VLSI Design

Flexible Radio Design: Trends and Challenges in Digital Baseband Implementation

Guest Editors: Guido Masera, Amer Baghdadi, Frank Kienle, and Christophe Moy

Flexible Radio Design: Trends and Challenges in Digital Baseband Implementation

VLSI Design

Flexible Radio Design: Trends and Challenges in Digital Baseband Implementation

Guest Editors: Guido Masera, Amer Baghdadi, Frank Kienle, and Christophe Moy

Editorial Board

Contents

Hindawi Publishing Corporation
VLSI Design
Volume 2012, Article ID 549768, 2 pages
doi:10.1155/2012/549768

Editorial

Flexible Radio Design: Trends and Challenges in Digital Baseband Implementation

Guido Masera,[1] Amer Baghdadi,[2] Frank Kienle,[3] and Christophe Moy[4]

[1] *Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy*
[2] *Telecom Bretagne/Lab-STICC, Technopôle Brest-Iroise, CS 83818, 29238 Brest, France*
[3] *Microelectronics System Design, Technical University of Kaiserslautern, Postfach 3049, 67653 Kaiserslautern, Germany*
[4] *Supélec/IETR, Campus de Rennes, Avenue de la Boulaie, CS 47601, 35576 Cesson-Sévigné Cedex, France*

Correspondence should be addressed to Guido Masera, guido.masera@polito.it

Received 1 August 2012; Accepted 1 August 2012

Fourth-generation communications systems call for a high amount of computational power due to multiantenna and multimode features. The level of flexibility required is growing rapidly with the number of modes to be supported for a single protocol and the number of protocols to be supported by a single receiver. Such high level of flexibility becomes a key feature of new and legacy radio applications in many domains (military radio, broadcast systems, aeronautic communications, etc.), which call for adopting a software-defined radio (SDR) approach, or even for incorporating additional adaptive capabilities, such as suggested by cognitive radio (CR) research.

In general, the design of flexible base-band platforms raises several critical problems, including the high level of required performance, the dissipated power, and the reconfiguration process itself. Several alternatives have been partially explored to implement flexible base-band building blocks and a lot of research is still required to bring efficiency into programmable platforms.

The design of flexible channel decoders for low density parity check codes is addressed in two papers. M. Awais and C. Condo provide a rich overview of state-of-the-art decoders able to deal with multiple LDPC codes adopted in communication standards such as DVB-S2, IEEE 802.11n (WiFi), IEEE 802.3an (10GBASE-T), and IEEE 802.16e (WiMAX). Similarly, P. Schläfer et al. presents a systematic investigation of the design space for flexible multigigabit LDPC applications, such as ultrawideband communications (WiMedia), wireless personal area networks (IEEE 802.15.3c), and gigabit wireless local area networks (IEEE 802.11ad). Both papers give extensive comparisons among already available results and also introduce new implementation examples proposed by the authors. Interestingly, from the reading of the two papers, it can be understood that incorporation of flexibility in LDPC decoding architectures introduces different constraints in medium throughput and multigigabit throughput applications.

The use of two popular soft-input soft-output detectors for multiple-input multiple-output (MIMO) communications is analysed by C. Gimmler-Dumont et al. in the context of both open- and closed-loop architectures. A depth-first sphere detector and a breadth-first fixed effort detector are deeply studied and compared in terms of communications performance and implementation complexity. The work shows that the fixed effort detector is advantageous in a throughput centric scenario, where very high-processing speed has to be ensured at moderate communications performance. On the contrary, the sphere detector offers much higher efficiency in a communication centric scenario, where very low-error rate is required.

Two further papers of this selection deal with the digital and analog front-ends, respectively. In particular, K. Grati et al. propose efficient implementations for FIR decimation filters to be used in multistandard receiver designs. Polyphase decomposition has been shown to provide a relevant reduction in the dissipated dynamic power with respect to the direct form implementation and this advantage is obtained at the cost of an increased static power consumption.

Finally, an overview of the key technology challenges in the development of RF front-end, and analog-to-digital and digital-to-analog interfaces for cognitive radio system is presented by V. T. Nguyen et al.

Guido Masera
Amer Baghdadi
Frank Kienle
Christophe Moy

Hindawi Publishing Corporation
VLSI Design
Volume 2012, Article ID 730835, 16 pages
doi:10.1155/2012/730835

Review Article

Flexible LDPC Decoder Architectures

Muhammad Awais and Carlo Condo

Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy

Correspondence should be addressed to Carlo Condo, carlo.condo@polito.it

Received 4 November 2011; Revised 14 February 2012; Accepted 22 February 2012

Academic Editor: Amer Baghdadi

Flexible channel decoding is getting significance with the increase in number of wireless standards and modes within a standard. A flexible channel decoder is a solution providing interstandard and intrastandard support without change in hardware. However, the design of efficient implementation of flexible low-density parity-check (LDPC) code decoders satisfying area, speed, and power constraints is a challenging task and still requires considerable research effort. This paper provides an overview of state-of-the-art in the design of flexible LDPC decoders. The published solutions are evaluated at two levels of architectural design: the processing element (PE) and the interconnection structure. A qualitative and quantitative analysis of different design choices is carried out, and comparison is provided in terms of achieved flexibility, throughput, decoding efficiency, and area (power) consumption.

1. Introduction

With the word flexibility regarding channel decoding, we mean the ability of a decoder to support different types of codes, enabling its usage in a wide variety of situations. Much research has been done in this sense after the great increase in number of standards, standard complexity, and code variety witnessed during the last years. Next-generation wireless standards such as DVB-S2 [1], IEEE 802.11n (WiFi) [2], IEEE 802.3an (10GBASE-T) [3], and IEEE 802.16e (WiMAX) [4] feature multiple codes (LDPC, Turbo), where each code comes with various code lengths and rates. The necessity for flexible channel decoder intellectual properties (IPs) is evident and challenging due to the often unforgiving throughput requirements and narrow constraints of decoder latency, power, and area.

This work gives an overview of the most remarkable techniques in context of flexible channel decoding. We will discuss design and implementation of two major functional blocks of flexible decoders: processing element (PE) and interconnection structure. Various design choices are analyzed in terms of achieved flexibility, performance, design novelty, and area (power) consumption.

The paper is organized as follows. Section 2 provides a brief introduction to LDPC codes and decoding. Section 3 gives an overview of flexible LDPC decoders classifying them on the basis of some important attributes for example, parallelism, implementation platforms, and decoding schedules. Sections 4 and 5 are dedicated to PE and interconnection structure, respectively, where we depict various design methodologies and analyze some state of the art flexible LDPC decoders. Finally, Section 6 draws the conclusions.

2. LDPC Decoding

2.1. Introduction. LDPC codes [5] are a special class of linear block codes. A binary LDPC code is represented by a sparse parity check matrix $\mathbf{H}$ with dimensions $M \times N$ such that each element h_{mn} is either 0 or 1. N is the length of the codeword, and M is the number of parity bits. Each matrix row $\mathbf{H}_{(i,(1\leq j\leq N))}$ introduces one parity check constraint on the input data vector $x = \{x_1, x_2, \ldots, x_N\}$:

$$\mathbf{H_i} \cdot \mathbf{x^T} = \mathbf{0} \bmod \mathbf{2}. \quad (1)$$

The complete $\mathbf{H}$ matrix can best be described by a Tanner graph [6], a graphical representation of associations between code bits and parity checks. Each row of $\mathbf{H}$ corresponds to a check node (CN), while each column corresponds to a variable node (VN) in the graph. An edge e_{ji} on the Tanner Graph connects a VN_j with CN_i only if the corresponding element h_{ij} is a1 in $\mathbf{H}$. If the number of edges entering in

a node is constant for all nodes of the graph, the LDPC code is called regular, being otherwise irregular in case of variable node degree. Irregular LDPC codes yield better decoding performance compared to regular ones.

Next-generation wireless communication standards adopt structured LDPC codes, which hold good interconnection, memory and scalability properties at the decoder implementation level. In these codes, the parity check matrix $\mathbf{H}$ is associated to a $\mathbf{H}_{\text{BASE}}$ matrix, as defined in [7]:

$$\mathbf{H}_{\text{BASE}} = \begin{bmatrix} \Pi_{0,0} & \Pi_{0,1} & \dots & \Pi_{0,N_b} \\ \Pi_{1,0} & \Pi_{1,1} & \dots & \Pi_{1,N_b} \\ \vdots & \vdots & \ddots & \vdots \\ \Pi_{M_b,0} & \Pi_{M_b,1} & \dots & \Pi_{M_b,N_b} \end{bmatrix}. \tag{2}$$

$\mathbf{H}_{\text{BASE}}$ has M_b block rows and N_b block columns; it is expanded, in order to generate the $\mathbf{H}$ matrix, by replacing each of its entries $\Pi_{i,j}$ with a $z \times z$ permutation matrix, where z is the expansion factor. The permutation matrix can be formed by a series of right shifts of the $z \times z$ identity matrix according to a determined shifting factor, equal to the value $\Pi_{i,j}$. The same base matrix is used as a platform for all the different code lengths related to a selected code rate: implementation of a full-mode decoder is thus a challenging task, due to huge variations in code parameters. For example, current IEEE 802.16e WiMAX standard features four code rates, that is, 1/2, 2/3, 3/4, and 5/6 with $\mathbf{H}_{\text{BASE}}$ matrices of size 12×24, 8×24, 6×24, and 4×24, respectively. Each code rate comes with 19 different codeword sizes ranging from 576 bits ($z = 24$) to 2304 bits ($z = 96$), with granularity of 96 bits ($\Delta z = 4$).

Algorithm 1 (The Standard TPMP Algorithm).

(1) *Initialization: For* $j \in \{1, \dots, N\}$

$$\alpha_{i,j}^{0} = \ln \frac{P\left(VN_j = 0 \mid y_j\right)}{P\left(VN_j = 1 \mid y_j\right)} = \frac{2y_j}{\sigma^2}. \tag{3}$$

(2) *CN Update Rule:* $\forall CN_i,\ i \in \{1, \dots, M\}$ *do*

$$\beta_{i,j}^{n} = \operatorname{sgn}\beta_{i,j}^{n} \cdot \left|\beta_{i,j}^{n}\right|, \tag{4}$$

$$\begin{aligned} \operatorname{sgn}\beta_{i,j}^{n} &= \prod_{j' \in \mathcal{N}(i) \setminus j} \operatorname{sgn}\left(\alpha_{ij'}^{(n-1)}\right), \\ \left|\beta_{i,j}^{n}\right| &= \underset{j' \neq j}{\otimes} \left(\alpha_{i,j'}^{n-1}\right). \end{aligned} \tag{5}$$

(3) *VN Update Rule:* $\forall VN_j,\ j \in \{1, \dots, N\}$ *do*

$$\alpha_{i,j}^{n} = \alpha_{i,j}^{0} + \sum_{i' \in \mathcal{M}(j) \setminus i} \beta_{i',j}^{n}. \tag{6}$$

(4) *Decoding: For each bit, compute it's a posteriori LLR*

$$\alpha_{j}^{n} = \alpha_{i,j}^{0} + \sum_{i' \in \mathcal{M} \setminus (j)} \beta_{i',j}^{n}. \tag{7}$$

Estimated codeword is $\widehat{C} = (\widehat{c}_1, \widehat{c}_2, \dots, \widehat{c}_N)$, where element $\widehat{c}_j$ is calculated as

$$\widehat{c}_j = \begin{cases} 0 & \text{if } \alpha_j^n > 0 \\ 1 & \text{else.} \end{cases} \tag{8}$$

If $\mathbf{H}(\widehat{C})^T = 0$, then stop, with correct codeword $\widehat{C}$.

2.2. LDPC Decoding Algorithms. The nature of LDPC decoding algorithms is mainly iterative. Most of these algorithms are derived from the well-known belief propagation (BP) algorithm [5]. The aim of BP algorithm is to compute the a posteriori probability (APP) that a given bit in the transmitted codeword $c = [c_0, c_1, \dots, c_{N-1}]$ equals 1, given the received word $y = [y_0, y_1, \dots, y_{N-1}]$. For binary phase shift keying (BPSK) modulation over an additive white Gaussian noise (AWGN) channel with mean 1 and variance σ^2, the reliability messages represented as logarithmic likelihood ratio (LLR) are computed in two steps: (1) check node update and (2) variable node update. This is also referred to as two-phase message passing (TPMP). For nth iteration, let $\alpha_{i,j}^{n}$ represent the message sent from variable node VN_j to check node CN_i, $\beta_{i,j}^{n}$ represent the message sent from CN_i to VN_j, $\mathcal{M}(j) = \{i : \mathbf{H}_{ij} = 1\}$ is the set of parity checks in which VN_j participates, $\mathcal{N}(i) = \{j : \mathbf{H}_{ij} = 1\}$ the set of variable nodes that participate in parity check i, $\mathcal{M}(j) \setminus i$ the set $\mathcal{M}(j)$ with check CN_i excluded, and $\mathcal{N}(i) \setminus j$ the set $\mathcal{N}(i)$ with VN_j excluded. The standard TPMP algorithm is described in Algorithm 1.

As given in (4), the CN update consists of sign update and magnitude update, where the latter depends on the type of decoding algorithm, of which several are commonly used (Table 1). The sum product (SP) algorithm [8] gives near-optimal results; however, the implementation of the transcendental function $\Phi(x)$ requires dedicated LUTs, leading to significant hardware complexity [9]. Min-Sum (MS) algorithm [10] is a simple approximation of the SP: its easy implementation suffers an 0.2 dB performance loss compared to SP decoding [11]. Normalized Min-Sum (NMS) algorithm [12] gives better performance than MS by multiplying the MS check node update by a positive constant λ_k, smaller than 1. Offset Min-Sum (OMS) is another improvement of standard MS algorithm which reduces the reliability values β_{ij}^{n} by a positive value β: for a quantitative performance comparison for different CN updates, refer to [13, 14].

2.3. Layered Decoding of LDPC Codes. Modifying the VN update rule (6):

$$\alpha_{i,j}^{n} = \alpha_{j}^{n} - \beta_{i,j}^{n}, \tag{9}$$

we can merge the CN and VN update rules into a single operation, where the CN messages $\beta_{i,j}^{n}$ are computed from $\alpha_{j}^{(n-1)}$ and $\beta_{i,j}^{(n-1)}$. This technique is called layered decoding [15]. Layered decoding considers the $\mathbf{H}$ matrix as a concatenation of l layers (block rows) or constituent subcodes, that is, $\mathbf{H}^{\mathbf{T}} = [\mathbf{H}_{\mathbf{1}}^{\mathbf{T}}\mathbf{H}_{\mathbf{2}}^{\mathbf{T}}, \dots, \mathbf{H}_{\mathbf{l}}^{\mathbf{T}}]$, where the column weight

Table 1: Check node update for LDPC decoding algorithms.

Algorithm	Formulation: $\underset{j' \neq j}{\otimes} \left(\alpha_{i,j'}^{(n-1)}\right)$
SP	$\Phi\left(\sum_{j' \in \mathcal{N}(i) \setminus j} \Phi\left(\lvert \alpha_{ij'}^{(n-1)} \rvert\right)\right)$
	$\Phi(x) = -\log(\tanh(x/2))$
MS	$\min_{j' \in \mathcal{N}(i) \setminus j} \{\lvert \alpha_{i,j'}^{(n-1)} \rvert\}$
OMS	$\max\{\min_{j' \in \mathcal{N}(i) \setminus j} \{\lvert \alpha_{i,j'}^{(n-1)} \rvert\} - \beta, 0\}$
	$\beta \geq 0$
NMS	$\lambda \cdot \min_{j' \in \mathcal{N}(i) \setminus j} \{\lvert \alpha_{i,j'}^{(n-1)} \rvert\}$
	$\lambda < 1$

of each layer is at most 1. In this way, a decoding iteration is divided into l subiterations. Formally, the algorithm for layered decoding Min-Sum is described in Algorithm 2.

After CN update is finished for one block row, the results are immediately used to update the VNs, whose results are then used to update the next layer of check nodes. Therefore, an updated information is available to CNs at each subiteration. Based on the same concept, the authors in [7] introduced the concept of turbo decoding message passing (TDMP) [16] using the BCJR algorithm [17] for their architecture-aware LDPC (AA-LDPC) codes. TDMP results in about 50% decrease in number of iterations to meet a certain BER, which is equivalent to $a \times 2$ increase in throughput and significant memory savings as compared to the standard TPMP schedule. Similar to the TDMP schedule is the vertical shuffle scheduling (VSS) [18]: while TDMP relies on horizontal divisions of the parity check matrix, VSS divides the horizontal layers into subblocks. It is a particularly efficient technique with quasicyclic LDPC codes [19], where each subblock is identified by a parity check submatrix.

