BRIDGING THE ENERGY GAP IN SIZE, WEIGHT AND POWER CONSTRAINED SOFTWARE DEFINED RADIO: AGILE BASEBAND PROCESSING AS A KEY ENABLER

Bruno Bougard, Min Li, David Novo, Liesbet Van der Perre, Francky Catthoor

IMEC, Leuven, Belgium

{bougardb, minli, novo, vdperre, catthoor}@imec.be

Abstract - The diversity and evolution of wireless communication standards are fast-pacing. This requires a wide-variety of baseband implementations within short time-to-market. Besides, always deeper submicron technology significantly increase design cost. This yields an increasing need for using reconfigurable or programmable solutions for an always larger part of wireless modems. Mapping the whole baseband functionality on a programmable architecture, as foreseen in tier-2 SDR, will become a must in future implementation. In handhelds where the multi-mode trend adds extra needs for programmability, the energy efficiency of SDR baseband is however a major concern. New processor architectures with major improvements on energy efficiency (GOPS/mW) are emerging but are still not sufficient to catch the continuously increasing complexity of wireless physical layers within the shrinking energy budget. To enable SDR in size, weight and power constrained devices, innovation is also needed at the software side. Specifically, a thorough architecture-aware algorithm implementation methodology is needed for the baseband signal processing functions, which account for most of the SDR computational complexity. We present the premise of such a methodology and illustrate its effectiveness on the design of key kernels from present and future wireless baseband systems.

***Keywords** – Portable Radio Communication, Software-Defined Radio, Energy Awareness*

I INTRODUCTION

Nowadays handhelds devices are integrating an increasing variety of wireless communication and connectivity standards, which depicts a multitude of operations modes. This diversity, combined with the increasing cost of silicon implementation, claims for flexible implementations wherever possible. The Tier-2 Software Defined Radio (SDR) approach, where the whole baseband functionality is run on programmable architectures, is an attractive way to obtain that flexibility. However, SDR typically comes with a lower energy efficiency than equivalent hardwired implementation. This energy gap remains to be filled in order to make SDR appropriate for size, weight and power constrained systems as handhelds are. Most SDR researches focus on reducing that gap by mean of more efficient processsor architectures. In [1-4], several processor- or multi-processor platform architectures are proposed with power consumption compatible with handhelds. Their performance however still prohibits the direct implementation of the most advanced standards (IEEE802.11n, 3GPP-LTE), with 100+Mbps throughputs and an increasing number of operations per bit.

Besides computer architectures, innovation is also needed in the way baseband processing is handled in software, which has strong relations with signal processing. Typically, baseband signal processing algorithms are designed and optimized with a dedicated hardware (ASIC) implementation in mind, which requires regular and manifest computation structures as well as simple control flow, maximum functional blocks reuse and minimum data word width. Programmable architectures have other requirements. Typically, they can accommodate more complex control flows. Functional reuse is not a must since not the entire area but only the instruction memory footprint benefit from it. However, they have more limitations in terms of maximal computational complexity and energy efficiency. Moreover, data types must be aligned. Taking these characteristics into account when developing the baseband algorithms is a key to enable energy-efficient SDRs. We propose the premise of such an architecture-aware algorithm implementation methodology. After developing the approach in section II, we discuss two case studies. The first (section IV) focuses on algorithmic transformation, rather independently of their implementation. The second case (section V) focuses on fixed-point implementation. Prior, the main characteristics of the considered processor architecture is summarized (Section III).

II APPROACH

As aforementioned, contrarily to traditional ASIC implementations, programmable baseband platforms have more expensive computational resource. This is mainly due to the fact that, a priori, for each operation, an instruction must be fetched and an expensive control structure is activated to decode and execute it. This overhead is however significantly mitigated if the operations can be arranged in such a way that data and/or instruction level parallelism can be exploited. Therefore, in our approach, ***software-pipelining*** [5] is first extensively applied to enabled maximum ***instruction level parallelism*** (ILP). ***Data level parallelism*** (DLP) is then enabled keeping explicit the vector formulation that is typical of communication algorithms. The limiting factor becomes the memory accesses with similar constraints for programmable or ASIC implementation on condition that an energy-efficient, partly distributed memory organization is used.

