are often a poor match. Flexibility is required to reduce masks and design costs, high computational power must match the growing complexity of applications, and low power consumption is needed to ensure portability, under severe battery capacity constraints.

A common way to tackle such constraints is by mapping critical computational kernels on custom-designed hardware units inside the processor pipeline [25], [26], providing application specific extensions to the standard instruction set. However, in most cases, extensions are defined at mask level, severely limiting the flexibility and application field of the device. Such application-specific standard processors (ASSPs) also involve high up-front non-recurring design costs, justified only for long lifespan and high volume products. Low volume or frequently updated products require novel methodologies to match flexibility with high computational power and low cost. *Reconfigurable processors* are an appealing option [27]–[33], combining standard processor cores with embedded programmable hardware. Reconfigurable processors offer high programming flexibility by means of *run time extension* of the instruction set. Non-critical computations or control-dominated tasks can be efficiently mapped on the hardwired portion of the processor, taking advantage of its software programmability and shortening the overall development time. According to Amdahl's law and the 90-10 rule (90% of time is spent executing 10% of the code, [34]), performance can be enhanced by up to one order of magnitude by simply focusing implementation efforts on identification and improvement of relatively small critical kernels. Several techniques can be used to exploit spatial (i.e. concurrency between resources) and temporal (i.e. pipelining) parallelism at the reconfigurable instruction level. The widespread knowledge of the ANSI-C programming language among embedded systems and
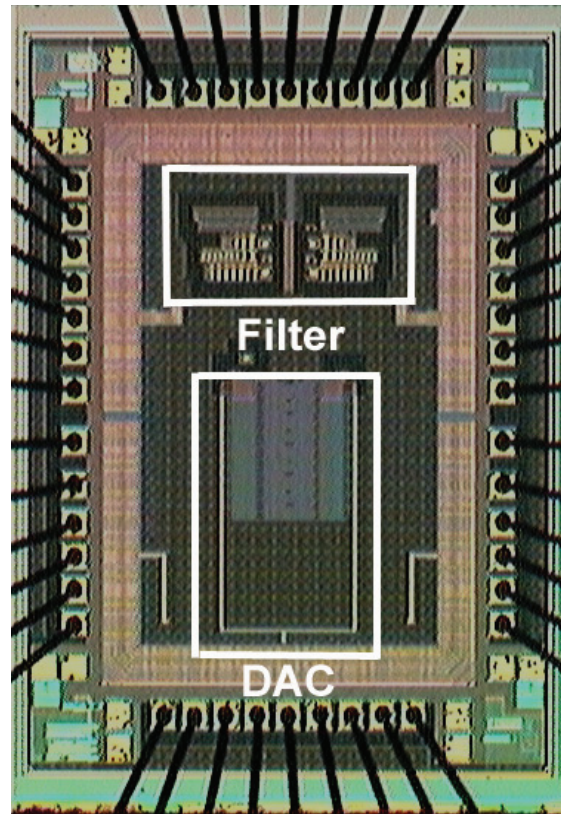
Fig. 11.   DAC+filter chip micro-photograph

wireless algorithm developers suggests using it as the application description language for reconfigurable processors as well. This introduces the problem of translating behavioural C into some form of HDL description, or directly into hardware (in the case of reconfigurable devices, configuration bits). Most existing reconfigurable architectures use automated or semi-automated C-to-HDL conversion tools to plug into standard synthesis and Place & Route techniques for configuring the hardware accelerator. Unfortunately, the introduction of these abstraction layers hides many implementation choices from the designer, making it difficult to obtain high-quality results without a deep understanding of the tools and the underlying architecture.

In this section we present a reconfigurable system based on a VLIW processor architecture including a runtime reconfigurable embedded function unit called PiCoGA. The integration of the PiCoGA inside the processor core reduces communication overhead towards other functional units, thus making it easier to use for a variety of computation kernels. In what follows, we will use the name "XiRisc" to refer to the processor architecture with the tightly integrated reconfigurable unit, as opposed to a standard processor core accelerated by an external device. In this context, trading off some potential performance speed-up for a higher level of programmability was a key design decision, since it permits a dramatic decrease in application development time, and hence it may allow a design team, always working under severe time pressure, to speed up a larger number of critical kernels. Consequently,

TABLE II

SUMMARY OF THE MEASURED TRANSMITTER ANALOG BASEBAND CHANNEL FEATURES

Parameter	Value	
Technology	CMOS 0.13μm	
Supply voltage	1.2V	
Core area	0.8mm^2	
Standard	**WLAN**	**UMTS**
F_s	100MHz	50MHz
Filter bandwidth	11MHz	2.11MHz
Differential output swing	1.8V$_{pp}$	1.8V$_{pp}$
DR (@ FS)	54dB	58dB
SFDR (@ FS)	54dB (@ 3MHz)	61dB (@ 0.6MHz)
THD (@ FS)	−51dB (@ 3MHz)	−60dB (@ 0.6MHz)
OIP3	29.3dBm	32.5dBm
DAC power consumption	5.4mW	5.4mW
Filter power consumption	5.6mW	3mW
Total power consumption	11mW	8.4mW

the PiCoGA was designed to be programmed using a C-like description language, resulting in tighter integration of the hardware and software design flows.