Algorithm 2 (The Layered Decoding Min-Sum).

(1) *Initialization:* $\forall CN_i,\ i \in \{1, \ldots, M\}$ *do* $\beta_{i,j}^0 = 0$.

(2) *CN Update Rule:* $\forall CN_i,\ i \in \{1, \ldots, M\}$ *do*

$$\alpha_j^n = \alpha_{i,j}^0, \tag{10}$$

$$\begin{aligned} \beta_{i,j}^n = &\prod_{j' \in \mathcal{N}(i) \setminus j} \operatorname{sgn}\left\{\alpha_j^{(n-1)} - \beta_{i'j}^{(n-1)}\right\} \\ &\times \min_{j' \in \mathcal{N}(i) \setminus j} \left|\alpha_j^{(n-1)} - \beta_{i,j'}^{(n-1)}\right|, \end{aligned} \tag{11}$$

$$\alpha_j^n = \alpha_j^n + \beta_{i,j}^n. \tag{12}$$

Estimated codeword is $\hat{C} = (\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_N)$, where element $\hat{c}_j$ is calculated as

$$\hat{c}_j = \begin{cases} 0 & \text{if } \alpha_j^n > 0 \\ 1 & \text{else.} \end{cases} \tag{13}$$

If $\mathbf{H}(\hat{C})^T = 0$, then stop, with correct codeword $\hat{C}$.

3. Flexible Decoders

3.1. Parallelism. The standard TPMP algorithm described in the previous section exploits the bipartite nature of the Tanner Graph: since no direct connection is present between nodes of the same kind, all CN (or VN) operations are independent from each other and can be performed in parallel. Thus, a first broad classification of LDPC decoders can be done in terms of the degree of parallelism. The hardware implementation of LDPC decoders can be serial, partially parallel, and fully parallel.

Serial LDPC decoder implementation is the simplest in terms of area and routing. It consists of a single check node, a single variable node, and a memory. The variable nodes are updated one at a time, and then check nodes are updated in serial manner. Maximum flexibility could be achieved by uploading new check matrices in memory. However, each edge of the graph must be handled separately: as a result, throughput is usually very low, insufficient for most of standard applications.

A fully parallel architecture is the direct mapping of Tanner graph to hardware. All node operations (CNs and VNs) are directly realized in hardware PEs and connected through dedicated links. This results in huge connection complexity that in extreme cases dominates the total decoder area and results in severe layout congestion: maximum throughput can be, however, theoretically reached. In [20], a 1024-bit, fully parallel decoder is presented, achieving 1 Gbps throughput with logic density of only 50% to accommodate the complexity of interconnection: it comprises of 9750 wires with 3-bit quantization. None of the parallel implementations in [20–22] grant multimode flexibility due to wired connections. In addition, almost all existing fully parallel LDPC decoders are built on custom silicon, which precludes any prospect of reprogramming. An alternative approach is the partially parallel architecture which divides the node operations of Tanner graph over P PEs, with $P < (N + M)$. This means that each PE will perform the computation associated to multiple nodes, necessitating memories to store intermediate messages between tasks. Time sharing of PEs greatly reduces the area and routing overhead. Partially parallel architectures are studied extensively and provide a good trade off in throughput, complexity, and flexibility, with some solutions obtaining throughputs up to 1 Gbps.

3.2. Implementation Platforms. Hardware implementation of LDPC decoders is mainly dictated by the nature of application. LDPC codes have been adopted by a number of communication (wireless, wired, and broadcast) standards and storage applications: a few of them are briefly summarized in Table 2.

In wireless communication domain, LDPC codes are adopted in IEEE 802.16e WiMAX which is a wireless metropolitan area network (WMAN) standard and IEEE 802.11n WiFi which is a wireless local area network (WLAN) standard. Both standards have adopted LDPC codes as an optional channel coding scheme with various code lengths and code rates. LDPC codes are also used in digital video broadcast via satellite (DVB-S2) standard which requires

Table 2: LDPC codes applications.

Application	Standard	Code length	Code rates	Throughput
WMAN	IEEE 802.16e	576–2304	1/2–5/6	70 Mb/s
WLAN	IEEE 802.11n	648–1944	1/2–5/6	450 Mb/s
Broadcast	DVB-S2	6400,64800	1/4–9/10	90 Mb/s
Wired	10Gbase-T	2048	Arbitrary	6.4 Gbps

very large code lengths of 64800 bits and 16200 bits with 11 different codes rates, and a 90 Mb/s decoding throughput. In wireline communication domain, LDPC codes are adopted in 10 Gbit Ethernet copper (10GBASE-T) standard which specifies a high code rate LDPC code with a fixed code length of 2048 bits, with a very high decoding throughput of 6.4 Gbps.

There is no standard for magnetic recording hard disk; however, they demand high code rate, low-error floor, and high decoding throughput. In [23], a rate-8/9 LDPC decoder with 2.1 Gbps throughput has been reported for magnetic recording. The decoder utilizes four block lengths with maximum consisting of 36864 bits.

The varied nature of applications makes the selection of a suitable hardware platform an important choice. Typical platforms for LDPC decoder implementation include programmable devices (e.g., microprocessors, digital signal processors (DSPs) and application-specific instruction set processors (ASIPs)), customized application-Specific integrated circuits (ASICs), and reconfigurable devices (e.g., FPGAs).

General purpose microprocessors and DSPs utilize strong programmability to achieve highly flexible LDPC decoding, allowing to modify various code parameters at run time. Programmable devices are often used in the design, test, and performance comparison of decoding algorithms. However, they are usually constituted by a limited number of PEs that execute in a serial manner, thus limiting the computational power to a great extent. An LDPC decoder implemented on TMS320C64xx could yield 5.4 Mb/s throughput running at 600 MHz [24]. This performance is not sufficient to support high data rates defined in new wireless standards.

Reconfigurable hardware platforms like FPGAs are widely used due to several reasons. First, they speed up the empirical testing phases of decoding algorithms which are not possible in software. Secondly, they allow rapid prototyping of decoder. Once verified, the algorithm can be employed on the same reconfigurable hardware. It also allows easy handling of different code rates and SNRs, power requirements, block lengths, and other variable parameters. However, FPGAs are suited for datapath intensive designs and have programmable switch matrix (PSM) optimized for local routing. High parallelism and the intrinsic low adjacency of parity check matrix lead to longer and complex routing, not fully supported by most FPGA devices. Some designs [16, 25], used time sharing of hardware and memories that reduces the global interconnect routing, at a cost of reduced throughput.

Customized ASICs are a typical choice which yield a dedicated, high-performance IC. ASICs can be used to fulfill high computational requirements of LDPC decoding, delivering very high throughputs with reasonable parallelism. The resulting IC usually meets area, power, and speed metrics. However, ASIC designs are limited in their flexibility and usually intended for single standard applications only: flexibility, if reached at all, comes at the cost of very long design time and nonnegligible area, power or speed sacrifices. An alternative or parallel approach is the usage of ASIPs, that greatly overcome the limitations of general purpose microprocessors and DSPs. Fully customized instruction set, pipeline and memory achieve efficient, high-performance decoding: ASIP solutions are able to provide inter- and intra-standard flexibility through limited programmability, guaranteeing average to high throughput.

3.3. Decoding Schedule. A partial parallel architecture becomes mandatory to realize flexible LDPC decoding. Generally, functional description of a generic LDPC decoder can be broken down into two parts:

(i) node processors;

(ii) interconnection structure.

A partially parallel decoder with parallelism P consists of P node processors, while an interconnection structure allows various kinds of message passing according to the implemented architecture. Based on the decoding schedule that is, TPMP or Layered decoding, the datapath can be optimized accordingly. Figure 1 shows two possible datapath implementations of partially parallel LDPC decoder which are discussed as follows.

3.3.1. TPMP Datapath. In the TPMP structure depicted in [26] for a generic belief propagation algorithm, each VN consists of 4 dual port RAMs: I, Sa, Sb, and E, as shown in Figure 1(a). RAM I stores the channel intrinsic information, while RAMs Sa and Sb manage the sum of extrinsic information for previous and current iteration, respectively, and RAM E stores the extrinsic information for current iteration. The decoding process consists of D iterations: during iteration $d + 1$, the intrinsic information $(\alpha_{i,j}^{0})$ fetched from RAM I is added to the contents of RAM Sa $(\sum_{i' \in \mathcal{M}(j)} \beta_{i',j}^{d})$. Simultaneously, the extrinsic information generated by the current parity check during the previous iteration, $\beta_{i,j}^{d}$, is retrieved from RAM E and subtracted from the total. The result of the subtraction is fed to the PE, which executes the chosen CN update (see Table 1). The $d + 1$ updated extrinsics are then accumulated with the iteration d ones (RAM Sb) and replace the old extrinsic information in RAM E. At iteration $d + 2$, the roles of RAM Sa and RAM Sb are exchanged.

3.3.2. Layered Datapath. The layered decoding datapath described in [27] is shown in Figure 1(b). The VN structure is simplified and consists of RAM I only, which stores α_{j}^{d} at each subiteration. Equation(9) is computed inside the check node, that consists of a PE, a FIFO, and RAM S which stores $\beta_{i,j}^{d-1}$. During iteration d, these are subtracted from

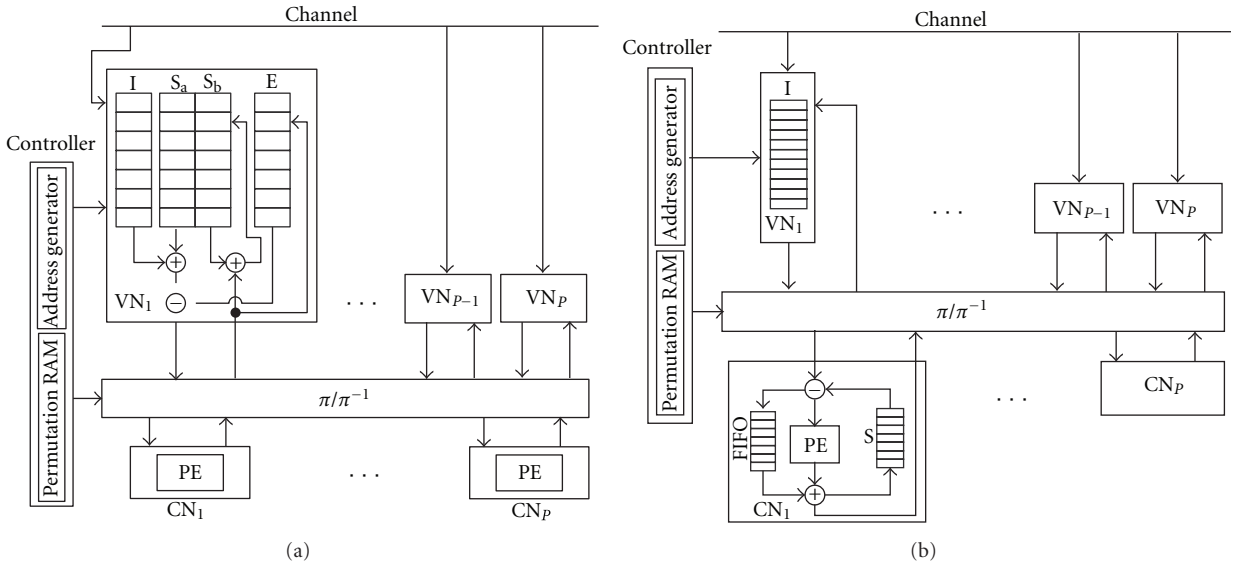


Figure 1: Generalized datapath of LDPC Decoder. (a) TPMP Decoding, (b) Layered Decoding.

the message incoming from RAM I, to generate the VN-CN message. The updated extrinsic generated by PE is added to the corresponding input coming from the FIFO, storing the resulting $\beta_{i,j}^{d}$ in RAM S.

In both datapath architectures described above, assignment of PEs to nodes (VNs and CNs) is determined by a given code structure and can be done efficiently by designing LDPC codes using permuted identity matrices. Considering parallelism P $=$ z, the z VNs are connected to z CNs through a $z \times z$ interconnection network π/π^{-1} which has to realize the permutations of identity matrix. Typically, a highly flexible barrel shifter allows all possible permutations (rotations) of identity matrix. In some implementations, a single node type joining both VN and CN operations is present, thus changing the nature and function of the connections. The controller is typically composed of address generators (AGs) and permutation RAM (PRAM). The address generators generate the address of RAMS (I, Sa, Sb and E), while the permutation RAM generates the control signals for permutation network according to the rotation of identity matrix. Multimode flexibility is achieved by reconfiguring AGs and PRAMs each time a new code needs to be supported.

In order to realize an efficient LDPC decoder, optimization is required both at PE and interconnection level. Overall complexity and performance of decoder are largely determined by the characteristics of these two functional units. In the next two sections we will discuss them in detail and analyze various design choices aimed at realizing high-performance flexible LDPC decoder.

4. Processing Element

The PE is the core of the decoding process, where the algorithm operations are performed. Its design is an important step that heavily affects overall performance, complexity and flexibility of decoder. The PE can be designed to be serial, with internal pipelining to maximize throughput, or parallel, processing all data concurrently. Depending on this initial choice, critical design issues can arise in either latency and memory requirements or complex interconnection structures and extended logic area.

4.1. Serial PE. As described in Section 2.1, the LDPC codes specified by majority of standards are based on the so-called structured LDPC codes. Considering a decoder parallelism P $=$ z, as in state-of-the-art layered decoders, one sub matrix (equal to P edges) is processed per clock cycle, with one operation completed by each PE working in a serial fashion. Figure 2(a) shows a generalized architecture for serial PE implementing the Min-Sum algorithm. In Min-Sum decoding, out of all LLRs considered by a CN, only two magnitudes are of interest, that is, minimum and the second minimum. The PE works serially maintaining three variables, namely, MIN, MIN2, and INDEX. MIN and MIN2 store the minimum and second minimum of all values, respectively, whereas INDEX stores the position index of minimum value. Each time a new VN-CN message α_{ij} is received, its magnitude is compared with MIN and MIN2, possibly substituting one of the two, with consequent position storage in INDEX. For each outgoing message β_{ij}, either the value is MIN ($i \neq$ INDEX) or MIN2 ($i =$ INDEX). Such method avoids storing all VN-CN messages and results in considerable memory saving in CN kernel.

Table 3 collects some information about WiMAX and WiFi standards different parameters. Mb denotes the number of block rows in $\mathbf{H}_{\text{BASE}}$ matrix whereas W_r and W_c denote the maximum row and column weights (i.e., CN and VN degrees), respectively. A full-mode LDPC decoder for WiMAX must support 6 code rates with weights ranging from 7 to 20. Serial CN implementation is particularly suitable for this scenario, as it allows run time flexibility to process any value of CN degree with the same number of comparators, allowing efficient hardware usage. However,

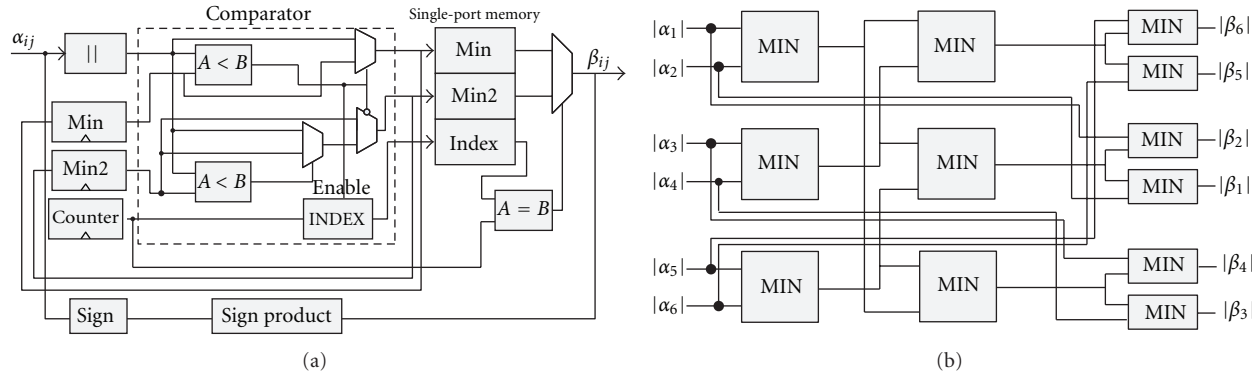


Figure 2: Min-Sum PE block scheme. (a) Serial Approach, (b) Parallel Approach.