On the other end, programmable architectures accommodate more easily agile implementations with more complex control structures. This feature is an asset when considering the wide-range dynamics existing in wireless communication systems. First, the wireless baseband environment is inherently time varying, this includes channel conditions, interference, spectrum utilization and so on. The environment dynamics are traditionally exploited to improve performance and throughput. Complementarily, it can be utilized to reduce the average processing load. In addition, dynamics also exists in user requirements. In practical systems, the data rate, the tolerable error (precision), the latency, and so on, are context dependent. Not all applications require the same communication quality-of-service (QoS) to deliver satisfactory user experiences. For a given application, these QoS requirements are also varying in time. These observations have triggered research toward joint energy-QoS management with development of energy-scalable analog and digital hardware structures [6]. As they allow more complex control structure, that approach gets lot more value when programmable architectures are considered. The dynamics are then exploited through a fine-grain closed-loop adaptation of the algorithm characteristics and their low-level (software) implementation to the varying requirements. If we categorize the complexity as (1) gross computation and memory complexity, (2) structural complexity, which is determined by the heterogeneity and control flows. The main idea is to increase the structural complexity while aiming at reducing the gross average computation and memory complexity. That reduction eventually translates in potentially (very) large energy benefits, the cost being an increased program footprint.

In particular, for a given functionality, several software implementation with different complexity/performance tradeoffs are still possible. the proposed approach aims at identifying the most appropriate algorithm implementation, characterized by a given ***precision*** and a given ***computation load***, for different predefined operation scenarios (Fig. 1). A ***monotonically increasing relation between precision and computation load*** is desired. Capitalizing again on the characteristics of programmable architectures, one can switch between these implementations in a few cycles (direct function call latency), enabling quick adaptation.

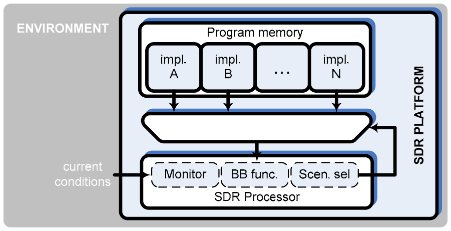

Figure 1 Scenario-based agile baseband processing

III Programmable baseband architecture

Very long instruction word (VLIW) instruction set processors with SIMD (Single Instruction – Multiple Data) functional units are mostly considered to exploit the data level and instruction level parallelism with limited instruction fetching overhead [1,2,4]. Besides, data flow dominance is often exploited in coarse grain reconfigurable arrays (CGA) [7,8]. We have formerly proposed a hybrid CGA-SIMD processor made of 16 densely interconnected 64-bit 4-way SIMD units with shared and distributed register-files (Figure 2) [9]. The CGA is associated with a 4-bank data scratchpad (L1). It can be programmed from C based on the DRESC compiler framework [10]. A limited number of units can be operated in VLIW mode, accepting arbitrary C-compiled code (glue code) fetched through a 32K 128-bit wide instruction cache. When in array mode, C-compiled DSP kernels are executed while keeping configuration into local buffers that are configured through direct memory access (DMA). In this work, we used a 32-bit derivative of that architecture with 2x16-bit and 4x8bit SIMD support.