A. XiRisc Architecture Description

XiRisc can be described as a Very Long Instruction Word (VLIW) RISC processor, with a 32-bit datapath (see Fig. 12). The basic XiRisc instruction set includes a set of DSP-specific instructions such as *multiply-and-accumulate*, *branch-and-decrement*, SIMD (*Single Instruction Multiple Data*) and saturating arithmetic operations. The XiRisc architecture [35] is strictly separated into control logic and data path. The micro-architecture was designed to provide a simple straightforward control model and to offer the programmer full control over the processor resources. All Functional Units (FUs) are independent, concurrent, and fully pipelined.

The control architecture is based on the classic RISC five stage pipeline, with a strict load/store architecture that may result in a bottleneck for memory intensive applications. In order to maintain a high data throughput to and from the FUs, the processor is structured as a Very Long Instruction Word machine, fetching and decoding two 32-bit instructions each clock cycle. The instruction pairs are then executed concurrently on the set of available FUs, determining two symmetric execution flows. Simple, commonly used FUs such as ALU and Shifter are duplicated in both data channels, while other FUs such as the multiplier and the branch unit are shared between the two channels. To simplify hazard handling, software compilation schedules instruction pairs which avoid simultaneous access to
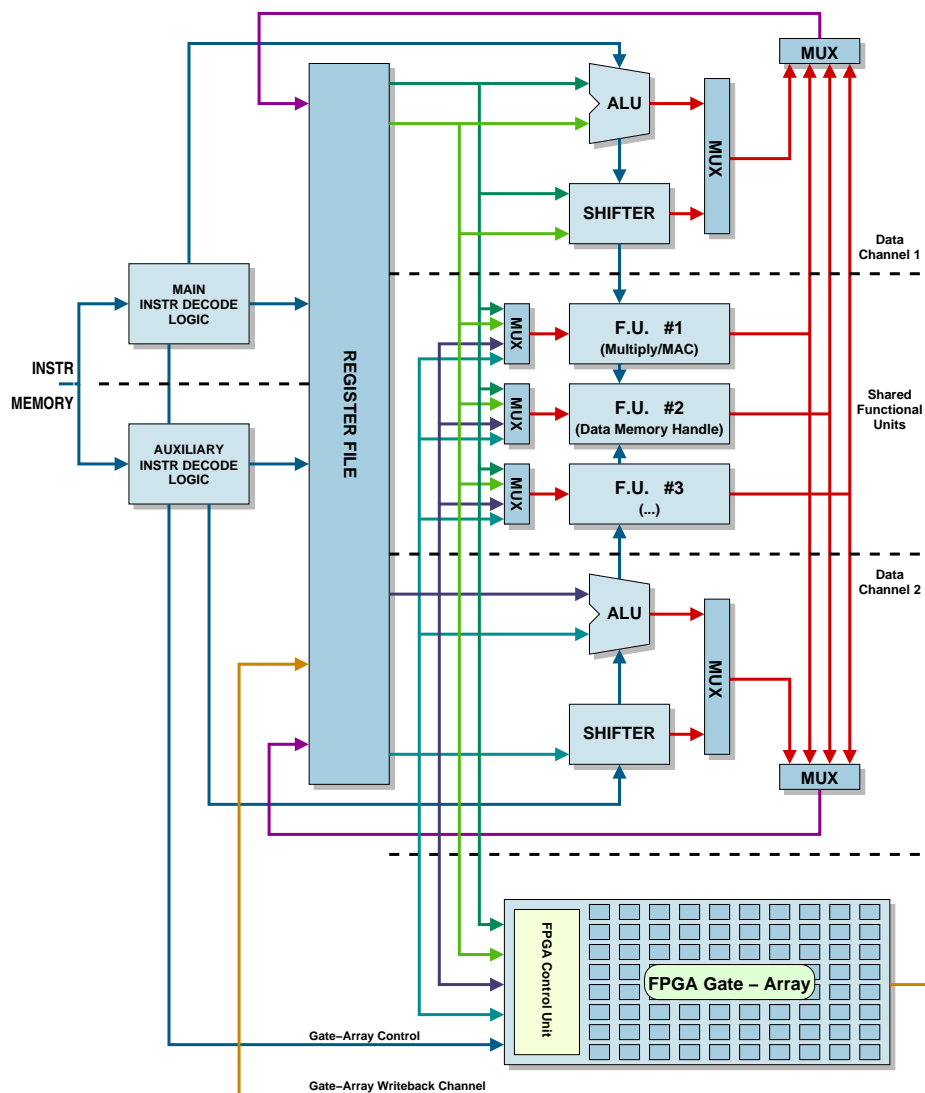
Fig. 12.   XiRisc architecture