Table 3: $\mathbf{H}_{\text{BASE}}$ parameters for WiMaX and WiFi LDPC codes.

Code rate		1/2	2/3	3/4	5/6
$\mathbf{H}_{\text{BASE}}$ matrix (WiMax/WiFi)		12×24	8×24	6×24	4×24
M_b (WiMax/WiFi)		12	8	6	4
$W_r - W_c$	WiMax	$7-6$	$11-6$	$15-6$	$20-4$
	WiFi	$8-12$	$11-8$	$15-6$	$22-4$

very large values of CN degree increase the latency and limit the achievable throughput to a great extent, requiring a high degree of parallelism to achieve medium-to-high throughputs (Table 4).

4.2. Parallel PE. Realizing high throughput decoders (supporting data rates up to few hundred Mb/s) either asks for massive parallelism or high clock frequency, resulting in significant area and power overhead. However, parallelism at CN level can bring significant increase in throughput with affordable complexity. A parallel PE manages all VN-CN messages in parallel and writes back the results simultaneously to all connected VNs. This results in lower update latency and consequently higher throughput. A parallel Min-Sum PE for $d_c = 6$ is shown in Figure 2(b). This unit computes the minimum among different choices of five out of six inputs. PE outputs the result to output ports corresponding to each input which is not included in the set, for example, $\beta_1 = \min(a_2, a_3, \ldots, a_6)$. The PE is capable of supporting all values of d_c less than or equal to 6, whereby unused inputs are initialized to $+\infty$. Supporting higher values of d_c requires additional circuitry which adds to complexity and latency of PE. As shown in the figure, the complexity of PE is dominated by logic components (e.g., comparators) and increases almost linearly with node degree. Such type of PE architectures is mostly employed to structured LDPC codes, where the check node degrees are either fixed or show small variations throughout the decoding process. To achieve code rate flexibility, the check node PE is synthesized for maximum check node degree (d_{cmax}) required by a particular application, supporting all values of d_c less than or equal to d_{cmax}.

4.3. State-of-the-Art. Flexibility as a design parameter is not always addressed as an important figure of merit, but various design techniques have been reported in the literature which can be compared in terms of throughput, complexity, and number of supported decoding modes, thus evaluating the obtainable degree of flexibility.

4.3.1. ASIC Implementations. The partially parallel decoder presented by Kuo and Willson in [32] offers a simple and tailored solution to the mobile WiMAX problem. The designed ASIC is able to work, upon reconfiguration, on all the mobile WiMAX standard LDPC codes. The quasicyclic structure of such codes allows an effective implementation of the layered decoding approach, here exploited with a variable degree of parallelism, and simple interconnection between memories and processing units. The implemented decoding algorithm is the OMS, and it is fixed. Each component of [32] is not flexible per se, but a serial architecture and programmable parallelism extend its range of usable codes to any code with parameters smaller than or equal to the WiMAX ones (block length, column and row weights, total number of exchanged information), although without guaranteeing compliance with standard throughput requirements.

In [33], the intrastandard flexibility comes together with a choice among two decoding approaches, the layered decoding and the TPMP. Although this ASIC performance has been evaluated only in case of the structured QC-LDPC codes of WiMAX, the true benefit from the dual algorithm comes in case of unstructured codes. The usually more performing layered decoding generates data collisions that are transparent to the two-phase decoding process, enabling the presence of superimposed sub-matrices in the code $\mathbf{H}$ matrix. The central processing unit consists of a Reconfigurable Serial Processing Array (RSPA) incorporating a serial Min-Sum PE and reconfigurable logarithmic barrel shifter. The RSPA can be dynamically reconfigured to choose between the two decoding modes according to different block LDPC codes. With intelligent hardware reuse via modular design the overhead due to the double decoding approach is reduced to a minimum, with an overall acceptable power consumption. The decoder operates at 260 MHz achieving

Table 4: Flexible LDPC decoder ASIC implementations. CMOS technology process (Tech), area occupation (A), A_{norm} (normalized area @ 130 nm), scheduling (sched. TDMP/TPMP), code type (C.T), block length (N), number of decoding modes supported (DM), flexibility (flex.), dt (design time), rt (run time reconfigurable), decoding iterations (It.), throughput (T.P), clock frequency (f), PE structure (PE) (serial Se, parallel Pa), number of datapaths (Dp), throughput area ratio (TAR)(Mb/s × It/mm^2 = T.P × It/A_{norm}), decoding efficiency (DE) (bits/cycle = T.P × It/f), and flexibility efficiency (FE) (DM × bits/cycle/mm^2 = DE × DM/A_{norm}).

Design	Tech. (nm)	A mm^2	A_{norm} mm^2	Sched.	C.T	N Bits	DM	Flex.	It.	T.P Mb/s	f MHz.	PE	Dp	TAR	DE	FE
		1.337	5.348		QC-WiMAX	576–2304	114		25, 20	48–333			24–96	1245.3	16.7	356.0
[28]	65	3.861	15.444	TDMP	DVB-S2	64800	20	D.T	50, 15	60–708	400	Se	90	687.6	26.6	34.4
		1.023	4.092		QC-WiFi	648–1944	12		25, 20	54–281			27–81	1373.4	14.1	41.4
[29]	65	0.51	2.04	TDMP	WiMedia	1200–1320	8	R.T	5, 3	1120–1220	264	Pa	90	1794.1	13.9	54.5
[30]	90	9.60	20.03	TDMP	DVB-S2	64800	20	R.T	15	181–998	320	Se	360	747.4	46.8	46.7
[31]	90	0.42	0.87	TDMP	QC-WiFi	1944	12	R.T	30	43	294	Pa	324	1482.7	4.4	60.7
[32]	180	3.39	1.768	TDMP	QC-WiMAX	576–2304	114	R.T	10	68	100	Se	24	384.6	6.8	438.5
[33]	130	6.3	6.3	TDMP/TPMP	Block-LDPC	576–2304	114	R.T	15	205	260	Se	24–96	487.5	11.8	213.5
[34]	130	2.46	2.46	Overlapped TDMP	QC-WiMAX	576–2304	114	R.T	15	248–287	150	Se	96	1750.0	28.7	1330.0
[27]	130	3.843	3.843	TDMP/TPMP	QC-WiMAX	576–2304	114	D.T	10,15	83–610	333	Se	24–96	2380.9	18.3	4.8
[35]	90	0.679	1.416	TDMP	QC-WiMAX	576–2304	114	R.T	8–12	200	400	Pa	16	1694.9	4.0	322.0
[36]	90	6.25	13.04	TDMP	QC-WiMAX	576–2304	114	R.T	20	105	150	Se	24–96	161.0	14.0	122.4
[37]	130	8.29	8.29	TDMP	QC-WiMAX	576–2304	19	R.T	2–8	222	83.3	Pa	4–8	214.2	5.3	12.1
[38]	130	4.94	4.94	TPMP	Arbitrary	1536	Arbitrary	R.T	2–8	86	125	Pa	Arbitrary	139.3	1.37	N/A

a handsome throughput which meets the WiMAX standard specifications.

When designing an efficient multi mode decoder a typical approach is to find similarities between different modes and then implementing common parts as reusable hardware components. Controlling the data flow between reusable components guarantees multi mode flexibility. One of such efforts is the work by Brack et al. [27]. It portraits an IP core of a full mode LDPC decoder that can be synthesized for a selected code rate specified by WiMAX standard. The unified decoder architecture proposes two different datapaths for TPMP and layered decoding, as in [33], and combines them in a single architecture sharing the components common to both. Code rate and codeword size flexibility is achieved by realizing serial check node PEs, and the chosen decoding algorithm is $\lambda - 3$ Min [43].

As discussed in Section 3.3.2, a partially parallel TDMP decoder performs serial scanning of block rows of $\mathbf{H}_{\text{BASE}}$ matrix. The check node reads the bit-LLR connected to a block row serially and stores them in FIFO whose size is proportional to row weight W_r. For multirate irregular QC decoders, the utilization ratio of FIFO is low for smaller W_r. In addition, due to random location of nonzero submatrices and correlations between consecutive block rows of $\mathbf{H}_{\text{BASE}}$, extrinsic information exchange can lead to memory access conflicts. These limitations were addressed by Xiang et al. in [34], whereby the authors presented an overlapped TDMP decoding algorithm. The design proposes a block row and column permutation criterion in order to reduce correlation between consecutive rows and uniform distribution of zero and nonzero matrices in columns, with a smart memory management technique. The resulting decoder is a full-mode, QC LDPC decoder for WiMAX. The decoder achieves a maximum throughput of 287 Mb/s, with support for other similar QC-LDPC codes.

An interesting way to tackle the flexibility issue is proposed in [35]. Here the different code parameters are handled via what has been called processing task arrangement. This work presents a decoder based on the decoding approach described in [44], layered message passing decoding with identical core matrices (LMPD-ICMs). LMPD-ICM is a variation of the original layered decoding: the $\mathbf{H}$ matrix is partitioned in several layers, with each layer yielding a core matrix. This consists of the nonzero columns of that layer. The resulting core matrix is further divided into smaller and identical tasks. Applying LMPD-ICM to a QC-LDPC code reveals that core matrices of layers are column-permuted versions of each other and show similarities not only among different layers in a single code, but also among different codes within a same code class. This technique is applied to QC-LDPC codes for WiMAX in [35], and a novel task arrangement algorithm is proposed to assign the processing operation for a variety of QC-LDPC codes to different PEs. The design features four-stage pipelining for task execution, flexible address generation to support multirate decoding, and early termination strategy which dynamically adjusts the number of iterations according to SNR values to save power. The decoder achieves a moderate throughput of 200 Mb/s at 400 MHz frequency utilizing parallelism P of 4.

A single design flow is exploited in [28] to provide three different implementations, each supporting a different standard. The adaptive single-phase decoding (ASPD) [45] scheduling is enforced, that allows to detach the decoder's memory requirements from the weight of rows and columns of the $\mathbf{H}$ matrix, leaving them dependent on the codeword length only. Though sacrificing up to 0.3 dB in BER performances, this technique accounts for 60–80% reduction in memory bits. The offset Min-Sum decoding algorithm is employed: different sizes of memories are able to comply with DVB-S2, 802.11n, and 802.16e standards. At run time, the standard serial node architecture enables intrastandard flexibility.

A classical layered scheduling is used in the DVB-S2 decoder proposed in [30]: the 360 PEs, whose architecture is detailed along with the iteration timing, are able to process a whole layer concurrently. A 360×360 barrel shifter manages the interlayer communication: since the number of rows that compose the layer never change in DVB-S2, a change of code will mean a different workload on the communication structure, but very easy reconfiguration.

The work in [36] presents a reconfigurable full-mode LDPC decoder for WiMAX. A so-called phase overlapping algorithm similar to TDMP is proposed which resolves the data dependencies of CNs and VNs of consecutive subiterations and overlaps their operation. The proposed decoder features serial check nodes with Min-Sum algorithm implementation. Parallelism of 96 yields a throughput of 105 Mb/s at 20 iterations.

In addition to serial check node architectures, the state-of-the-art for flexible LDPC decoders also reports some solutions utilizing parallel check nodes. The work in [37] proposes a reconfigurable multimode LDPC decoder for Mobile WiMAX. The authors applied the matrix reordering technique [46] to the $\mathbf{H}_{\text{BASE}}$ matrix of rate 1/2 WiMAX. This improved matrix reordering technique allowing overlapped operation of CNs and VNs and results in 68.75% reduction in decoding latency compared to nonoverlapped approach. A reconfigurable address generation unit and improved early stopping criterion help to realize a low-power flexible decoder which supports all the 19 block lengths (576–2304) of WiMAX. Parallel check nodes implementing Min-Sum algorithm help to achieve a throughput of 222 Mb/s with low frequency of 83.3 MHz and parallelism of 4.

The work in [38] features a parallel check node based on divided group comparison technique, adaptive code length assignment to improve decoding performance, and early termination scheme. The proposed solution is run time programmable to support arbitrary QC-LDPC codes of variable codes lengths and code rates. However, no compliance with standardized codes is guaranteed. A reasonable throughput of 86 Mb/s at 125 MHz is achieved. In [47], the authors presented a parallel check node incorporating the value reuse property of Min-Sum algorithm, for nonstandardized, rate 0.5 regular codes.

The WiMedia standard [48] requires very high throughput: the work by Alles et al.[29] manages to deliver more than 1 Gb/s throughputs for most code lengths and rates. The result is achieved via the instantiation of 3 PEs with

internal parallelism of 30: the similarity of WiMedia codes with the QC-LDPC of WiMAX allows a multiple submatrix-level parallelism in the decoder.

A more technological point of view is given in [31], where low-power VLSI techniques are used in a 802.11n LDPC decoder design. The decoder exploits the TDMP VSS technique with a 12-datapath architecture: separate variable-to-check and check-to-variable memory banks are instantiated, one per type for each datapath. Each of these "macrobanks" contains 3 "microbanks," each storing 9 values per word. The internal degree of parallelism is effectively sprung up to 12×27. VLSI implementation is efficiently tackled in less-than-worst case thanks to VOS [49] and RPR [50] techniques, saving area and power consumption.

4.3.2. ASIP Implementations. Future mobile and wireless communication standards will require support for seamless service and heterogeneous interoperability: convolutional, turbo, and LDPC codes are established channel coding schemes for almost all upcoming wireless standards. To provide the aforementioned flexibility, ASIPs are potential candidates. The state-of-the-art reports a number of design efforts in this domain, thanks to good performance and acceptable degree of flexibility.

The work portrayed in [39] outlines a multicore architecture based on an ASIP concept. Each core is characterized by two optimized instruction sets, one for LDPC codes and one for turbo codes. The complete decoder only requires 8 cores, since each of them can handle three processing tasks at once. The simple communication network maintains this parallelism, allowing for efficient memory sharing and collision avoidance. The intrinsic flexibility of the ASIP approach allows multiple standards (WiFi, WiMAX, LTE, and DVB-RCS) to be easily supported: the exploitation of the diverse datapath allows a very high best case throughput, while the reduced network parallelism keeps the complexity low.

A possible dual turbo/LDPC decoder architecture is described in [40]. Here, a novel approach to processing element design is proposed, allowing a high percentage of shared logic between turbo and LDPC decoding. The TDMP approach allows the usage of BCJR decoding algorithm for both codes, while the communication network can be split into smaller supercode-bound interleavers, effectively merging LDPC and turbo tasks.

In [41], the authors proposed a flexible channel coding processor FlexiCHAP. The proposed ASIP extends the capabilities of previous work FlexiTrep [51] and is capable to decode convolutional codes, binary/duobinary turbo codes and structured LDPC codes. The proposed ASIP is based on the single-instruction multiple datapath (SIMD) paradigm [52], that is, a single IP with an internal data parallelism greater than one, and processes whole submatrix of parity check matrix with single instruction. Multistandard multi mode functionality is achieved by utilizing 12-stage pipeline and elaborated memory partitioning technique. The proposed ASIP is able to decode binary turbo codes up to 6144 information bits, duobinary turbo codes and convolutional codes upto 8192 information bits. LDPC decoding capability of block length up to 3456 bits and check node degree up to 28 is sufficient to cover all code rates of WiMAX and WiFi standards. The ASIP achieves payloads of 237 Mb/s and 257 Mb/s for WiMAX and WiFi, respectively.

The solution proposed in [42] makes use of a single SIMD ASIP with maximum internal parallelism of 96. A combined architecture is strictly designed for LDPC codes, but the supported ones range from binary (WiFi and WiMAX) to nonbinary (Galois Field of order 9): the decoding approach is turbo decoder-based. While the decoding mode can be changed at runtime, flexibility is guaranteed at design time, by instancing wide enough rotation engines for the different LDPC submatrix sizes, and a sufficient number of memories. These memories dominate the area occupation, mainly due to the non-binary decoding process.

Tables 4 and 5 summarize the specifications of various state-of-the-art ASIC and ASIP solutions for flexible LDPC decoders discussed above. To simplify the comparison, the area of each decoder has been scaled up to 130 nm process represented as normalized area (A_{norm}). A parameter called throughput to area ratio (TAR) defined as TAR = Throughput $\times$ It/Area has also been included in the table to evaluate the area efficiency of proposed decoders. Another metric named decoding efficiency (DE) given as DE = (Throughput $\times$ It.)$/f$ [53] has been defined to give a good comparison regardless of different clock frequencies. DE gives the number of decoded bits per clock cycle per iteration, while Dp is the total degree of parallelism, taking into account both the number of PEs and their possible multiple datapaths.