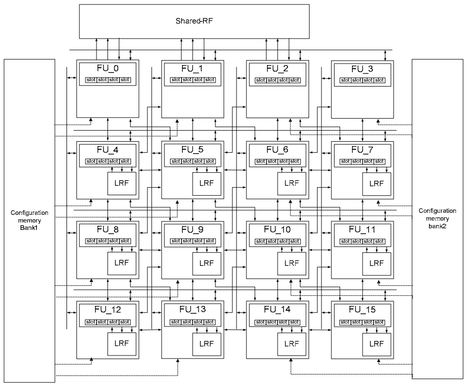

Figure 2 SDR Processor Core

IV Case 1: OFDMA (De)modulation

A) Overview

(MIMO-)OFDMA transmissions have been adopted in many emerging standards, such as 3GPP LTE and IEEE 802.16e. Comparing to OFDM systems, in OFDMA each user occupies only part of the available sub-carriers. Therefore, a much larger FFT/IFFT is required, compared to the number of allocated sub-carriers. For instance, in the 20 MHz mode of 3GPP LTE, if 12 users equally share the sub-carriers, the FFT/IFFT size is 2048 whereas only 100 sub-carriers are allocated to each user. Contrarily, in IEEE 802.11g for instance, the FFT/IFFT size is 64, 48 sub-carriers are allocated to the same user. Hence, in OFDMA, the complexity of the FFT/IFFT based (de)modulator is significant and the energy-efficient implementation is critical. In this section, we illustrate how a precision/load scalable implementation of the FFT/IFFT function can be achieved both in the receiver (subsection B) and in the transmitter (subsection C) and how the scalability impacts the average energy consumption.

B) OFDMA Demodulator Based on PFFT

First we propose modifications to the demodulator so that it can scale to match user requirements. As mentioned above, each user in a cell communicates on only part of the sub-carriers. Hence, for a specific user, in the demodulator only part of the FFT outputs is useful. The FFT with only partial output is called Partial FFT (PFFT). By pruning the useless dataflow, the PFFT can potentially achieve a significant speedup for demodulations.

Although theoretical aspects of the PFFT have been thoroughly studied in past three decades [11-12], energy- and cost-efficient implementations were rarely reported. An important obstacle is the highly irregular dataflow and the associated control flow, which are however well-fitted for programmable architectures.

Our case study targets PFFT-based OFDMA demodulator implementation on ILP architectures as mentioned in Section III. Constraints and opportunities of algorithms and architecture are analyzed and exploited to enable efficient software-pipelining. After exploration of dataflow variants, we adopted a multiphase partitioning as in Figure 3, bringing heterogeneous control structures and heterogeneous software pipelining schemes to minimize control overheads and to maximize parallelism. With a representative 3GPP LTE demodulation benchmarks, our work reduces the cycle-counts from 20.5% to 85.2% and the data memory access from 11.2% to 82.4% depending on the subcarriers allocation. Intrinsically, the amount of processing scales with the number of subcarriers to compute, which enable context adaptation.

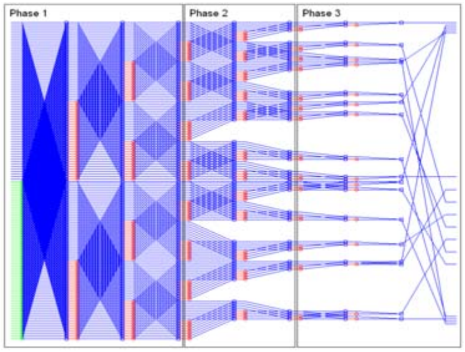

Figure 3 Multi-phase portioned PFFT scheme. The PFFT is decomposed into three phases, heterogeneous control structures and software pipelining schemes are combined to preserve the efficiency of parallel execution.

C) Quality/Energy Scalable OFDMA Modulator

At the transmit side, we proposed a novel OFDMA modulation scheme with accuracy/load scalability. The proposed modulator can scale the modulation accuracy (Relative Constellation Error, RCE) to the link requirements. This way, the computation load and associated energy-consumption can be reduced whereas the predefined constellation requirements are still guaranteed. The scalability is achieved by replacing the original large-size IFFT from the direct implementation by a scalable approximation consisting of: (1) a variable-size IFFT; (2) a variable-size linear-interpolator; (3) a signal rotator (Figure 4).

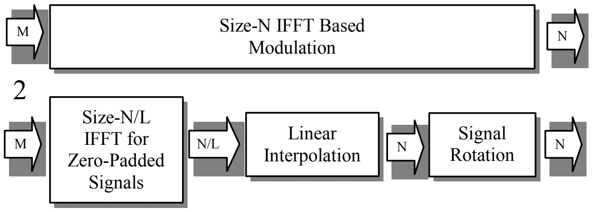

Figure 4 Interpolated IFFT for scalable OFDMA modulation

The novel scheme can be easily mapped on the targeted ILP architecture, with highly efficient software-pipelining. The proposed scheme has been evaluated in the context of an IEEE 802.16e MIMO-OFDMA transmitter. For each signaling mode (modulation, code rate), an implementation with maximum interpolation factor (the ratio between the numbers of interpolated and non-interpolated outputs) under RCE constraint is chosen. The RCE constraint is set according to the 802.16e specs. The achieved load reduction is depicted in Table 1.