the same shared FU, so that pairs of instructions never need to be separately stalled during computation. All other pipeline hazards are resolved at run-time by a fully bypassed architecture and a hardware stall mechanism. The PiCoGA is handled by the control logic and the compilation tool chain as a shared functional unit. Operands are read from and results written into the register file, and specific assembly instructions are used to control both array elaboration and configuration. The XiRisc instruction set is extended at runtime by mapping new functionalities on the gate-array through the issue of *pGA-load* and *pGA-free* instructions. As a result, after a configuration latency of a few hundred cycles, extended instructions called *pGA-op*'s are available to execute custom computations on the PiCoGA pipelines. After an execution latency that may range from 1 to 24 cycles *pGA-op* results are written back into the processor register file.
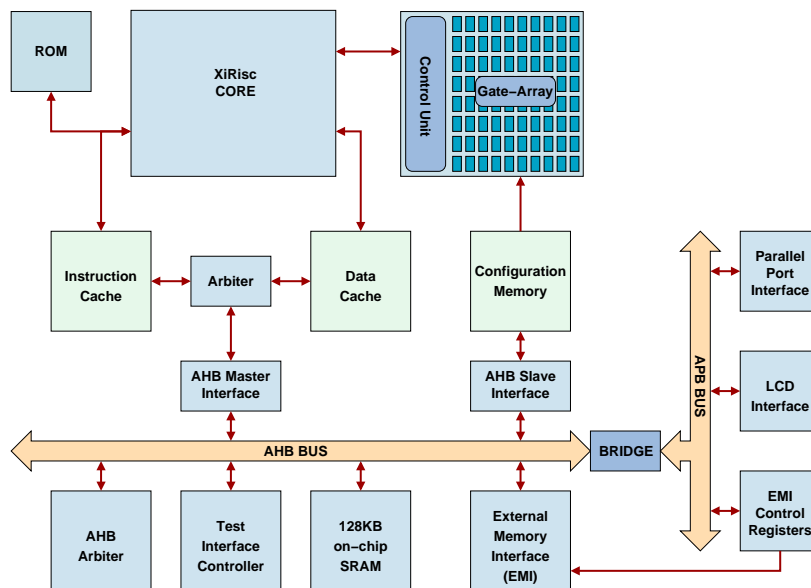
Fig. 13.   System architecture

From an architectural point of view, the main differences between the PiCoGA and the other FUs are:

1) The PiCoGA supports up to 4 source and 2 destination registers for each instruction. In order to avoid bottlenecks on the write-back channels, a special purpose register file has been designed, featuring four read and four write ports, of which two are reserved for the PiCoGA.

2) The PiCoGA instructions may have unpredictable latency, if they directly include data-dependent loops. A special register locking mechanism was designed to maintain program flow consistency in case of data dependency between PiCoGA instructions and other instructions.

The PiCoGA is seen as a customizable unit that dynamically adapts the instruction set to the application workload. Application-specific custom instructions are synthesized using a dataflow-oriented paradigm. They are modeled as ANSI-C procedures which are automatically translated by the compiler into data-flow graphs (DFGs). The DFG extraction step from the ANSI-C function description is performed using a customized version of the Impact compiler [36].

B. System Architecture

The high-bandwidth memory transfer requirements that are typical of DSP algorithms required the careful implementation of a layered memory hierarchy (see Fig. 13), supported by an on-chip AMBA bus architecture. A single AHB channel is used to load instructions and data from the system memory. Possible conflicts on the bus interface are resolved by a dedicated arbitration logic within the AHB master. Processor transfers are converted into AHB cycles, supporting wrap burst transfers for cache line refills and locked accesses. The AMBA system is connected to a set of IO peripherals, on-chip SRAM and an interface for off-chip memories (EMI) featuring up to

1.328 Gbit/s bandwidth. In order to support a parallel load of data and instructions according to its Harvard internal structure, the processor includes direct-mapped instruction and data caches.
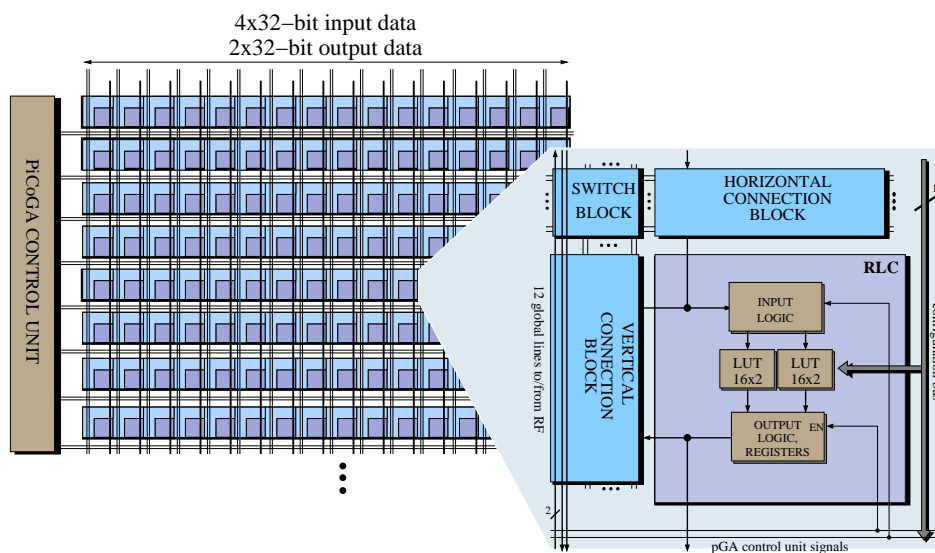

Fig. 14. PiCoGA structure

Cache memory sizes are programmable at time of synthesis. The memory management unit supports run-time cacheable space configuration, allowing a flexible distribution of program, data and gate array configuration among different memory spaces. Furthermore, the data cache may alternate at runtime between write-back and write-through policies, depending on the running application.