To evaluate the effective flexibility of each decoder, and its cost, a metric called *flexibility efficiency* is introduced, and computed as

$$\mathrm{FE} = \frac{\mathrm{DE} \times \mathrm{DM}}{A_{norm}}. \tag{14}$$

It gives a measure of each decoder's flexibility through its different decoding modes (DMs), taking in account the normalized throughput performances (DE) in relation to the normalized area occupation A_{norm}. The metric is applied to ASIC decoders only, since in these cases the cost of flexibility is reflected on the area much more directly than in the ASIP case.

As shown in Table 4, the work in [27] dominates in terms of throughput and TAR but offers only design time flexibility (effective D.M = 1): for this reason, its FE value is very low. On the contrary, the design time flexibility of [28] results in a runtime flexibility once the standard to be supported has been chosen: since the implementation of the WiMAX decoder requires a higher number of codes to be supported at the same time than DVB-S2 and WiFi, its FE will be higher. Best DE value is held by the DVB-S2 decoder presented in [30], together with an average TAR. Explicitly enabling the decoding of just 20 codes, however, lowers its FE measure. Among serial PE-based run time flexible solutions discussed above, the work in [34] achieves very high throughput, TAR and DE with a small area occupation of 2.46 mm^2, yielding the best FE of all decoders. A full-mode reconfigurable solution based on parallel check node in [35]

Table 5: Flexible LDPC decoders ASIP Implementations. CMOS technology process (Tech), area occupation (A), normalized area (A_{norm})@ 130 nm, code type (C.T), flexibility (Flex.) design time (D.T), run time (R.T), maximum throughput (T.P), maximum iterations (It.), number of datapaths (Dp), operating frequency (f), processing element (PE) (serial Se, parallel Pa), throughput area ratio (TAR)(Mb/s $\times$ It/mm^2 = t.p $\times$ It/A_{norm}), and decoding efficiency (DE) (bits/cycle = t.p $\times$ It/f).

Design	Tech. (nm)	A mm^2	A_{norm} mm^2	C.T	Flex.	It.	T.P Mb/s	f MHz.	Dp	PE	TAR	DE
Multicore ASIP [39]	90	2.6	5.42	LDPC-{WiMAX,WiFi} Turbo-{BTC-LTE,DBTC-WiMAX}	R.T	10 6	{312, 263} {173, 173}	500	24	Se	{575.6, 485.2} {191.4, 191.4}	{6.24, 5.26} {2.07, 2.07}
2D NOC ASIP [40]	130	N/A	N/A	Turbo LDPC	R.T	8	86.5 11.2	200	16	Se	N/A	3.46 0.448
FlexiCHAP [41]	65	0.62	2.48	LDPC-{WiMAX, WiFi} Turbo {BTC,DBTC}	R.T	10–20 5	{237, 257} {18.6, 37.2}	400 (max.)	27	Se	{955.6, 1036.3} {37.5,75.0}	{5.9, 6.42} {0.23, 0.46}
Bin/non-Bin [42]	65	3.4	13.6	Bin LDPC-{WiMAX, WiFi} Non-Bin LDPC {GF(9)}	R.T	10 1	90 12.5	400	96	Se	66.2 0.92	2.25 0.03

achieves a handsome throughput of 200 Mb/s and the best FE among the parallel node solutions. Its A_{norm} of 1.416 mm^2 is the minimum among all WiMAX solutions discussed above, but with very low DE stains overall performance.

Among the ASIP solutions (Table 5), the work in [40] cannot effectively be compared to the others in terms of area, not providing complete estimations. The work in [41] yields the higher TAR and DE in LDPC mode: the solution proposed in [39], however, yields the best decoding efficiency and the top TAR in turbo mode, while at the same time reaching a very good DE in LDPC mode too.

5. Interconnection Structures

As shown through the previous sections, in the great majority of current LDPC decoders, some kind of intradecoder communication is necessary. Except for very few single-core implementations based on the single-instruction single datapath (SISD) paradigm, the need for message routing or permutation is a constant throughout the wireless communication state of the art. As a first classification, two scenarios can be roughly devised:

(i) *single PE architectures*: some state-of-the-art decoders propose single core solutions with internal parallelism greater than one, that rely on smart memory sharing and on programmable permutation networks. These decoders make often use of TDMP and VSS, that require either reduced communication or very regular patterns: the involved interconnection structures are simple;

(ii) *partially parallel architectures*: referring to the graph representation of the LDPC **H** matrix within a selected decoding approach, it is possible to map the graph nodes onto a certain number of processing cores. In the partially parallel approach, the number of graph nodes is much higher than the PEs. Each node is connected to a set of other nodes distributed on the available PEs: different nodes will have different links, resulting in a widely varied PE-to-PE communication pattern. This situation calls for flexible and complex interconnection structures.

5.1. Shift and Shuffle Networks. Structured LDPC codes decoding, regardless of their implementation, often require shift or shuffle operation to route information between PEs or to/from memories. This is particularly true for some kinds of LDPC codes, as QC-LDPC and *shift*-LDPC [54].

The barrel shifter (BS) is a well-known circuit designed to perform all the permutations of its inputs obtainable with a shift operation, thus being well suited for the circularly shifted structure of QC-LDPC **H** matrix.

Rovini et al. in [55] exploit the simple structure of the barrel shifter to design a circular shifting network for WiMAX codes. This network must be able to handle all the different submatrix sizes of the standard, thus effectively becoming a multisized circular shifting (MS-CS) network. This MS-CS network is composed of a number of $B \times B$ BSs, where B is the greatest common divisor among all the supported block sizes. Each BS rotates of the same shift amount all the blocks of B data, that are subsequently rearranged by an adaptation network into the desired order according to the current submatrix size. Implementation results show that the proposed MC-CS network outperforms in terms of complexity previous similar solutions as [56–59], with a saving ranging from 30.4% to 67.2%.

In [36] is designed a shift network for WiMAX standard, based on a self-routing technique. The network is sized to handle the largest submatrix size of the standard, 96: when decoding a smaller code, dummy messages are routed as well, with a dedicated flag. Two stages of barrel shifters provide the shift function to real and dummy messages alike, together with a single permutation network: a lookup engine finally selects the useful ones basing its decision on the flag bits, shift size and submatrix size.

Barrel shifters, though providing the most immediate implementation of the shift operation, often lack the necessary flexibility to directly tackle multiple block sizes. For this reason, they are usually joint to more complex structures.

One of the most common implementations among the simplest interconnection structures is the Benes network (BeN). This kind of network is a rearrangeable nonblocking network frequently used as a permutation network. Defining S_M the number of inputs and outputs, an $S_M \times S_M$ BeN can perform any permutation of the inputs creating a one-to-one relation with the outputs with $(S_M/2 \times (2\log_2 S_M - 1))2 \times 2$ switches. Its stand-alone use is thus confined to situations in which sets of data tend not to intertwine: its range of usage, though, can be extended through smart scheduling.

In [60] a flexible ASIC architecture for high-throughput *shift*-LDPC decoders is depicted. *Shift*-LDPC codes are subclass of structured LDPC: the **H** of an $(N, M)(M_b, N_b)$ *Shift*-LDPC is structured as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{1,1} & \mathbf{P_m}\mathbf{H}_{1,1} & \cdots & \mathbf{P_m}^{N_b-1}\mathbf{H}_{1,1} \\ \mathbf{H}_{2,1} & \mathbf{P_m}\mathbf{H}_{2,1} & \cdots & \mathbf{P_m}^{N_b-1}\mathbf{H}_{2,1} \\ \cdots & \cdots & \ddots & \cdots \\ \mathbf{H}_{M_b,1} & \mathbf{P_m}\mathbf{H}_{M_b,1} & \cdots & \mathbf{P_m}^{N_b-1}\mathbf{H}_{M_b,1} \end{bmatrix}, \quad (15)$$

where $N \times M$ are the dimensions of the H matrix, and the leftmost M_b submatrices are randomly row-permuted versions of the $z \times z$ identity matrix **I**. Matrix $\mathbf{P_m}$ identifies a $z \times z$ permutation matrix, obtained by cyclically shifting right the columns of **I** by a single position. The operations involved in the definition of each the $M_b \times N_b$ submatrices guarantee that $\mathbf{P_m}^k\mathbf{H}_{i,1}$ is $\mathbf{P_m}^{k-1}\mathbf{H}_{i,1}$ with the rows shifted up one position, so all matrices of row i can be found cyclically shifting $\mathbf{H}_{i,1}$.

To exploit at best the proprieties of *Shift*-LDPC, a VSS scheme has been selected, along with a highly parallel implementation: z variable node units (VNUs) and M check node units (CNUs) perform a whole iteration in N_b steps. Since every **H** submatrix is a shifted version of the previous one, also the connections between z VNUs and z CNUs shift cyclically every clock cycle: this observation leads to the joint design of the Benes global permutation network and the CN shuffler.

The BeN is used to define the links between VNUs and CNUs: these links are static, once the parameters z, N_b, and the structure of the M_b leftmost $\mathbf{H}_{x,1}$ have been fixed. Its high degree of flexibility is exploited to guarantee support over a variety of different codes. The inter-CN communication required by the VSS approach is handled by the CN shuffle network. Its function is to cyclically shift the submatrix rows assigned to each CNU: this means that while each CNU will be physically connected to the same VNU for the whole decoding, the row of the **H** matrix they represent will change. The BeN has consequently no need to be rearranged.

The flexible decoder is able to achieve 3.6 Gb/s with an area of 13.9 mm^2 in 180 nm CMOS technology: the area occupation is relatively small w.r.t. the very high degree of parallelism thanks to the nonuniform 4-bit quantization scheme adopted.

In [61], a SIMD-based ASIC is proposed for LDPC decoding over a wide array of standards. The ASIC is composed of 12 parallel datapaths able to decode both turbo and LDPC codes through the BCJR algorithm. As most of the SIMD cores, the decoder handles communication by means of shared memories. Memory management can be challenging, especially in case the parallel datapaths are assigned to fractions of the same codeword. According to [62], it is possible to avoid collisions in such cases with adhoc mapping of the interleaving laws together with one (for LDPC) or two (for turbo) permutation networks to interface with memories. These two networks are implemented in [61] with 8×8 BeN, the one at the input of the extrinsic values memory being transparent in LDPC mode.

Not every supported standard require all the 12 datapaths to be active: the chosen parallelism is the minimum necessary for throughput compliance, and the same can be said for the working frequency. The implementation results show full compliance with WiMAX, WiFi, 3GPP-HSDPA, and DVB-SH, at the cost of 0.9 mm ^{2} in 45 nm CMOS, technology, and total power consumption of 86.1 mW.

One of the limitations of the traditional BeN is the number of its inputs and outputs, that are bound to be a power of 2. However, LDPC decoders often need a permutation of different size: for example, WiMAX codes require shift permutations of sizes corresponding to the possible expansion factors, that is, from 24 to 96 with steps of 4. In [63], an alternative switch network is designed that makes use of 3×3 switches as well, leading to a more hardware-efficient design. The introduction of 3×3 switches allows in fact $S_M = 3 \times 2^i$. A fully compliant WiMAX LDPC decoder shift operation can thus be implemented with a novel 96×96 switch network: it requires $3 \times 2^i + 3 \times 2^i \log_2 2^i = 5762 \times 2$ switches, against the 832 necessary for a traditional 128×128 BeN. Together with efficient control signal generation, this solution outperforms in terms of both complexity and flexibility other modified Benes-based decoders as [57, 64], that exploit a secondary BeN to rearrange the first one.

In [65], Lin et al. propose an optimized Benes-based shuffle network for WiMAX. Unlike [63], the starting point of the design is the nonoptimized 128×128 BeN: from here all the switches are removed where no signal is passed, whereas switches with fixed output are replaced by wires. This adhoc trimming technique, together with an efficient algorithm for control signal creation, allow a 26.6%–71.1% area reduction with respect to previously published shift network solutions like [27, 55, 66].

Similar to the BeN is the Banyan network (ByN) [67], that can be seen as a trade-off between the flexibility of the BeN and the complexity of the BS. Although not non-blocking in general, the ByN is non-blocking in case of shift operations only. Moreover, it is composed of averagely half of the 2×2 switches of a BeN and requires fewer control signals.

The work described in [68] portrays a highly parallel shuffle network based on the ByN paradigm. Like the BeN, also ByN is bound to a power-of-two number of inputs: as Oh and Parhi have done in [63], also here the introduction of 3×3 switches allows to handle WiMAX standard various submatrix sizes. The implemented decoder guarantees a very high degree of flexibility with complexity lower or comparable to [36, 63–65].

5.2. Networks-on-Chip. Networks-on-Chip (NoCs) [69] are versatile interconnection structures that allow communication among all the connected devices through the presence of routing elements. Recently, LDPC decoders for both turbo and LDPC codes based on the NoC paradigm have been proposed, thanks to the intrinsic flexibility of NoCs. NoC-based decoders are multiprocessor systems composed of various instantiations of the same IP associated to a routing element, which are linked in defined pattern. This pattern can be represented with a graph, in which every node corresponds to a processor and a router: the arcs are the physical links among routers, thus identifying a topology.

In [70], a De Bruijn topology [71] NoC is proposed for flexible LDPC decoders. Since the TPMP has been selected, the NoC must handle the communication between VNs and CNs. The NoC design, however, is completely detached from code and decoding parameters, effectively allowing usage for any LDPC code. The router embeds a modified shortest path routing algorithm that can be executed in 1 clock cycle, together with deadlock-free and buffer-reducing arbitration policies and is connected to its PE via the network interface. The network is synthesized and compared to other explored network topologies, as the 2-dimensional mesh [72], Benes [73] and MDN [74]: the degree of flexibility and scalability that the proposed topology guarantees is unmatched.

The performance of another topology, the 2D toroidal mesh, is evaluated in [40]. The routing element implements the near-optimal X-Y routing for the torus/mesh [75]. A whole set of communication-centric parameters is varied in order to evaluate the impact of the network latency on the whole decoder performance. It has been shown that small PE sending periods R, that is, cycles between two available data from the processor, increase latency to unsustainable levels, with the smallest values at $R \approx 7$. Also, the variations in throughput due to different NoC parallelisms are shaded by the impact of latency.

The work in [76] describes a flexible LDPC decoder design tackling the communication problem with two different NoC solutions. The first network is a De Bruijn NoC

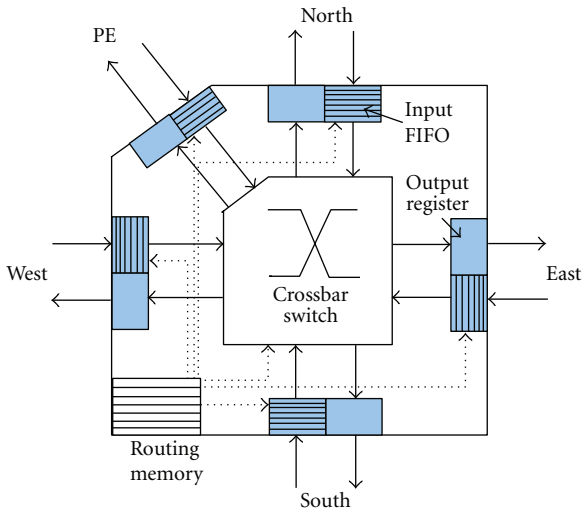


Figure 3: ZONoC routing element.

adopting online dynamic routing, implementing the same modified shortest path algorithm described in [70], while the second, a 2D torus, is based on a completely novel concept named zero overhead NoC (ZONoC). Given a mapping of VNs and CNs over a topology, the message exchange pattern is deterministic, along with the status of the network at each instant. The ZONoC exploits this propriety by running offline simulations and storing routing information into dedicated memories, effectively wiping out the time spent for routing and traffic control. Overall network complexity is scaled down, since no routing algorithm is necessary; FIFO length can be trimmed to the minimum necessary, while router architecture is as simple as possible (Figure 3). A cross-bar switch controlled by the routing memory receives messages from the FIFOs connected to its PE and other routers, while outgoing messages are sent to as many output registers. Implementation results show a significant reduction in complexity with respect to [28, 36, 56, 72] and comparable or superior throughput.

5.3. Reducing the NoC Penalty. The NoC approach guarantees a very high degree of flexibility and, in theory, a NoC-based decoder can reach very high throughput. The achievable throughput is proportional to the number of PEs: but increasing the size of the network means rising the latency, and thus degrading performance back. Very few state of the art solutions have managed to solve this problem, and those who do suffer from large complexity and power consumption. We have tried to overcome these shortcomings in some recent works.