Table 1. Energy Reduction Rate in Scalable Modulator

Modu. Scheme	Coding Rate	Maximal RCE	Interp. Factor	LoaReduction
BPSK	½	-13.0 dB	16	92%
QPSK	½	-16.0 dB	16	92%
QPSK	¾	-18.5 dB	16	92%
16QAM	½	-21.5 dB	8	84%
16QAM	¾	-25.0 dB	8	84%
64QAM	2/3	-28.5 dB	8	84%
64QAM	¾	-31.0 dB	4	68%

V CASE 2: SCALABLE DATA-FORMAT IN OFDM

In the previous section, we have shown that precision/load scalability can be achieved by selecting particularly designed algorithms to implement key functional building block. The particularities of the architecture have been taken into account by enabling high ILP through software pipelining. However, DLP has not yet been considered. Fixed-point refinement was conducted at design-time fixing all word-width as the native processor word-width. As an alternative, in this section, a scenario-oriented fixed-point refinement is considered. Situations where the application exhibits a different tolerance to the quantization noise are identified. At design-time, separated fixed-point refinements are performed for each of these situations (scenarios), resulting in multiple software implementations. At run-time, the actual scenario that best suits the current working conditions is detected and the corresponding implementation is selected by a simple controller.

As an example, an OFDM system according to the IEEE 802.11g standard is considered. Similarly to what we did in the previous case study, scenarios corresponding to the different signaling mode are considered. The varying noise-robustness of the different signaling scheme is used to also adapt the fixed-point accuracy constraints.

Typically, the quantization of a signal is modeled by the sum of this signal and a stationary, not correlated, uniformly-distributed random variable. Thus, the effect of refining an ideal (infinite precision) data flow into a fixed-point implementation can be modeled as the initial data flow with ideal operators fed with the sum of the ideal operands and a noise component (quantization noise). The effect of the noise can be modeled with an equivalent noise-source at the inputs or outputs of the data-flow, assumed to belong to the channel. Intuitively, transmission modes that tolerate higher levels of channel noise should also be able to accept higher levels of quantization noise. These modes will hence require fewer bits to maintain the necessary accuracy.

As the end goal is to achieve precision/load scalability and because of the large amount of data parallelism in OFDM (similar processing is applied independently to all the su-carriers), we leverage on the SIMD feature to translate reduced bit-width into a lower computation load. In particular, we exploit the fact that multiple data (sub-words) can be packed together and operated on as a single word. Figure 5 sketches the packing of 4 sub-words, which correspond to different OFDM symbols, into a single operand. The size of these sub-words is variable and can be selected from a discrete set (typically of powers-of-2 for addressing reasons). The different sub-word configurations can share the same hardware operators, which are configured depending on the current sub-word size (e.g. cutting the carry propagation in an adder). The cost in energy and time associated with the operation (operand load, execution, result storage) is shared by all the sub-words. Consequently, the fewer the bits required to represent the data, the more the data we can pack together and the cheaper the processing per sub-word.

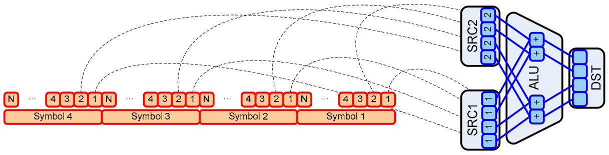

Figure 5: Packing of 8 different sub-carriers into 2 operands to perform 4 additions in parallel with the cost of a single addition.

The inner modem of our OFDM example is implemented considering 3 different cases: a baseline with traditional all-16 bit implementation; an implementation with 0.5 dB extra quantization noise tolerance and an implementation with 2dB tolerance. The throughput envelopes of the 3 implementations are plot in Fig. 6a. The latter envelopes correspond to the maximum achievable throughput on top of the medium access control layer, under different SNR conditions, considering all the modes. In addition, their corresponding energy per transmitted bit, estimated by detailed gate-level simulations of the target architecture, is plot in Fig. 6b. One can observe that the low rate configurations consume more energy than the high rate ones. This can be easily understood since for transmitting the same amount of information, the low rate configurations need to send more OFDM symbols. Consequently the processor needs to process during a longer time consuming more energy per bit of information.