Memory resources have also been adapted to handle configuration bits of the reconfigurable gate-array, which are described in the processor addressing space (viewed as part of the program code). Configuration bits for different *pGA-op* operations can thus be placed anywhere in the addressing space but, in order to minimize configuration latency, they are loaded on a specific 64KB on-chip *configuration cache* memory directly connected to the array. Programming data can be loaded into this memory via the AHB bus from a larger third library of configurations stored in an external SRAM or FLASH. The peripheral APB bus contains configuration and status registers for several system peripherals, a programmable timer, an interface for external LCDs, and a parallel port interface for connection to an external host.

C. Pipelined Configurable Gate Array

The PiCoGA (Pipelined Configurable Gate Array) [37] is a reconfigurable datapath which has been designed specifically to be integrated inside a processor core. The aim was to provide a device which can reduce both execution time and energy consumption over a wide range of heterogeneous applications and can be easily programmed by a software developer.

Computational efficiency in a reconfigurable fabric is achieved by exploiting parallelism and implementing customized operators with the minimum size required by the operands. Achievable parallelism is related to the capacity of the device, while operand size customization depends on the array granularity. Both are design parameters whose values have been a direct consequence of the architectural choice of integrating the accelerator inside the processor core. In this context there is no communication overhead between the processor FUs and the configurable device, so that they can cooperate together, under direct control of the processor logic. The computation can be partitioned at a very fine level of granularity between the reconfigurable device and the processor functional units, almost without penalty. As a consequence, a relatively small array with a fast reconfiguration time can achieve high acceleration of computational kernels.

Concerning device granularity, we considered that the processor functional units perform very well 32-bit standard operations, while they are very inefficient when dealing with unusual operations over a few bits. Therefore, in order to efficiently cover the widest range of applications, we designed the configurable array with fine granularity to balance well the overall architecture.

The computational model for the PiCoGA has been chosen to be easily integrated inside a processor core. For this reason the gate array provides a hardware platform where application specific instructions can be easily implemented and added to the native instruction set. This is achieved by means of a special structure which supports direct mapping of pipelined computations. Concurrency does not have to be explicitly described by the user, in order to help software developers (who are used to sequential high level languages). Instruction level parallelism is extracted from sequential code by our tools in order to program the dedicated PiCoGA control unit. Once the PiCoGA is configured, the pipeline activity is automatically controlled, handling irregular input data flow, loops and other synchronization issues between pipeline stages. When a *pGA-op* instruction is decoded, new input data are provided by the register file to the PiCoGA and, depending on the connectivity of the mapped DFG, the control unit activates each PiCoGA row in the right order following a dataflow paradigm, whenever its operands are ready. At the end of the computation a write-back operation is performed.

From a structural point of view the PiCoGA is an array of rows, each representing a stage (or part of a stage) of the implemented pipeline. The width of the configurable datapath has been designed to fit the processor architecture, so each row is able to process 32-bit operands. As shown in Fig. 14, each row is connected to the others via configurable interconnect channels, and to the processor register file via six 32-bit global busses.

1) Configuration Caching: One of the reasons for tightly integrating an FPGA in a processor core is the opportunity to use it frequently, for many different computational kernels. However reconfiguration of a traditional FPGA can take hundreds or even thousands of cycles, depending on the re-programmed region size. Although execution can still continue on other processor resources, scheduling will hardly find enough instructions to avoid stalls that could nullify any benefit from the use of dynamically configurable arrays. Furthermore in some algorithms the exact function to be executed is only known at runtime, so that reconfiguration cannot be done in advance. In such cases it is very difficult to take advantage of a configurable unit.

Three different approaches have been adopted to overcome these limitations. A first level cache (or more

precisely a scratchpad, since it is managed directly by the software) has been implemented, which stores 4 configuration contexts for each logic cell inside the PiCoGA [38], [39]. Context switch takes place in one clock cycle, providing 4 immediately available *pGA-op* instructions. Furthermore Partial Run-Time Reconfiguration (PRTR) [40] is supported, allowing reconfiguration of just a portion of the array to implement more functions in the same context layer. While the PiCoGA is computing, reconfiguration of the next instruction can be performed, so that cache misses are rare, even when the number of configurations used is large.

These two mechanisms are useful to accelerate parts of a kernel or different kernels of the same algorithm. However, in the case of a reconfigurable cellular terminal, the communication protocol can even change during the same phone call, requiring the PiCoGA to be completely reconfigured. Therefore reconfiguration time has been shortened, exploiting a wide configuration bus to the PiCoGA. Reconfigurable Logic Cells (RLC) in a row are written in parallel with 192 dedicated wires, taking a few hundred cycles to complete the reconfiguration of one PiCoGA context layer. A dedicated second-level on-chip cache is needed to feed such a wide bus, while the whole set of available functions can be stored in an off-chip memory.

2) Reconfigurable Logic Cell: A Reconfigurable Logic Cell (RLC) is composed of a cluster of 24-input LUTs, each having 2-bit granularity. An RLC contains four pairs of registers, which are controlled by the configurable control unit or by another RLC. A single RLC can implement purely combinational logic, as well as both single cycle and two-cycle pipeline stages, in order to support timing optimization and improve the maximum throughput in a pipeline having a complex Data Flow Graph.

The RLC includes two internal feed-back paths: a synchronous one and an asynchronous one. The first one is used for implementing accumulator-like operators, while the second one enables a LUT to be fed with the output of the other one. Both internal paths are useful for reducing the amount of routing resources required. Initialization of state inside the array (e.g. value stored in an accumulator) is performed by dedicated hardware logic managed by the PiCoGA control unit.