5.3.1. NoC-Based WiMAX LDPC Decoder. The solution described in [76] supports the WiMAX standard LDPC codes, but does not guarantee a high enough throughput. Stemming from it, we have developed an LDPC ZONoC-based decoder fully compliant with WiMAX standard: although having a more convoluted graph structure, relies on a smaller number of exchanged messages and guarantees $a \times 2$ factor in convergence speed. We designed a sequential PE implementing the normalized Min-Sum decoding algorithm, as described by Hocevar in [77]: unlike in [76], we adopt the layered decoding approach. The PE architecture is independent of code parameters, and the memory capacity sets the only limit to the size of supported codes. Together with the PE, we devised a decoder reconfiguration technique to upload the data necessary for routing and memory management when switching between codes.

In order to comply with WiMAX standard throughput requirements, the size of the 2D torus mesh has been risen from 16 nodes to 25. As detailed in [78], the decoder guarantees more than 70 Mb/s for all rates and block sizes of the standard, with an area of 4.72 mm^2 in 130 nm CMOS technology.

5.3.2. Bandwidth and Power Reduction Methods. While the former decoder is compliant with WiMAX in worst case, that is, when the maximum allowed number of iteration is performed, a codeword is averagely corrected with fewer iterations: the unnecessary iterations significantly contribute to the NoC high-power consumption. In [79], two methods aimed at reducing power and increasing throughput are studied and implemented. The first one is the iteration early-stopping (ES) criterion proposed in [80], that allows to stop the decoding when all the information bits of a codeword are correct, regardless of the redundancy bits. The other is a threshold-based message stopping (MS) criterion, that reduces the traffic load on the network by avoiding injection of values which carry information about high-probability correct bits.

Various combinations of the two methods have been tried, together with different parallelisms of the 2D torus mesh. Implementation of the ES criterion requires a dedicated processing block with minimal PE modifications, while MS requires a threshold comparison block for each PE and switching to online dynamic routing. This is necessary since stopping a message invalidates the statically computed communication pattern. While the ES method guarantees an average 10% energy per frame decoding reduction regardless of the implementation, the MS method's results change with the size of the NoC. Since stopped messages can lead to additional errors, a performance sacrifice must be accepted: among the solutions presented in [79], with 0.3 dB BER loss, a 9-PE NoC is sufficient to support the whole WiMAX standard.

5.3.3. NoC Analysis for Turbo/LDPC Decoders. In [81], an extensive analysis of performance of various NoC topologies is performed in the context of multiprocessor turbo decoders. Flexibility can be explored also in terms of types of code supported: in [82], we have consequently extended the topology analysis to LDPC codes, in order to find a suitable architecture for a dual turbo/LDPC codes. As a case of study, we focused our research on the WiMAX codes.

The performance of a wide set of topologies (ring, spidergon, toroidal meshes, honeycomb, De Bruijn, and Kautz) has been evaluated in terms of achievable throughput and complexity, considering different parallelisms. Exploiting

a modified version of the cycle-accurate simulation tool described in [81], a range of design parameters has been taken in consideration, including data injection rate, message collision management policies, routing algorithms, node addressing modes, and structure of the routing element, allowing to span from a completely adaptive architecture to a ZONoC-like precalculated routing.

The simulations revealed the Kautz topology [83] to be the best trade-off in terms of throughput and complexity between LDPC and turbo codes, with a partially adaptive router architecture and FIFO-length based routing. Two separate PEs for turbo and LDPC codes have been designed: different NoC and PE working frequencies allow to trim the message injection rate in the network. The full decoder, complete with turbo and LDPC separated PEs, has been synthesized with 90 nm CMOS technology: the decoder is compliant with both turbo and LDPC throughput requirements for all the WiMAX standard codes. Worst-case throughput results overperform the latest similar solutions as [39, 41, 61, 84], with a small area occupation and particularly low-power consumption (59 mW) in turbo mode.

6. Conclusions

A complete overview of LDPC decoders, with particular emphasis on flexibility, is drawn. Various classifications are depicted, according to degree of parallelism and implementation choices, focusing on common design choices and elements for flexible LDPC decoders. An indepth view is given over the PE and interconnection part of the decoders, with comparison with the current state-of-the-art, the latest work by the authors on NoC-based decoders is briefly described.

References

[1] A. Morello and V. Mignone, "DVB-S2: the second generation standard for satellite broad-band services," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 210–227, 2006.

[2] J. Lörincz and D. Begušić, "Physical layer analysis of emerging IEEE 802.11n WLAN standard," in *Proceedings of the 8th International Conference Advanced Communication Technology (ICACT '06)*, pp. 189–194, February 2006.

[3] The IEEE p802.3an, 10GBASE-T task force, http://www.ieee802.org/3/an/.

[4] "IEEE standard for local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems amendment 2: physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1," *IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005 (Amendment and Corrigendum to IEEE Std 802.16-2004)*, 2006.

[5] R. Gallager, "Low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[6] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[7] M. M. Mansour and N. R. Shanbhag, "A 640-Mb/s 2048-bit programmable LDPC decoder chip," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 684–698, 2006.

[8] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[9] G. Masera, F. Quaglio, and F. Vacca, "Finite precision implementation of LDPC decoders," *IEE Proceedings on Communications*, vol. 152, no. 6, pp. 1098–1102, 2005.

[10] N. Wiberg, *Codes and decoding on general graphs*, Ph.D. dissertation, Linkoping University, Linkoping, Sweden, 1996.

[11] M. Daud, A. Suksmono, Hendrawan, and Sugihartono, "Comparison of decoding algorithms for LDPC codes of IEEE 802.16e standard,," in *Proceedings of the 6th International Conference on Telecommunication Systems, Services, and Applications (TSSA '11)*, pp. 280–283, October 2011.

[12] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of LDPC codes and extension to turbo decoding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '01)*, p. 189, June 2001.

[13] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X. Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, 2005.

[14] M. Martina, G. Masera, S. Papaliralabos, P. Mathiopoulos, and F. Gioulekas, "On practical implementation and generalization of max* operation for Turbo and LDPC decoders," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 4, pp. 888–895, 2012.

[15] E. Yeo, P. Pakzad, B. Nikolić, and V. Anantharam, "High throughput low-density parity-check decoder architectures," in *Proceedings of the IEEE Global Telecommunicatins Conference (GLOBECOM '01)*, pp. 3019–3024, November 2001.

[16] M. Mansour and N. Shanbhag, "Memory-efficient turbo decoder architectures for LDPC codes," in *Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS '02)*, pp. 159–164, October 2002.

[17] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[18] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," in *Proceedings of the 36th Asilomar Conference on Signals Systems and Computers*, vol. 1, pp. 8–15, November 2002.

[19] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. Costello, "LDPC block and convolutional codes based on circulant matrices," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 2966–2984, 2004.

[20] A. J. Blanksby and C. J. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, 2002.

[21] L. Fanucci, P. Ciao, and G. Colavolpe, "VLSI design of a fully-parallel high-throughput decoder for turbo gallager codes," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 89, no. 7, pp. 1976–1986, 2006.

[22] V. Nagarajan, N. Jayakumar, S. Khatri, and O. Milenkoviç, "High-throughput VLSI implementations of iterative decoders and related code construction problems," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 1, pp. 361–365, December 2004.

[23] H. Zhong, W. Xu, N. Xie, and T. Zhang, "Area-efficient min-sum decoder design for high-rate quasi-cyclic low-density parity-check codes in magnetic recording," *IEEE Transactions on Magnetics*, vol. 43, no. 12, pp. 4117–4122, 2007.

[24] G. Lechner, J. Sayir, and M. Rupp, "Efficient DSP implementation of an LDPC decoder," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. V-665–V-668, May 2004.

[25] T. Zhang and K. Parhi, "A 54 Mbps (3,6)-regular FPGA LDPC decoder," in *Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS '02)*, pp. 127–132, October 2002.

[26] E. Boutillon, J. Castura, and F. Kschichang, "Decoder first code desig," in *Proceedings of the 2nd International Symposium on Turbo Codes and Related Topics*, pp. 459–462, September 2000.

[27] T. Brack, M. Alles, F. Kienle, and N. Wehn, "A synthesizable IP core for WiMax 802.16E LDPC code decoding," in *Proceedings of the 17th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '06)*, pp. 1–5, September 2006.

[28] T. Brack, M. Alles, T. Lehnigk-Emden et al., "Low complexity LDPC code decoders for next generation standards," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE '07)*, pp. 331–336, April 2007.

[29] M. Alles, N. Wehn, and F. Berens, "A synthesizable IP core for WiMedia 1.5 UWB LDPC code decoding," in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB '09)*, pp. 597–601, September 2009.

[30] B. Zhang, H. Liu, X. Chen, D. Liu, and X. Yi, "Low complexity DVB-S2 LDPC decoder," in *Proceedings of the 69th IEEE Vehicular Technology Conference (VTC '09)*, pp. 1–5, April 2009.

[31] J. Cho, N. R. Shanbhag, and W. Sung, "Low-power implementation of a high-throughput LDPC decoder for IEEE 802.11N standard," in *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS '09)*, pp. 40–45, October 2009.

[32] T. C. Kuo and A. N. Willson, "A flexible decoder IC for WiMAX QC-LDPC codes," in *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC '08)*, pp. 527–530, September 2008.

[33] S. Huang, D. Bao, B. Xiang, Y. Chen, and X. Zeng, "A flexible LDPC decoder architecture supporting two decoding algorithms," in *Proceedings of the IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems (ISCAS '10)*, pp. 3929–3932, June 2010.

[34] B. Xiang, D. Bao, S. Huang, and X. Zeng, "A fully-overlapped multi-mode QC-LDPC decoder architecture for mobile WiMAX applications," in *Proceedings of the 21st IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '10)*, pp. 225–232, July 2010.

[35] Y. L. Wang, Y. L. Ueng, C. L. Peng, and C. J. Yang, "Processing-task arrangement for a low-complexity full-mode WiMAX LDPC codec," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 2, pp. 415–428, 2011.

[36] C. H. Liu, S. W. Yen, C. L. Chen et al., "An LDPC decoder chip based on self-routing network for IEEE 802.16e applications," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 684–694, 2008.

[37] X. Y. Shih, C. Z. Zhan, C. H. Lin, and A. Y. Wu, "An 8.29 mm2 52 mW multi-mode LDPC decoder design for mobile WiMAX system in 0.13 μm CMOS process," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 672–683, 2008.

[38] X. Y. Shih, C. Z. Zhan, and A. Y. Wu, "A real-time programmable LDPC decoder chip for arbitrary QC-LDPC parity check matrices," in *Proceedings of the IEEE Asian Solid-State Circuits Conference (A-SSCC '09)*, pp. 369–372, November 2009.

[39] P. Murugappa, R. Al-Khayat, A. Baghdadi, and M. Jezequel, "A flexible high throughput multi-ASIP architecture for LDPC and turbo decoding," in *Proceedings of the 14th Design, Automation and Test in Europe Conference and Exhibition (DATE '11)*, pp. 228–233, March 2011.

[40] M. Scarpellino, A. Singh, E. Boutillon, and G. Masera, "Reconfigurable architecture for LDPC and turbo decoding: a NoC case study," in *Proceedings of the 10th International Symposium on Spread Spectrum Techniques and Applications (ISSSTA '08)*, pp. 671–676, August 2008.

[41] M. Alles, T. Vogt, and N. Wehn, "FlexiChaP: a reconfigurable ASIP for convolutional, turbo, and LDPC code decoding," in *Proceedings of the 5th International Symposium on Turbo Codes and Related Topics (TURBOCODING '08)*, pp. 84–89, September 2008.

[42] F. Naessens, A. Bourdoux, and A. Dejonghe, "A flexible ASIP decoder for combined binary and non-binary LDPC codes," in *Proceedings of the 17th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT '2010)*, pp. 1–5, November 2010.

[43] F. Guilloud, E. Boutillon, and J. Danger, "λ-min decoding algorithm of regular and irregular LDPC codes," in *Proceedings of the 3rd International Symposium on Turbo Codes and Related Topics*, pp. 451–454, September 2003.

[44] Y. Dai, N. Chen, and Z. Yan, "Memory efficient decoder architectures for quasi-cyclic LDPC codes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 9, pp. 2898–2911, 2008.

[45] M. Castano, M. Rovini, N.E. L'Insalata et al., "Adaptive single phase decoding of LDPC codes," in *Proceedings of the 6th International ITG-Conference on Source and Channel Coding (TURBOCODING), 4th International Symposium on Turbo Codes&Related Topics*, pp. 1–6, April 2006.

[46] I. C. Park and S. H. Kang, "Scheduling algorithm for partially parallel architecture of ldpc decoder by matrix permutation," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, vol. 6, pp. 5778–5781, May 2005.

[47] K. Gunnam, G. Choi, and M. Yeary, "A parallel VLSI architecture for layered decoding for array LDPC codes," in *Proceedings of the 6th International Conference on Embedded Systems, 20th International Conference on VLSI Design*, pp. 738–743, January 2007.

[48] High Rate UWB PHY and MAC Standard, Standard ECMA-368 Std, http://www.ecma-international.org.

[49] R. Hegde and N. R. Shanbhag, "A voltage overscaled low-power digital filter IC," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 2, pp. 388–391, 2004.

[50] B. Shim, S. R. Sridhara, and N. R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 5, pp. 497–510, 2004.

[51] T. Vogt and N. Wehn, "A Reconfigurable application specific instruction set processor for convolutional and turbo decoding in a SDR environment," in *Proceedings of the Design, Automation and Test in Europe (DATE '08)*, pp. 38–43, March 2008.

[52] M. Flynn, "Very high-speed computing systems," *Proceedings of the IEEE*, vol. 54, no. 12, pp. 1901–1909, 1966.

[53] M. Awais, A. Singh, E. Boutillon, and G. Masera, "A novel architecture for scalable, high throughput, multi-standard LDPC decoder," in *Proceedings of the 14th Euromicro Conference on Digital System Design (DSD '11)*, vol. 31, pp. 340–347, September 2011.

[54] J. Sha, Z. Wang, M. Gao, and L. Li, "Multi-Gb/s LDPC code design and implementation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 2, pp. 262–268, 2009.

[55] M. Rovini, G. Gentile, and L. Fanucci, "Multi-size circular shifting networks for decoders of structured LDPC codes," *Electronics Letters*, vol. 43, no. 17, pp. 938–940, 2007.

[56] F. Quaglio, F. Vacca, C. Castellano, A. Tarable, and G. Masera, "Interconnection framework for high-throughput, flexible LDPC decoders," in *Proceedings of the Design, Automation and Test in Europe (DATE '06)*, p. 6, March 2006.

[57] K. K. Gunnam, G. S. Choi, M. B. Yeary, and M. Atiquzzaman, "VLSI architectures for layered decoding for irregular LDPC codes of WiMax," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 4542–4547, June 2007.

[58] J. Dielissen, A. Hekstra, and V. Berg, "Low cost LDPC decoder for DVB-S2," in *Proceedings of the Design, Automation and Test in Europe (DATE '06)*, vol. 2, pp. 1–6, March 2006.

[59] M. Karkooti, P. Radosavljevic, and J. R. Cavallaro, "Configurable, high throughput, irregular LDPC decoder architecture: tradeoff analysis and implementation," in *Proceedings of the 17th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '06)*, pp. 360–367, September 2006.

[60] C. Zhang, Z. Wang, J. Sha, L. Li, and J. Lin, "Flexible LDPC decoder design for multigigabit-per-second applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 116–124, 2010.

[61] G. Gentile, M. Rovini, and L. Fanucci, "A multi-standard flexible Turbo/LDPC decoder via ASIC design," in *Proceedings of the 6th International Symposium on Turbo Codes and Iterative Information Processing (ISTC '10)*, pp. 294–298, September 2010.

[62] A. Tarable, S. Benedetto, and G. Montorsi, "Mapping interleaving laws to parallel turbo and LDPC decoder architectures," *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 2002–2009, 2004.

[63] D. Oh and K. K. Parhi, "Low-complexity switch network for reconfigurable LDPC decoders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 1, pp. 85–94, 2010.

[64] J. Tang, T. Bhatt, V. Sundaramurthy, and K. K. Parhi, "Reconfigurable shuffle network design in LDPC decoders," in *Proceedings of the 17th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '06)*, pp. 81–86, September 2006.

[65] J. Lin, Z. Wang, L. Li, J. Sha, and M. Gao, "Efficient shuffle network architecture and application for WiMAX LDPC decoders," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 3, pp. 215–219, 2009.

[66] C. H. Liu, C. C. Lin, S. W. Yen et al., "Design of a multimode QC-LDPC decoder based on shift-routing network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 9, pp. 734–738, 2009.

[67] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann Publishers, 1992.