When extra quantization tolerance is allowed (e.g. less than 0.5 dB), negligible system performance loss is observed. However the energy per bit of the lower rate configurations is considerably reduced. For instance, in the region from 0-6 dB of the SISO case (see Fig. 6) the 16bits sub-word implementation can be reduced to 8bits, resulting in a 43% energy saving. When more quantization tolerance is allowed (less than 2 dB), the performance starts to suffer. As an indicator, when less than 2dB degradation is allowed and the receiver works with a 3dB SNR, the maximum throughput drops by 53%. However a 4bits implementation can now be accommodated, which reduces the energy by 66%. In this case some performance is traded off for energy.

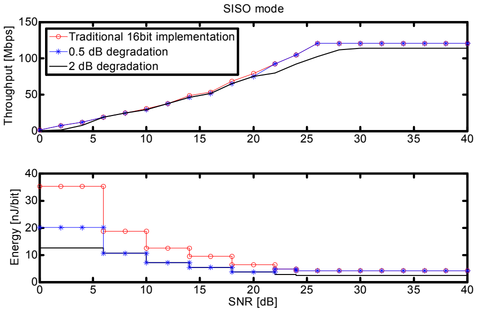

Figure 6: Curves of a OFDM communication system: throughput (a) and energy consumption (b).

VI CONCLUSIONS

SDR has high change become the only valid approach for cost effective silicon implementation of digital baseband processing in future multi-mode multi-standards handhelds. However, although significant processor architecture improvement can be reported, there remains a gap in energy efficiency between SDR and traditional ASIC implementation. Also, SDR platform gross performances are still not sufficient to sustain the most recent highest throughput standards. To bridge this gap, we have presented and illustrated the premise of a systematic architecture-aware algorithm implementation approach. The key idea is to leverage on the fitness of programmable architecture for more agile implementation to increase the system ability to continuously adapt to the varying user and environment requirements. This eventually reduces the average computation load and thereby the power.

REFERENCES

[1] K. Van Berkel et al., "Vector Processing as an Enabler for Software Defined Radio in Handsets for 3G+WLAN Onwards," *SDR Forum Tech. Conf.*, 2004, pp. 125-130.

[2] J. Glossner et al, "A software-defined communications baseband design," *Communications Magazine, IEEE*, Vol. 41, pp. 120-128, 2003.

[3] A. Nilsson and Dake Liu, "Area Efficient Fully Programmable Baseband Processsors, Proc. of the 7th SAMOS Workshop, pp. 333-343, Samos, Greece, June 2007

[4] L. Yuan et al., "SODA: A Low-power Architecture For Software Radio," pp. 89-1012006.

[5] V. Allan et al., "Software Pipelining". *ACM Computing Surveys*, 27(3), September 1995

[6] A. Dejonghe et al., "Green Reconfigurable Radio Systems", IEEE Signal Proc. Mag., pp 90-101, May 2007

[7] A. Lodi et al., "XiSystem: a XiRisc-based SoC with reconfigurable IO module," *Solid-State Circuits, IEEE Journal of*, Vol. 41, pp. 85-96, 2006

[8] H. Singh et al., "MorphoSys: an integrated reconfigurable system for data-parallel and computation-intensive applications," *IEEE Trans. on Computers* (49), pp. 465-481

[9] B. Bougard, "A Coarse-Grained Array based Baseband Processor for 100Mbps+ Software Defined Radio", Proc. DATE, May 2008

[10] B. Mei et al., "Exploiting loop-level parallelism on coarse-grained reconfigurable architectures using modulo scheduling," *IEEE Proc. on Computers and Digital Techniques*, Vol. 150, pp. 255-261, 2003.

[11] J. D. Markel, FFT pruning, *IEEE Trans. Audio Electroacoustic*, vol. 19, pp. 305-311, Dec. 1971.

[12] H. V. Sorensen and C. S. Burrus, Efficient computation of the DFT with only a subset of input or output points, *IEEE Trans. Signal Process.*, vol. 41, pp. 1184-1200, Mar. 1993