Dedicated wires along each row are provided to achieve fast propagation of carry signals. A carry-select architecture has been implemented, where a 2-to-1 multiplexer, driven by the carry-in signal coming from a previous RLC, is used to select the correct carry-out. One of the LUTs (LUT1 in Fig. 15) computes carry-out signals, in the case of carry-in equal to both 0 and 1. Appropriate programming of LUT1 also allows one to use the chain for efficient implementation of a number of useful functions such as comparators, wide input logic gates (AND,OR,...), parity bit generator/checker and sign inversion. Furthermore the PiCoGA carry chain is enhanced combining the carry select architecture with a lookahead level-1 technique, which roughly cuts the critical path in half.

3) Programmable Interconnections: The PiCoGA routing network has been designed with 2-bit granularity, reducing the area occupation with respect to a single-bit granularity. However, input connection blocks provide both 2-bit and 1-bit connection granularity in order to maintain routability and efficiency of resources in cases like odd shifting and single bit control signals. Channels are composed of 15 pairs of tracks with a length of 3 tiles, which has been found to be a good trade-off between propagation delay and routability. Furthermore, four global horizontal lines have also been designed to support fast propagation of multi fanout control signals.
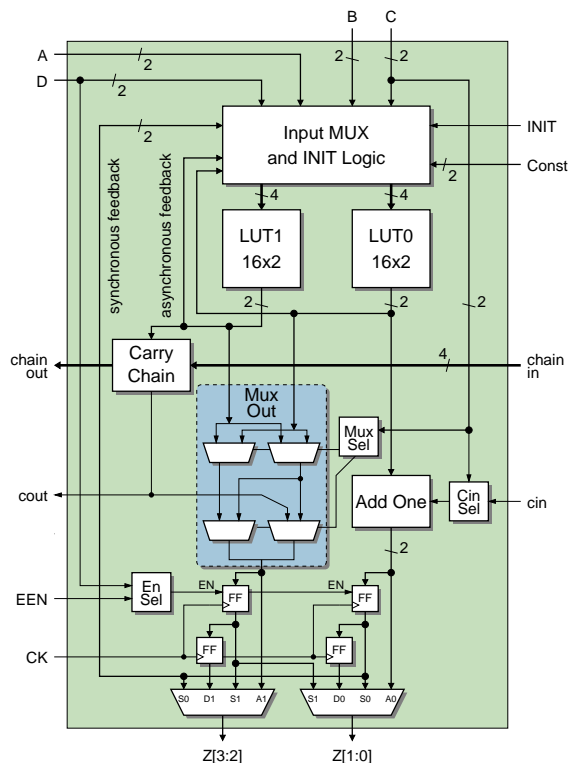
Fig. 15.   RLC architecture

In standard FPGAs the routing network is largely responsible for most of the area occupation and delays. This becomes even worse in the case of multi-context arrays, where configuration bits have to be replicated. For this reason connect and switch blocks include a decoding stage between configuration memories and programmable switches. For example the output connect block can connect an output line of the RLC to only a single wire in a routing channel, because of the 1 over N decoding logic introduced. This causes a small loss of routability, but allows one to have a logarithmic reduction of the number of multi-context SRAM cells needed [41].

D. XiRisc Computation Pattern

The exploitation of run-time reconfigurability requires the designer to explore a complex space in order to identify and optimize critical kernels in the target application [42]. Typical kernels, for a wide spectrum of embedded applications, are the cores of innermost loops, which can be usually described using traditional data-flow graphs. Significant speed-ups can be achieved by overlapping successive loops iterations. In case of configurable computing, we can increase throughput by pipelining overlapped and/or unrolled loop kernels. Of course, this technique requires accurate management of the custom pipeline. Starting from a source code written in (slightly extended) ANSI-C [43], a scheduler creates a pipelined DFG and then maps:

- *DFG-node operators* to the array of Reconfigurable Logic Cells (RLCs) in the PiCoGA;

- *pipeline management* to a row-based dedicated control unit which enables the execution of pipeline stages implemented on the PiCoGA computational resources.

The pipeline is built through careful stage scheduling, starting from the above mentioned ANSI-C representation of the functionality of each extended instruction. The dedicated programmable control unit manages the pipeline activity, by starting new PiCoGA operations or by stalling them when requested resources are not yet available. The control unit has a minimum granularity of one array row, but more than one PiCoGA row can be used to build a wider pipeline stage. In order to maintain a fixed clock frequency, cascaded RLCs have to be mapped to different pipeline stages. When a pipeline stage completes its computation, it produces a "token" which is sent both to predecessor and to successor nodes through dedicated programmable interconnection channels.