[68] X. Peng, Z. Chen, X. Zhao, F. Maehara, and S. Goto, "High parallel variation banyan network based permutation network for reconfigurable LDPC decoder," in *Proceedings of the 21st IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '10)*, pp. 233–238, July 2010.

[69] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, 2002.

[70] H. Moussa, A. Baghdadi, and M. Jézéquel, "Binary de Bruijn on-chip network for a flexible multiprocessor LDPC decoder," in *Proceedings of the 45th Design Automation Conference (DAC '08)*, pp. 429–434, June 2008.

[71] N. De Bruijn, "A combinatorial problem," *Koninklijke Nederlandse Akademie*, vol. 49, pp. 758–764, 1946.

[72] T. Theocharides, G. Link, N. Vijaykrishnan, and M. J. Irwin, "Implementing LDPC decoding on network-on-chip," in *Proceedings of the 18th International Conference on VLSI Design: Power Aware Design of VLSI Systems*, pp. 134–137, January 2005.

[73] G. Masera, F. Quaglio, and F. Vacca, "Implementation of a flexible LDPC decoder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 6, pp. 542–546, 2007.

[74] F. Kienle, M. Thul, and N. When, "Implementation issues of scalable LDPC-decoders," in *Proceedings of the 3rd International Symposium on Turbo Codes and Related Topics*, 2003.

[75] D. Seo, A. Ali, W. T. Lim, N. Rafique, and M. Thottethodi, "Near-optimal worst-case throughput routing for two-dimensional mesh networks," in *Proceedings of the 32nd Interntional Symposium on Computer Architecture (ISCA '05)*, pp. 432–443, June 2005.

[76] F. Vacca, G. Masera, H. Moussa, A. Baghdadi, and M. Jezequel, "Flexible architectures for LDPC decoders based on network on chip paradigm," in *Proceedings of the 12th Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD '09)*, pp. 582–589, August 2009.

[77] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proceedings of the IEEE Workshop on Signal Processing Systems Design and Implementation*, pp. 107–112, October 2004.

[78] C. Condo, *A parallel LDPC decoder with Network on Chip as underlying architecture*, M.S. thesis, Politecnico di Torino, 2010.

[79] C. Condo and G. Masera, "A flexible NoC-based LDPC code decoder implementation and bandwidth reduction methods," in *Proceedings of the Conference on Design and Architectures for Signal and Image Processing (DASIP '11)*, pp. 1–8, November 2011.

[80] Z. Chen, X. Zhao, X. Peng, D. Zhou, and S. Goto, "An early stopping criterion for decoding LDPC codes in WiMAX and WiFi standards," in *Proceedings of the IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems (ISCAS '10)*, pp. 473–476, June 2010.

[81] M. Martina and G. Masera, "Turbo NOC: a framework for the design of network-on-chip-based turbo decoder architectures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 10, pp. 2776–2789, 2010.

[82] C. Condo, M. Martina, and G. Masera, "A Network-on-Chip-based high throughput turbo/LDPC decoder architecture," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '12)*, 2012.

[83] M. Imase and M. Itoh, "A design for directed graphs with minimum diameter," *IEEE Transactions on Computers*, vol. 32, no. 8, pp. 782–784, 1983.

[84] F. Naessens, B. Bougard, S. Bressinck et al., "A unified instruction set programmable architecture for multi-standard advanced forward error correction," in *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS '08)*, pp. 31–36, October 2008.

Hindawi Publishing Corporation
VLSI Design
Volume 2012, Article ID 942893, 10 pages
doi:10.1155/2012/942893

Review Article

Design Space of Flexible Multigigabit LDPC Decoders

Philipp Schläfer,[1] Christian Weis,[1] Norbert Wehn,[1] and Matthias Alles[2]

[1] *Microelectronic Systems Design Research Group, University of Kaiserslautern, 67663 Kaiserslautern, Germany*
[2] *R&D Department, Creonic GmbH, 67655 Kaiserslautern, Germany*

Correspondence should be addressed to Philipp Schläfer, schlaefer@eit.uni-kl.de

Received 2 December 2011; Accepted 7 February 2012

Academic Editor: Guido Masera

Multigigabit LDPC decoders are demanded by standards like IEEE 802.15.3c and IEEE 802.11ad. To achieve the high throughput while supporting the needed flexibility, sophisticated architectures are mandatory. This paper comprehensively presents the design space for flexible multigigabit LDPC applications for the first time. The influence of various design parameters on the hardware is investigated in depth. Two new decoder architectures in a 65 nm CMOS technology are presented to further explore the design space. In the past, the memory domination was the bottleneck for throughputs of up to 1 Gbit/s. Our systematic investigation of column- versus row-based partially parallel decoders shows that this is no more a bottleneck for multigigabit architectures. The evolutionary progress in flexible multigigabit LDPC decoder design is highlighted in an extensive comparison of state-of-the-art decoders.

1. Introduction

Over the past years, a trend to ever increasing data rates could be observed which is enforced mostly by wireless applications. New standards like WiMedia 1.5 [1] for ultra wideband (UWB), IEEE 802.15.3c [2] for wireless personal area networks (WPANs), and IEEE 802.11ad [3] for Gigabit-WLAN (WiGig) define data rates of several Gbit/s. These standards allow for wireless broadband internet, wireless HDMI, and other advanced technologies. To approach the channel capacity while serving these high data rates, Low-density parity-check (LDPC) codes are excellent candidates, see Section 2. Even though LDPC codes belong to the best channel coding schemes known today, their implementation poses big challenges. This is due to the iterative, computational expensive decoding procedure, requiring massive parallel architectures. Therefore, design decisions concerning the decoder architecture have to be carefully evaluated.

Many papers have been published in the past using different approaches of (partially) parallel LDPC decoder architectures. The decoder design space including hardware mapping, node parallelism, as well as other parameters is discussed in detail in Section 3. Each of these parameters has a strong impact on the resulting hardware efficiency, the achievable throughput, and the communications performance.

This is the first paper comprehensively discussing the design space in particular for flexible multigigabit LDPC decoders. Due to the high degree of parallelism, there are new associations which have not been observed earlier. This paper discusses the multigigabit LDPC decoder design space and its influence on the hardware implementation. The benefits and drawbacks of single design decisions are pointed out. The impact a decision has on the hardware is directly shown by the comparison of decoders. This knowledge can be used to determine design decisions at the outset, based on the aimed target properties.

Section 5 shows the progress in multigigabit decoder architectures. Two new decoders are published for the first time in this paper. By these designs, the flexible multigigabit design space is further explored and important information about decoder architectures is gained.

2. LDPC Codes

LDPC codes [4] are linear block codes defined by a sparse parity check matrix H of dimension $M \times N$, see Figure 1(a).

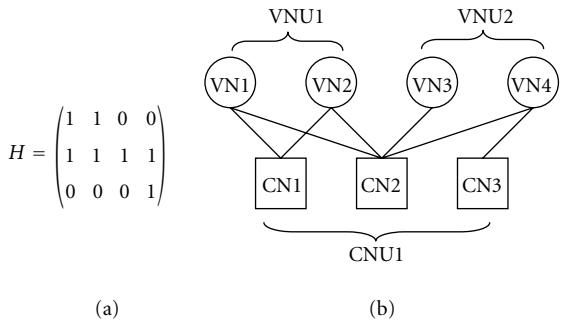


Figure 1: H matrix Tanner graph hardware mapping.

A valid code word $\vec{x}$ has to satisfy $H\vec{x}^T = \vec{0}$ in modulo-2 arithmetic. A descriptive graphical representation of the whole code is given by a Tanner graph. Each row of the parity check matrix is represented by a check node (CN) and corresponds to one of the M parity checks. Respectively each column corresponds to a variable node (VN) corresponding to one of the N code bits. The Tanner graph shown in Figure 1(b) is the alternative representation for the parity check matrix of Figure 1(a). Edges in the Tanner graph reflect the "1"s in the H matrix. There is an edge between VN n and CN m if and only if $H_{mn} = 1$. Figure 1(b) shows the mapping of the Tanner graph to hardware processing units. In the given example, each two VNs are mapped to one variable node unit (VNU). For the CNs, only one hardware instance, a check node unit (CNU), is instantiated to process all the CNs of the parity check matrix. This of course implies a time multiplexed operation of the nodes and spans a wide range of possible implementations. The different approaches are highlighted in Section 3 as they have great influence on the resulting architecture. However, random-structured Tanner graphs pose big problems for hardware implementations. The interconnection network between the different kind of nodes has to process the random message exchange, which results in inefficient architectures. For the construction of structured LDPC codes, shifted identity matrices I^x of size $P \times P$ and the zero matrix are used as submatrices in H, (1).

$$I^1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \tag{1}$$

Equation (2) shows the construction principle of the H matrix. This matrix structure in the first place allows for the exploitation of the algorithm's inherent parallelism by parallel instantiation of the functional units.

$$H = \begin{pmatrix} H_{1,1} & \dots & H_{1,N/P} \\ H_{2,1} & \dots & H_{2,N/P} \\ \vdots & \ddots & \vdots \\ H_{M/P,1} & \dots & H_{M/P,N/P} \end{pmatrix} H_{mp,np} = \begin{cases} 0, \\ I^x. \end{cases} \tag{2}$$

LDPC codes can be decoded by the Sum-Product Algorithm (SPA). Probabilistic messages are iteratively exchanged between variable and check nodes.

First, the variable nodes are initialized with the channel values λ_n^{ch}. According to the connections in the parity check matrix, these values are passed from each variable node n with $n \in \{0, \dots, N-1\}$ to the connected check nodes $\mathcal{M}(n)$ with

$$\mathcal{M}(n) = \{m \mid m \in \{0, \dots, M-1\} \wedge H_{mn} \neq 0\}. \tag{3}$$

The set of all variable nodes connected to check node m with $m \in \{0, \dots, M-1\}$ is defined as

$$\mathcal{N}(m) = \{n \mid n \in \{0, \dots, N-1\} \wedge H_{mn} \neq 0\}. \tag{4}$$

The check node computation can be split in two parts, the actual parity check, represented by the sign $\operatorname{sgn}(\epsilon)$, and the result's probability represented by the magnitude $|\epsilon|$. These extrinsic messages are computed as follows:

$$\operatorname{sgn}\left(\epsilon_{m\to n}^{(i)}\right) = \prod_{n' \in \mathcal{N}(m)\setminus n} \operatorname{sgn}\left(z_{n'\to m}^{(i)}\right),$$

$$\left|\epsilon_{m\to n}^{(i)}\right| = \Phi^{-1}\left(\sum_{n' \in \mathcal{N}(m)\setminus n} \Phi\left(\left|z_{n'\to m}^{(i)}\right|\right)\right), \text{ with} \tag{5}$$

$$\Phi(x) = \Phi^{-1}(x) = -\ln\left(\tanh\left(\frac{x}{2}\right)\right).$$

Finally, the messages are returned to the variable nodes which generate new intrinsic messages $z^{(i)}$ for iteration i:

$$z_{n\to m}^{(i)} = \lambda_n^{ch} + \sum_{m' \in \mathcal{N}(n)\setminus m} \epsilon_{m'\to n}^{(i-1)} = \Lambda_n^{(i-1)} - \epsilon_{m\to n}^{(i-1)}, \tag{6}$$

with

$$\Lambda_n^{(i)} = \lambda_n^{ch} + \sum_{m' \in \mathcal{M}(n)} \epsilon_{m'\to n}^{(i)}. \tag{7}$$

The sign of the a posteriori probability (APP) Λ can be interpreted as the hard decision bit. Message exchange between variable and check node continues until either a valid code word is found or a maximum number of iterations is exceeded.

3. Decoder Design Space

Figure 2 gives an overview of the multigigabit LDPC decoder design space. Appropriate decisions have to be met in order to derive an area and energy efficient decoder while serving the demanded flexibility. The final decisions heavily depend on constraints like communications performance, throughput, area, or energy consumption, which are dictated by the applications and supported standards. The following sections present the complex interdependencies which have to be considered during the design of hardware decoders. The focus in this overview is on design decisions of multigigabit decoder architectures.

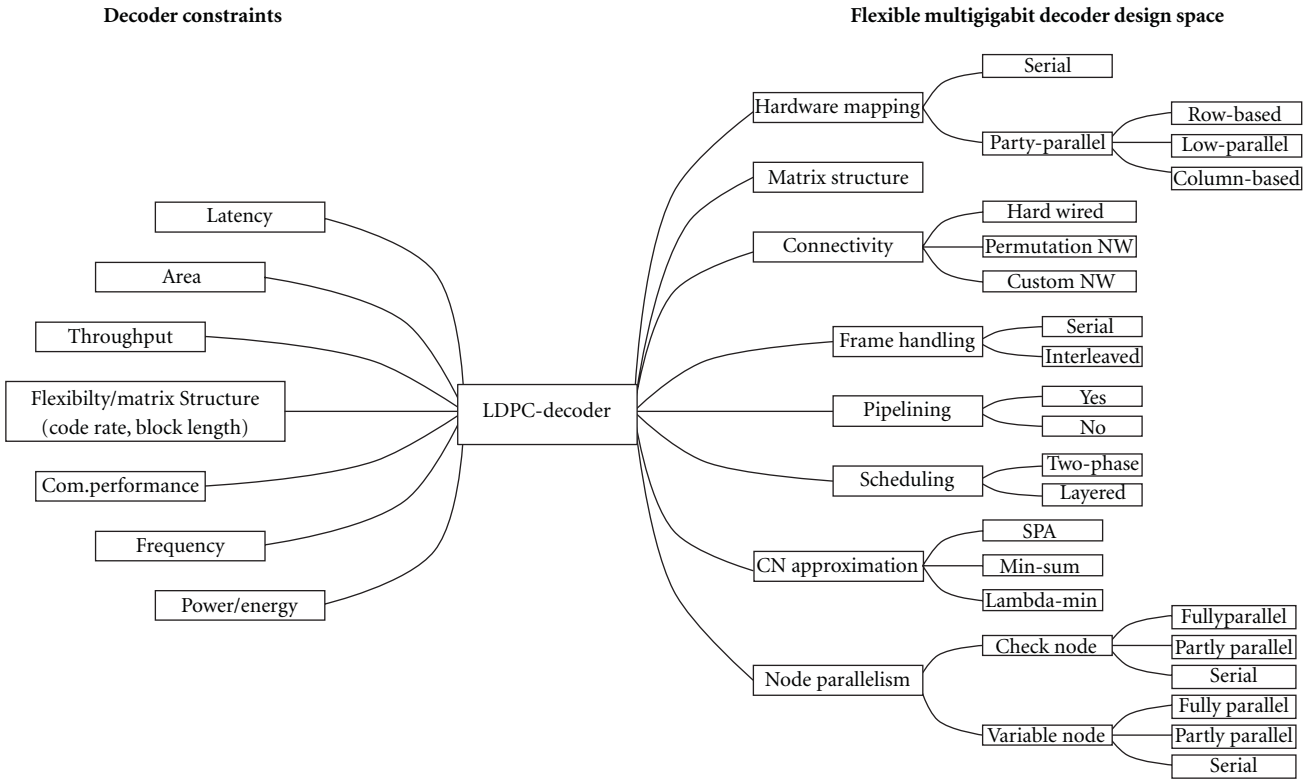


Figure 2: Flexible multigigabit LDPC decoder design space and decoder constraints.

3.1. Scheduling. A partially parallel decoder can use different node update schedules, two-phase, or layered scheduling. The two-phase schedule processes all check nodes before the variable nodes produce new messages. In the layered schedule, the check nodes are processed in a way that the variable nodes are intermediately updated [5–7]. Thus, the remaining CNs can take results into account which have been generated in the same iteration. The layered schedule can achieve the same communications performance as the two-phase schedule while reducing the needed iterations by about 50%. This allows for much more energy efficient decoders, reducing the frequency by 50% with the same throughput. Another possibility is to reduce the degree of parallelism and thus the decoder's area demand. Therefore, layered scheduling should be applied whenever possible.

3.2. Node Parallelism. The node degree defines how many edges a single node can read or write per clock cycle. This parameter can be configured independently for the check and variable nodes. A straight forward serial implementation reads one edge and produces one output value per cycle. This node architecture results in large flexibility as each node degree can easily be supported. However, serial node implementations are prohibitive for multigigabit decoders because of the moderate achievable throughput. For higher target throughputs, either the VNU, the CNU, or both node parallelisms must be increased. To get an efficient hardware utilization, the node degree must match the number of computational units generating data for the respective node. In this case, all produced data can be processed without introducing wait states. The required node parallelism thus needs to be derived from the chosen hardware mapping. Finally, the node degree is also a factor with strong impact on the functional unit size. The more edges a node can process in parallel the bigger it gets.