E. Digital System Performance

The digital system architecture discussed in this section has been implemented as a prototype chip (Fig. 16) [44] fabricated in 0.13μm, 1.2V, 6 metal layer CMOS technology provided by STMicroelectronics. The die size is 36mm^2 and contains 17.5M transistors. The SoC operates at a nominal frequency of 166MHz and fits in a 256-pin package with 151 I/Os. Static power consumption is 35mW, while average dynamic power consumption is 1.48mW/MHz for the whole system, except for the reconfigurable gate-array which consumes 100μW/MHz for each active row.
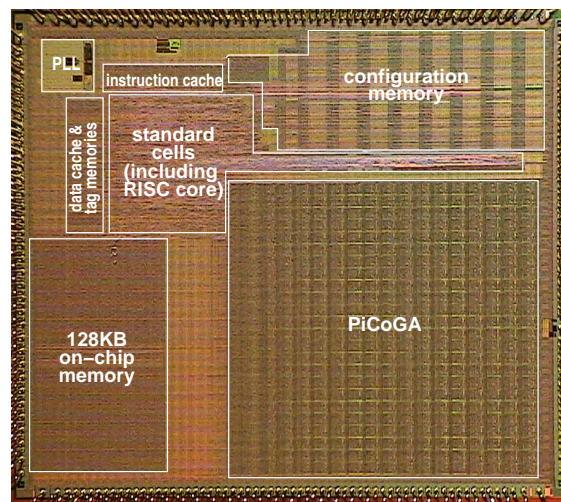

Fig. 16.   Digital system micro-photograph

1) Benchmarks: In order to prove the effectiveness of our reconfigurable digital processor for new generation cellular handsets, we first analyzed one of the most challenging wireless communication specifications, namely 3GPP. Of course, we cannot assume that digital architecture will be used only for telecommunication algorithms, since we expect that in the future the number of applications provided by network operators will grow very rapidly. We thus also evaluated the performance of the XiRisc architecture on multimedia applications.

TABLE III

PROTOTYPE CHIP CHARACTERISTICS

Parameter	Value
Technology	0.13μm CMOS process
	6 Metal Layers
Power Supply	1.2V
Clock Frequency	120MHz (WC-COM)
	166MHz (TYP)
Power consumption	Static: 35mW
	Dynamic (excluding PiCoGA): 1.48mW/MHz
	Dynamic (PiCoGA): 100μm/MHz per active row
SRAM Memory Size	Main memory: 128KB
	Instruction cache: 4KB
	Data cache: 4KB
	Tag memories: 1KB (x2)
	PiCoGA configuration cache: 64KB
Chip size	6x6 mm^2
PiCoGA size	11 mm^2
I/Os	151
Transistor count	17.5M

Tab. IV shows some results for different applications and algorithms, comparing the execution time and the energy consumption of the XiRisc with those of a DSP-like architecture with the same processor core, DSP-specific function units and cache memories, but with a single datapath.

In the telecom application field, we implemented a benchmark based on a turbo-decoder algorithm compliant with the 3GPP mobile communication specifications [45]. We chose a *Linear-log MAP* algorithm [42], [46], because it offers BER performances which are up to 0.5dB better than the more common *Max-log-MAP* algorithm with an increase in computational cost of about 40%. On a 640 bit block size, the proposed SoC requires 432 cycles/bit/iteration, corresponding to 384 kbps/iteration at a typical clock frequency of 166MHz. The same algorithm requires 3715 cycles/bit/iteration on a basic XiRisc processor, so that the resulting speed-up is 8.6x. The energy consumption is 1mJ (84% saving). Further optimizations are possible through manual optimizations at assembly level. For instance, with optimized scheduling, only 84 cycles/bit/iteration are required for the mathematical operations involved in the *Linear-log MAP* algorithm, providing a theoretical performance upper-bound of 2Mbps.

Another algorithm that is typically used for both channel coding environments and a variety of different applications is Reed Solomon (RS) error correction coding. The implementation of the RS encoder on a 239-byte message,

with 16 redundancy bytes, uses the PiCoGA device very efficiently, achieving an impressive 80x speedup. This is mainly due to the bit-level nature of the algorithm, which nicely fits the fine granularity of the reconfigurable array. The same considerations apply to the case of the DES encryption algorithm.

As a reference benchmark on multimedia applications, we present the results obtained on an MPEG-2 encoding, applied to a standard QCIF stream with a frame resolution of 176x144 pixels and half-pel precision. The reconfigurable architecture permits a 5x execution speed-up, thus allowing encoding of up to eight frames per second. The overall energy consumption to process a 12 frame stream is 718 mJ, corresponding to a 66% energy reduction compared to the traditional DSP. Setting a full-pel resolution, the frame rate increases to 12 frames/sec, as required by the H261 video compression standard.

Fig. 17 summarizes energy consumption contributions from the main architectural components and shows significant savings in energy consumption for both the processor and the bus, mainly due to the execution time decrease. The use of cache memories further reduces energy consumption of memory accesses by 90%. One notices that both energy and execution time are reduced very significantly over a broad range of applications.
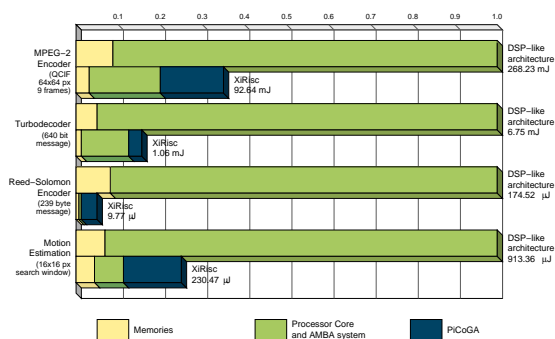

Fig. 17. Energy consumption for several DSP algorithms

TABLE IV

TIMING AND ENERGY CONSUMPTION PERFORMANCE FOR SOME DSP ALGORITHMS

Algorithm	Rows Occupation	Speed-up	Energy Saving
MPEG-2	23	5x	66%
Turbo-decoder	24	8.6x	84%
Reed Solomon Encoder	14	80x	94.5%
DES	5	13.5x	89%
CRC	11	4.3x	49%
Motion estimation	24	14.8	74.8%

IV. Conclusions

This paper has described the analog and digital baseband channels for a multi-standard reconfigurable terminal supporting GSM (with Edge), WCDMA (UMTS), WLAN and Bluetooth, developed within the "Enabling technologies for reconfigurable wireless terminals" project, funded by the Italian FIRB Project. The RF circuits of such a terminal are described in a companion paper. All the most important design issues, relating to the management of different standards in a reconfigurable transceiver, are analyzed. Experimental results on five test chips are reported to validate the solutions adopted both at the architecture and at the circuit level.