3.3. Hardware Mapping. The hardware mapping is one of the most important parameters for the decoder design because it has great impact on the resulting hardware. Three kinds of implementation styles for the Tanner graph exist: Fully parallel, partly parallel, and serial processing. For serial processing, only one instance of each functional unit, one VNU, and one CNU are implemented, gaining high flexibility but only moderate throughput. A fully parallel hardware mapping implementing all nodes of the Tanner graph is also possible, gaining high throughput. However, this hardware mapping lacks flexibility and is limited to small block sizes. Routing congestion is another obstacle when applying this approach. In [8] it is reported, that only 50% of the chip is used by logic for a fully parallel mapping. A possibility to relax the routing congestion problem is the use of bit-serial networks which reduce the node interconnections. As shown in [9] very efficient fully parallel decoder architectures are achievable by this approach. However fully parallel decoders lack the flexibility which is mandatory for the most current standards. Therefore, we will focus only on partly parallel hardware mappings in this paper. This still includes a wide span of decoder architectures which can be distinguished

Table 1: Multigigabit decoder hardware mapping.

Hardware mapping	CNU instances	VNU instances
Low-parallel	$i \times P$	$j \times P$
Row-based	$i \times P$	N
Column-based	M	$i \times P$

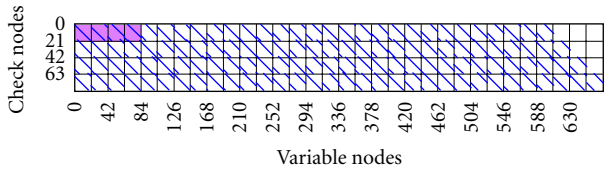


Figure 3: State-of-the-art decoder processing P CNs and $4 \times P$ VNs per cycle.

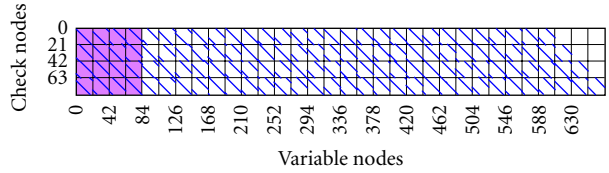


Figure 4: Column-based decoder processing M CNs and $4 \times P$ VNs per cycle.

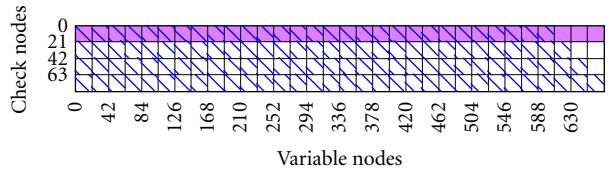


Figure 5: Row-based decoder processing P CNs and N VNs per cycle.

by the ratio of VNUs to CNUs. Basically, there are three approaches, a low-parallel, a column-based, and a row-based decoder. Table 1 points out the respective VNU to CNU ratios.

The low-parallel approach is useful for moderate throughput demands of up to 1 Gbit/s. It instantiates a small number of VNUs and CNUs and thus needs a huge number of cycles per iteration. Figure 3 shows the processing of four submatrices per cycle. The highlighted area represents the part of the parity check matrix which is processed in the first cycle. Overall this architecture needs 32 cycles per iteration. This hardware mapping requires partly parallel CNUs, processing j edges per cycle. The VNUs work serially, consuming and generating one edge per cycle. Overall only few functional units need to be instantiated. However, for low-parallel decoder architectures, we observe memory dominations of up to 70%. Our investigations have shown that the dominant memory is the extrinsic message RAM storing the messages which have to be passed from CNs to VNs. The poor ratio of logic to memory area is a result of the low parallelism and cannot be improved for this kind of hardware mapping.

The column-wise decoding making a one to one mapping of check nodes to CNUs can overcome this drawback. An example for a column-based decoder is given in Figure 4. All M CNUs and $4 \times P$ VNUs are instantiated. This architecture requires partly parallel CNUs, processing $4 \times P$ edges per cycle. Moreover, the VNUs need to be implemented fully parallel, consuming and generating M edges per cycle. Due to the higher degree of parallelism, this hardware mapping needs only eight cycles per iterations. All CNUs work in parallel and thus produce the CN to VN messages just in time. This makes memories between CNs and VNs redundant and can save a significant amount of area. However, for decoders instantiating all M CNUs for column-wise decoding, the hardware mapping restricts the possible scheduling to the two-phase approach. This is due to the parallel working CNUs making intermediate VNU updates impossible. Moreover, our investigations have shown that, due to their complexity, CNUs need significantly more area than VNUs. Especially for highly parallel decoders, instantiating a huge number of functional units, this fact is most important. In Section 5, it is shown that, for the same area, a higher degree of parallelism can be achieved by parallel instantiation of VNUs instead of CNUs. We will also show that these two drawbacks of the column-based decoder overcome the benefit of saved area in comparison with the row-based design.

For a row-based decoder, all N VNs are instantiated as VNUs and a multiple of P number of CNUs. Figure 5 highlights the part of the parity check matrix H which is processed in the first cycle. Serially working VNUs and fully parallel CNUs processing N edges per cycle are required. The given example instantiates P CNUs and needs only four cycles per iteration.

Furthermore, all three approaches can be extended by processing not only one row or column but instead several of them in parallel. Due to the parity check matrix structure, usually a multiple of the submatrix size is chosen, which yields a good unit utilization, see Table 1.

3.4. Pipelining. Adding register stages in the decoder core efficiently shortens the critical paths. This can be exploited to reduce the area by smaller but slower cells, or higher frequencies can be achieved within the same area. Regardless of these benefits, the pipeline has to be filled each iteration, which introduces a clock cycle penalty per stage and iteration. In most serially working decoders, this does no harm to the overall throughput because even without the pipeline delays, a high number of cycles is needed per iteration. In contrast to this, highly parallel architectures are processing one complete iteration in only a few clock cycles, often less than ten. For these decoders, each pipeline stage adding one cycle to the iteration period has an enormous impact on the throughput. Therefore, for massive parallel decoder architectures, the number of pipeline stages has to be kept as low as possible.

3.5. Frame Handling. The latency introduced by pipelining has strong impact on massive parallel iterative decoders

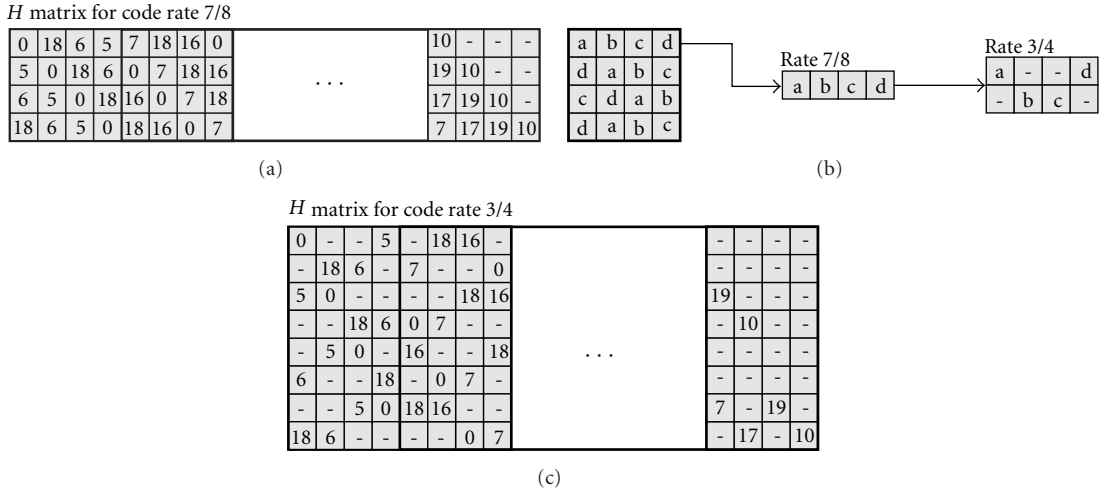


Figure 6: Row split pattern for IEEE 802.15.3c parity check matrix.

due to the data dependencies. To overcome this problem, several independent frames can be processed by the decoder in parallel. This frame-interleaved pipelining is comparable to pipelining as it is done in a modern general purpose CPU. A very high hardware utilization can be achieved by this technique because pipeline stalls hardly appear in this application. Due to the additional complexity and implementation effort introduced by this approach, there are only few decoders making use of this feature. However, for future decoders especially in the high throughput domain, where only few cycles per iteration are needed, handling of multiple frames should be concerned.

3.6. Matrix Structure Design. The parity check matrix H has significant impact on the decoder design. For high throughput applications-typically very sparse matrices are used to reduce the computational complexity. An innovation introduced in upcoming standards like [2, 3] is a special method to generate matrices for different code rates. By the so-called row splitting, the matrices for lower code rates are derived from one base matrix. Figure 6(a) shows the base matrix of the IEEE 802.15.3c code. The bold divided parts of the matrix are the so-called macroblocks. The principle of row splitting is presented in Figure 6(b). From each row of the base matrix, multiple rows for the lower code rate matrices are generated, see Figure 6(c). It can be observed that the number of nonempty entries never increases. This matrix design allows for efficient decoder hardware implementations. CNUs processing all nonzero entries of one macroblock need only little modification to split the inputs to several independently processed groups of inputs. Thus, the arising bigger macroblocks can still be processed by the same hardware, gaining flexibility while minimizing the hardware overhead.

3.7. Check Node Approximation. The optimal belief propagation algorithm SPA as it is presented in Section 2 is usually replaced by suboptimal solutions. Dependant on the requirements, an appropriate approximation needs to be chosen. If the requirements are tight, a more accurate algorithm has to be chosen, for example, the λ-Min [10] algorithm allows for a good communications performance for a wide range of code rates. If a performance degradation for low code rates is acceptable, a Min-Sum-based [11] algorithm can reduce the implementation complexity even further. This is also the algorithm of choice in multigigabit applications where a high degree of parallelism is applied, and thus the use of a suboptimal algorithm saves a significant amount of area. Additionally, applying extrinsic memory compression [10] can reduce the need for extrinsic message RAMs. Therefore, the extrinsic messages are not stored for each outgoing edge, but reconstructed, in the case of Min-Sum from only two messages. All decoders currently published for multigigabit standards make use of the Min-Sum Algorithm and apply extrinsic memory compression.

3.8. Connectivity. Different types of connectivity can be considered. For instance, fully parallel decoder architectures use a hard-wired connectivity. For partly parallel decoders however, flexible permutation networks are required [12, 13]. The most common interconnection networks are barrel shifters. They support the cyclically shifted identity matrices of structured codes. As mentioned before upcoming standards like IEEE 802.15.3c use a special row split pattern to generate different code rates. For efficient support of these kind of codes, custom networks are required in addition to the shifters. Furthermore, the parallelism of the interconnection network has to be considered. Most high throughput decoders found in literature pass messages bit-parallel. However, this results in routing dominated architectures. Bit-serial message passing drastically reduces the need for interconnection wires and relaxes the routing congestions. This leads to very power and area efficient fully parallel decoder designs as shown in [9]. Currently, there are to the best of our knowledge no publications applying

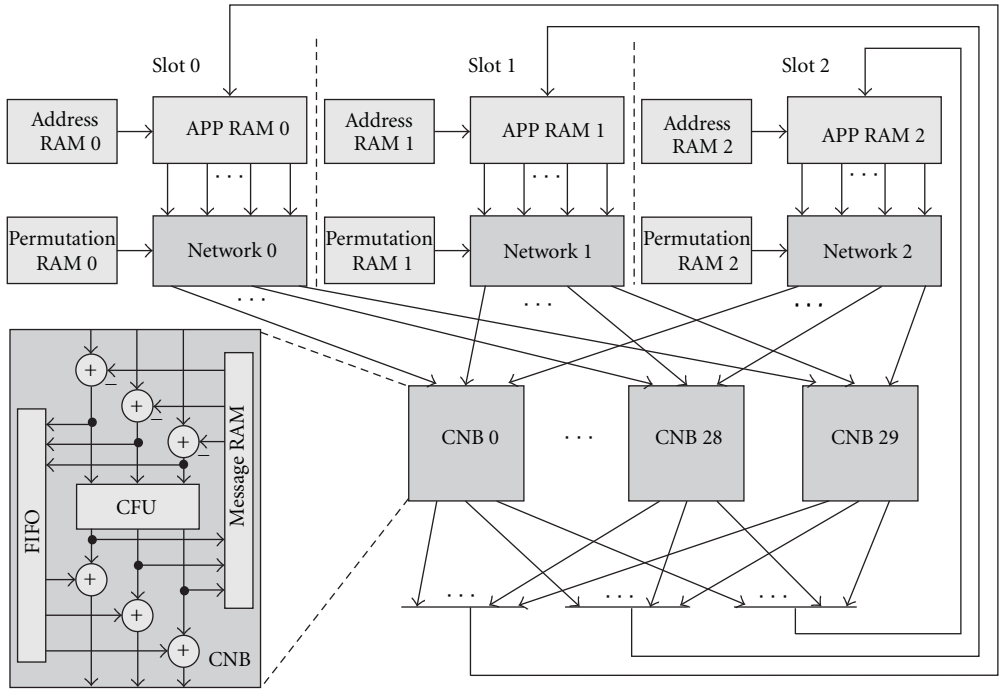


Figure 7: Slot-Layered low-parallel LDPC decoder, WiMedia 1.5 based.

bit-serial message passing to flexible partly parallel decoder architectures.

All these issues must not be considered as isolated decisions, they rather interact. In [14, 15], the interrelations are discussed in more detail. For example, the scheduling has impact on the required decoder parallelism, or a hard-wired interconnection is prohibitive for partly parallel architectures.

4. Decoder Architectures

The discussed design space allows for a variety of architectures. We present three decoder architectures in more detail to give a better understanding of the interoperation of the numerous design decisions. Because the hardware mapping has a major impact on the resulting architecture, each decoder uses a different mapping.

The first decoder we present makes use of a low-parallel hardware mapping and is designed for a WiMedia 1.5 based code. The architecture uses the concepts presented in [16], it is shown in Figure 7. Each set of 30 variable nodes is attributed to one of the three slots. One slot processes one submatrix per clock cycle. In contrast to a serial architecture, each of the check node blocks (CNBs) reads and writes three edges in parallel, with each edge coming from a different slot. The minimum search within the CNU is performed according to the Min-Sum algorithm. The variable node operations according to (6) and (7) are processed within the CNBs and do not require an explicit functional unit. This architecture gains an excellent flexibility while serving throughputs of more than 1 Gbit/s. Moreover, it supports layered decoding, resulting in much faster convergence than a comparable two-phase decoder.

The next approach we present implements the column-based hardware mapping. Figure 8 shows the decoder's basic architecture. The main building blocks are the VNUs, CNUs, and the permutation networks. As mentioned before, the different code rate matrices of the IEEE 802.15.3c standard are generated by the presented row-split pattern. A specialty for this decoder is the custom networks (CNWs) which are introduced before and after the CNUs. They are used to support the different code rates and introduce only little hardware overhead. This decoder shows how the extrinsic message RAM can be removed by the fully parallel instantiation of CNs. As explained before, the four groups of CNUs process all CNs at the same time. Thus extrinsic messages can immediately be consumed by the variable nodes. However, this architecture is only capable of two-phase decoding. This leads to a significantly increased number of iterations and thus lowers the efficiency of the architecture as will be discussed in Section 5.