V. Acknowledgments

References

[1] J. Rogin, I. Kouchev, and Q. Huang, "A 1.5V 45mW Direct Conversion WCDMA Receiver IC in 0.13μm CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC '03)*, February 2003, pp. 268–493.

[2] J. Bouras, S. Bouras, T. Georgantas, N. Haralabidis, G. Kamoulakos, C. Kapnistis, S. Kavadias, Y. Kokolakis, P. Merakos, J. Rudell, S. Plevridis, I. Vassiliou, K. Vavelidis, and A. Yamanaka, "A Digitally Calibrated 5.15-5.825GHz Transceiver for 802.11a Wireless LANs in 0.18μm CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC '03)*, February 2003, pp. 352–498.

[3] K. Gulati and H. S. Lee, "A Low-Power Reconfigurable Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 12, pp. 1900–1911, December 2001.

[4] A. Stojcevski, J. Singh, and A. Zayegh, "CMOS ADC with Reconfigurable Properties for Cellular Handset," in *Proceedings of the IEEE International Workshop on Electronic Design, Test and Applications (DELTA '04)*, January 2004, pp. 103–107.

[5] B. J. Minnis and P. A. Moore, "A Highly Digitized Multimode Receiver Architecture for 3G Mobiles," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 3, pp. 637–653, May 2003.

[6] A. Dezzani and E. Andre, "A 1.2-V Dual-Mode WCDMA/GPRS ΣΔ Modulator," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC '03)*, February 2003, pp. 58–59.

[7] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, February 2005.

[8] N. J. Drew, D. Williams, M. Dillinger, P. Mangold, T. Farnham, and M. Beach, "Reconfigurable Mobile Communications: Compelling Needs and Technologies to Support Reconfigurable Terminals," in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '00)*, vol. 1, September 2000, pp. 484–489.

[9] X. Li, A. R. Bugeja, and M. Ismail, "A Fast and Accurate Calibration Method for High-Speed High-Resolution Pipeline ADC," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 2, May 2002, pp. 800–803.

[10] H. S. Lee, "A 12-bit 600kS/s Digitally Self-Calibrated Pipeline Algorithmic ADC," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 4, pp. 509–515, April 1994.

[11] E. B. Blecker, T. M. McDonald, O. E. .Erdogan, P. J. Hurst, and S. H. Lewis, "Digital Background Calibration of an Algorithmic Analog-to-Digital Converter Using a Simplified Queue," *IEEE Journal Solid-State Circuits*, vol. 38, no. 6, pp. 1812–1820, june 2003.

[12] H. S. Lee and B. S. Song, "Digital-Domain Calibration of Multistep Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 4, pp. 1679–1688, December 1992.

[13] X. Wang, P. J. Hurst, and S. H. Lewis, "A 12-bit 20-MS/s Pipelined ADC with Nested Digital Background Calibration," in *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC '03)*, September 2003, pp. 21–24.

[14] B. Murmann and B. E. Boser, "A 12b 75MS/s Pipeline ADC using Open-Loop Residue Amplification," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC '03)*, February 2003, pp. 328–497.

[15] E. J. Siragusa and I. Galton, "Gain Error Correction Technique for Pipelined Analogue-to-Digital Converters," *Electronics Letters*, vol. 37, no. 7, pp. 617–618, March 2000.

[16] J. Ming and S. Lewis, "An 8-Bit 80-MSample/s Pipeline Analog-to-Digital Converter with Background Calibration," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 10, pp. 1489–1497, October 2001.

[17] A. Bosi, A. Panigada, G. Cesura, and R.Castello, "An 80MHz 4x Oversampled Cascaded ΔΣ-Pipelined ADC with 75dB DR and 87dBSFDR," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC '05)*, February 2005, p. ???

[18] J. McNeil, M. Coln, and B. Larivee, "A Split-ADC Architecture for Deterministic Digital Background Calibration of 16b 1MS/s ADC," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC '05)*, February 2005.

[19] J. Li and U. K. Moon, "Background Calibration Techniques for Multistage Pipeline ADCs With Digital Redundancy," *IEEE Transactions on Circuits and Systems II*, vol. 50, no. 9, pp. 531–538, September 2003.

[20] U. K. Moon and B. S. Song, "Background Digital Calibration Techniques for Pipelined ADC's," *IEEE Transactions on Circuits and Systems II*, vol. 44, no. 2, pp. 102–109, February 1997.

[21] I. Galton, "Digital Cancellation of D/A Converter Noise in Pipelined A/D Converters," *IEEE Transactions on Circuits and Systems II*, vol. 47, no. 3, pp. 185–196, March 2000.

[22] A. Baschirotto, N. Ghittori, P. Malcovati, and A. Vigna, "Design Trade-Offs for a 10 Bit, 80MHz Current Steering Digital-to-Analog Converter," in *Proceedings of the IEEE Northeast Workshop on Circuits and Systems (NEWCAS '04)*, June 2004, pp. 249–252.

[23] Y. Cong and R. L. Geiger, "Formulation of INL and DNL Yield Estimation in Current Steering D/A Converters," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 3, May 2002, pp. 149–152.

[24] M. J. Pelgrom, A. C. Duinmaijer, and A. P. Welbers, "Matching Properties of MOS Transistor," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 10, pp. 1433–1439, October 1989.

[25] Tensilica Inc. [Online]. Available: http://www.tensilica.com

[26] H. Zhang, V. Prabhu, V. George, M. Wan, M. Benes, A. Abnous, and J. M. Rabaey, "A 1V Hetero-geneus Reconfigurable Processor IC for Baseband Wireless Applications," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC '00)*, February 2000, pp. 68–69.

[27] R. Razdan and M. Smith, "A High-Performance Microarchitecture with Hardware-Programmable Functional Units," in *Proceedings of the Annual International Symposium on Microarchitecture*, November 1994, pp. 172–180.

[28] P. Athanas and H. Silverman, "Processor Reconfiguration Through Instruction-Set Metamorphosis," *IEEE Computer*, vol. 26, no. 3, pp. 11–18, March 1995.

[29] R. Wittig and P. Chow, "OneChip: An FPGA Processor With Reconfigurable Logic," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, March 1996, pp. 126–135.

[30] T. Callahan, J. Hauser, and J. Wawrzynek, "The Garp Architecture and C Compiler," *IEEE Computer*, vol. 33, no. 4, pp. 62–69, April 2000.

[31] Atmel FPSLIC. [Online]. Available: http://www.atmel.com/products/FPSLIC

[32] Motorola MRC6011 Reconfigurable Compute Fabric (RCF) Device. [Online]. Available: http://e-www.motorola.com/files/if/cnb/ASICRCFWP_D.pdf

[33] M. Borgatti, F. Lertora, B. Foret, and L. Calì, "A Reconfigurable System Featuring Dynamically Extensible Embedded Microprocessor, FPGA, and Customizable I/O," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, pp. 521–529, March 2003.

[34] D. Patterson and J. Hennessy, *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 1991.

[35] A. Lodi, M. Toma, F. Campi, A. Cappelli, R. Canegallo, and R. Guerrieri, "A VLIW Processor with Reconfigurable Instruction Set for Embedded Applications," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1876–1886, November 2003.

[36] P. Chang, S. Mahlke, W. Chen, N. Water, and W. Hwu, "IMPACT: An Architectural Framework for Multiple-Instruction-Issue Processors," in *Proceedings of the 18th Annual Int'l Symposium on Computer Architecture*, May 1991, pp. 266–275.

[37] A. Lodi, M. Toma, F. Campi, A. Cappelli, R. Canegallo, and R. Guerrieri, "A Pipelined Configurable Gate Array for Embedded Processors," in *Proceedings of the ACM/SIGDA International Symposium on FPGA*, February 2003, pp. 21–30.

[38] A. DeHon, "DPGA-Coupled Microprocessors: Commodity ICs for the Early 21st Century," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, April 1994, pp. 31–39.

[39] S. Trimberger, D. Carberry, A. Jhonson, and J. Wong, "Configuration Caching Techniques for FPGA," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, April 2000, pp. 34–49.

[40] S. Hauck, K. C. Compton, and Z. Li, "A Time Multiplexed FPGA," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, April 1997, pp. 34–40.

[41] A. Lodi, R. Giansante, C. Chiesa, L. Ciccarelli, F. Campi, and M. Toma, "Compact Buffered Routing Architecture," in *Lecture Notes in Computer Science, Field-Programmable Logic and Applications*, vol. 3203. Springer-Verlag, September 2003, pp. 179–188.

[42] A. La Rosa, L. Lavagno, and C. Passerone, "Software Development for High-Performance, Reconfigurable, Embedded Multimedia Systems," *IEEE Design and Test of Computers*, vol. 22, no. 1, pp. 28–38, January 2005.

[43] C. Mucci, C. Chiesa, A. Lodi, M. Toma, and F. Campi, "A C-based Algorithm Development Flow for a Reconfigurable Processor Architecture," in *Proceedings of the International Symposium on System-on-Chip*, November 2003, pp. 19–21.

[44] M. Bocchi, C. De Bartolomeis, C. Mucci, F. Campi, A. Lodi, M. Toma, R. Canegallo, and R. Guerrieri, "A XiRisc-based SoC for Embedded DSP for Embedded DSP Applications," in *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC '04)*, October 2004, pp. 595–598.

[45] *3G TS.25.212 V5.1.0 Multiplexing and Channel Coding (FDD)*, Technical Specification Group Radio Access Network, 3rd Generation Partnership Project, 2002.

[46] M. C. Valenti and J. Sun, "The UMTS Turbo Code and Efficient Decoder Implementation Suitable for Software-Defined Radios," *International Journal of Wireless Information Networks*, vol. 8, no. 4, pp. 203–216, April 2001.