Finally, a row-based hardware mapping for the IEEE 802.15.3c standard is investigated. The high parallelism of the decoder arises from the fully parallel instantiated VNUs. 672 VNUs and 21 CNUs in number are used to process one quarter of the parity check matrix in parallel. The basic structure is similar to the one of the low parallel decoder presented before. However, instead of only three, now 32 slots are instantiated, see Figure 9. This architecture contains an extrinsic RAM which is needed because the matrix is processed row-wise, the intermediate data has to be stored for the next iteration. Because the VNUs work serially, each

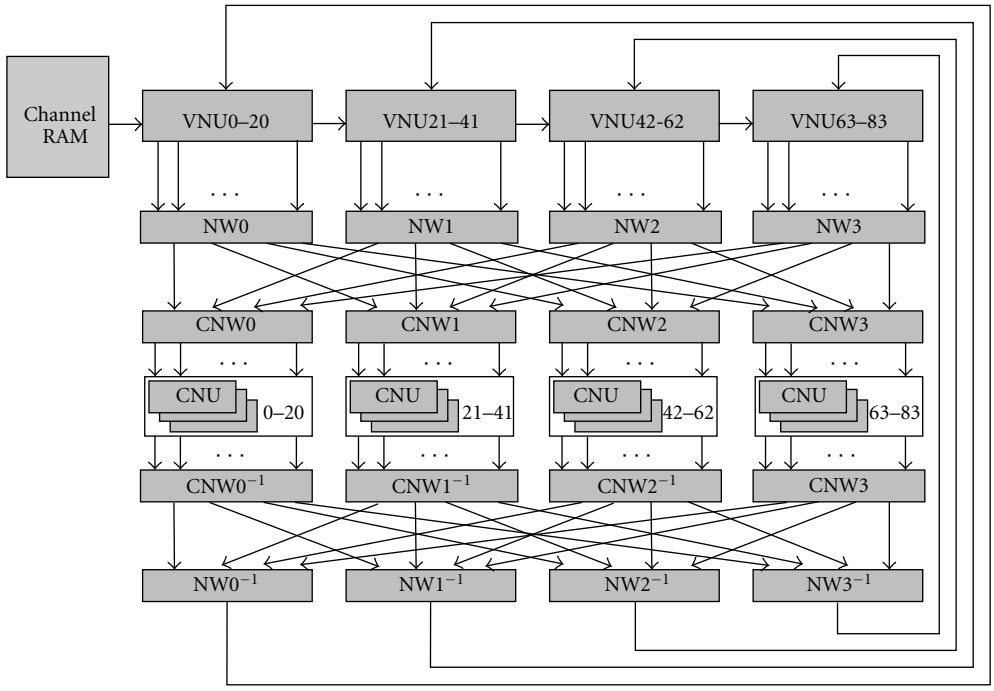


Figure 8: Column-based LDPC decoder architecture for IEEE 802.15.3c standard.

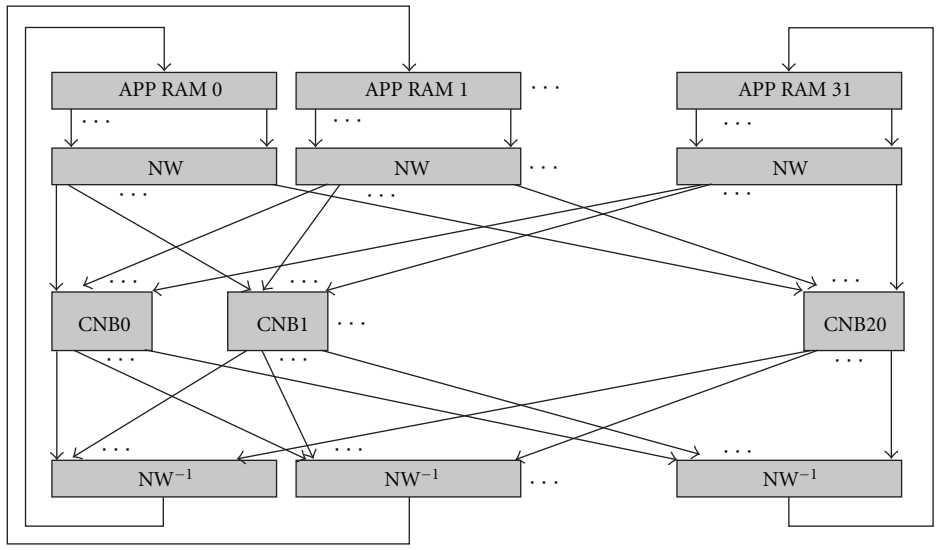


Figure 9: Row-based LDPC decoder architecture for IEEE 802.15.3c standard.

producing and consuming one edge per cycle, they are small in comparison with the partly parallel ones of the column-based decoder. The CNUs each are capable to process 32 edges which makes them complex to implement. However, the small number of CNU instances keeps the overall area demand in an acceptable range. Finally, the architecture has a very good VNU to CNU ratio, which results in a high efficiency as we discuss in the following section.

5. Decoder Evolution

Several approaches to design flexible high throughput LDPC decoders have been published in the past. However it is not yet investigated which of these designs is preferable in means of area and energy efficiency.

Estimating the decoders parameters before the implementation is a very complex task. Many factors like the

Table 2: Multigigabit decoder evolution.

	This paper	WPAN [17]	This paper	WPAN [18]	WiGig [19]	WPAN [20]
Standard	WiMedia 1.5-based	IEEE 802.15.3c draft	IEEE 802.15.3c	IEEE 802.15.3c	IEEE 802.11ad	IEEE 802.15.3c
Technology	65 nm	65 nm	65 nm	65 nm	65 nm	65 nm
Postplace and route	No	No	Yes	Yes	No	Yes
Codeword length	600	576	672	672	672	672
Code rates	1/2, 5/8, 3/4, 4/5	1/2, 3/4, 7/8	1/2, 5/8, 3/4, 7/8	1/2, 5/8, 3/4, 7/8	1/2, 5/8, 3/4, 13/16	1/2, 5/8, 3/4, 7/8
Submatrix size	15	18	21	21	42	21
Edges/cycle	45	288	336	672	672	672
$F_{\max}$ **[MHz]**	400	500	500	197	150	400
Scheduling	Layered	Two-phase	Two-phase	Layered	Two-phase	Layered
Algorithmic iterations	5	7	9	4	N/A	9
Cycles/iteration	40–45	10	11	4	4	4
Pipeline stages	3	2	3	0	4	4
Frame interleaved	No	No	No	No	Yes	Yes
Input quantization	6	6	5	6	N/A	6
Hardware mapping	Other	Column-based	Column-based	Row-based	Row-based	Row-based
VNU degree	1	4	4	1	1	1
CNU degree	3	1–4	1–4	8–32	8–16	8–32
VNU instances	45	72	84	672	672	672
CNU instances	15	72–288	84–336	21–84	42–84	21–84
Logic Area	30%	N/A	62%	N/A	N/A	68%
Memory area	70%	N/A	38%	N/A	N/A	32%
Area [mm^2]	0.37	0.50	1.15	1.56	1.30	1.30
Supply [V]	1.2	N/A	1.2	1.0	0.8	1.2
Power [mW]	N/A	N/A	630	361	84	538
Energy eff. [pJ/bit/Iter]	N/A	N/A	21	13	7	8
Thr. [Gbit/s/mm^2]	2.62	7.20	2.66	3.24	2.37	5.17
Thr. [bit/Cycle]	2.43	7.20	6.11	33.60	20.50	16.80
Thr. Air [Gbit/s]	0.97	3.60	3.06	6.62	3.08	6.72

matrix structure, underlying memory generators as well as the supported flexibility, have huge impact on the results and cannot be expressed analytically. For fully parallel decoders without flexibility, estimations are possible and recently published in [21]. Table 2 gives a detailed comparison of flexible multigigabit decoders. All the design space parameters presented in the previous section are pointed out. Thus, the table can be used to get an impression of the influence of the design space parameters.

A large overhead is introduced by routing which impacts area and energy. It makes synthesis results only not conclusive. Only for three of the decoders postplace and route (PAR), results are available which have to be considered for comparison.

State-of-the-art LDPC decoders process only few submatrices per cycle, and their area is dominated by memory. They make use of a low-parallel hardware mappings which results in poor logic to memory ratios as pointed out earlier. In the WiMedia 1.5 based decoder, we present in Table 2, 70% of the overall area is spent for memory.

As described in Section 3, instantiating all CNUs can solve this issue. A decoder using this approach for a draft of the IEEE 802.15.3c standard was first published in [17]. However, the mentioned paper presents only synthesis data which are not comparable to post PAR results. Therefore, we present a column-based decoder built in a 65 nm process using a low-power, low V_T library. Synthesis as well as PAR is executed under worst case conditions, ending up with 74% standard cell utilization. The results in Table 2 show the achieved logic domination for this architecture. Figure 10 underlines the area demand of the CNU groups. Only 38% of the overall area is used by memory, which is mainly made up by the channel RAM and pipeline registers. However, it has been shown that, due to their complexity, CNUs need more area than VNUs. For a given area, a higher parallelism can be achieved by increasing the number of VNUs instead of the CNUs as discussed in the previous section. The column-based design, using massive parallel instantiated CNUs, thus needs more area than a rowbased design with the same degree of parallelism. This can be

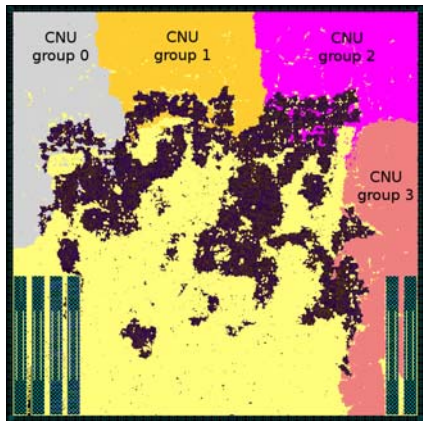


Figure 10: Physical design of the column-based LDPC decoder. The four check node groups are located in the top and right. The channel RAM is split in two parts and located in the bottom corners.

seen in the direct comparison of our column-based and the row-based decoder presented in [18]. Even though all 672 VNUs are instantiated and thus the parallelism is doubled in comparison to the column-based decoder, there is only a small growth in area. This yields an increased throughput per area result of 3.24 Gbit/mm^2.

For a further comparison of the logic to memory area ratio, the decoder presented in [20] is considered. Even though the row-based design needs extrinsic message RAM for the check to variable node messages, overall only 32% of the area is memory area. The memory domination seen in low parallelism decoders like the WiMedia-based decoder presented in this paper is no more an issue for massive parallel decoders. Due to the high number of instantiated VNUs, the RAMs play only a minor role in the overall area utilization. The other important difference between the row and the column-based decoder is the scheduling. While the column-based decoder is restricted to two-phase scheduling, row-based decoders can use layered scheduling. By this difference, the communications performance can be improved or, like in [18], the throughput can be enhanced by reduction of the iterations.

An architecture proposed in [19] for the IEEE 802.11ad standard uses a similar approach as the one presented before. The decoder is also row-based and thus instantiates all variable nodes. Even though area and power consumption are only estimated, the results show that this approach can also be adapted to other multigigabit applications.

The next evolutionary step is taken by the decoder presented in [20]. By the interleaved frame handling, regardless of the pipelining, no wait state is introduced. Finally, this represents the most advanced decoder architecture for IEEE 802.15.3c and is also a candidate for other multigigabit standards featuring LDPC codes.

6. Conclusion

In this paper, we have in detail presented the design space for flexible multigigabit LDPC decoders. The memory domination as known from state-of-the-art decoders is investigated in the context of highly parallel decoder architectures. It is pointed out that the need for memories can be reduced by column-based decoders because intermediate message storage becomes obsolete. Our systematic investigations have shown that, in row-based designs, the memory domination disappears because of the large amount of logic introduced by the high degree of parallel instantiated functional units. Further exploring the design space, two new decoder architectures serving throughputs of one and three Gbit/s are presented including post PAR results. Especially row- and column-based hardware mappings are investigated. It is shown that the row-based hardware mapping has significant benefits compared to column-based designs because of the VNU to CNU ratio and the possibility to apply layered scheduling.

Acknowledgment

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) within the project "Entwicklung effizienter und flexibler VLSI-Architekturen für die Kanaldecodierung in drahtlosen Multi-Gigabit-Kommunikationssystemen auf der Basis von LDPC-Codes."

References

[1] WiMedia Alliance, "Multiband OFDM physical layer specification, release candidate 1.5," March 2009.

[2] IEEE 802.15.3c, "Part 15.3: wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPANs)," IEEE 802.15.3c-2009, 2009.

[3] IEEE 802.11ad, ""Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications—amendment: enhancements for very high throughput in the 60 GHz band," IEEE 802.11ad-draft, 2010.

[4] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[5] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," in *Proceedings of the 36th Asilomar Conference on Signals Systems and Computers*, pp. 8–15, November 2002.

[6] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proceedings of the IEEE Workshop on Signal Processing Systems Design and Implementation (SiPS'04)*, pp. 107–112, Austin, Tex, USA, October 2004.

[7] J. Dielissen, A. Hekstra, and V. Berg, "Low cost LDPC decoder for DVB-S2," in *Proceedings of the Design, Automation and Test in Europe (DATE'06)*, Munich, Germany, March 2006.

[8] A. J. Blanksby and C. J. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, 2002.

[9] M. Korb and T. G. Noll, "Area and latency optimized high-throughput Min-Sum based LDPC decoder architectures," in *Proceedings of the 35th European Solid-State Circuits Conference (ESSCIRC'09)*, pp. 408–411, September 2009.

[10] F. Guilloud, E. Boutillon, and J. Danger, "λ-Min decoding algorithm of regular and irregular LDPC codes," in *Proceedings of the 3rd International Symposium on Turbo Codes & Related Topics*, pp. 451–454, Brest, France, September 2003.

[11] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, 2005.

[12] G. Masera, F. Quaglio, A. Tarable, and F. Vacca, "Interconnection structure for a flexible LDPC decoder," in *Proceedings of the Wireless Reconfigurable Terminals and Platforms (WiRTeP'06)*, pp. 58–62, April 2006.

[13] M. Rovini, G. Gentile, F. Rossi, and L. Fanucci, "A scalable decoder architecture for IEEE 802.11n LDPC codes," in *Proceedings of the Global Telecommunications Conference (GLOBECOM'07)*, Washington, DC, USA, November 2007.

[14] T. Brack, F. Kienle, and N. Wehn, "Disclosing the LDPC code decoder design space," in *Proceedings of the Design, Automation and Test in Europe (DATE'06)*, pp. 200–205, Munich, Germany, mARCH 2006.

[15] F. Kienle, *Implementation issues of low-density parity-check decoders*, Ph.D. dissertation, University of Kaiserslautern, 2006.

[16] M. Alles, F. Berens, and N. Wehn, "A synthesizable IP core for WiMedia 1.5 UWB LDPC code decoding," in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB'09)*, pp. 591–601, Vancouver, Canada, September 2009.

[17] J. Sha, J. Lin, Z. Wang, L. Li, and M. Gao, ""Ldpc decoder design for high rate wireless personal area networks," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 455–460, 2009.

[18] S.-Y. Hung, S.-W. Yen, C.-L. Chen, H.-C. Chang, S.-J. Jou, and C.-Y. Lee, "A 5.7 Gbps row-based layered scheduling LDPC decoder for IEEE 802.15.3c applications," in *Proceedings of the IEEE Asian Solid-State Circuits Conference (A-SSCC'10)*, pp. 309–312, Beijing, China, 2010.

[19] M. Weiner, B. Nikolic, and Z. Zhang, "LDPC decoder architecture for high-data rate personal-area networks," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'11)*, pp. 1784–1787, 2011.

[20] Z. Chen, X. Peng, X. Zhao et al., "A macro-layer level fully parallel layered LDPC decoder SOC for IEEE 802.15.3c application," in *Proceedings of the International Symposium on VLSI Design, Automation and Test (VLSI-DAT'11)*, pp. 298–301, 2011.

[21] M. Korb and T. G. Noll, "Ldpc decoder area, timing, and energy models for early quantitative hardware cost estimates," in *Proceedings of the International Symposium on System-on-Chip (SoC'10)*, pp. 169–172, 2010.

Hindawi Publishing Corporation
VLSI Design
Volume 2012, Article ID 870546, 15 pages
doi:10.1155/2012/870546

Research Article

Power Consumption Models for Decimation FIR Filters in Multistandard Receivers

Khaled Grati,[1] Nadia Khouja,[1] Bertrand Le Gal,[2] and Adel Ghazel[1]

[1] *Cirta'com Laboratory, Ecole Supérieure des Communications de Tunis, 2083 Ariana, Tunisia*
[2] *IMS Laboratory, Université de Bordeaux I, Bordeaux, 33405 Talence, France*

Correspondence should be addressed to Nadia Khouja, nadia.khouja@supcom.rnu.tn

Received 3 December 2011; Revised 9 March 2012; Accepted 15 March 2012

Academic Editor: Frank Kienle

Decimation filters are widely used in communication-embedded systems. In fact, decimation filters are useful for implementing channel filtering or selection with low-computation complexity requirements. Many multistandard receiver designs that are required in ubiquitous embedded systems are based on a cascade of decimation filter processing. Filter number and implementation architectures have a significant impact on system performances, such as computation complexity, area, throughput, and power consumption. In this work, we present filter power consumption estimation models for FIR filters. Power consumption models were obtained from a large number of FIR filter syntheses using a direct form. Several curves that estimate power consumption were extracted from these synthesis results. Then, we have evaluated the impact of polyphase decomposition on power consumption of FIR filter and compared it with the direct form results. Some tips regarding power consumption were deduced for the polyphase implementation form. The aim of this work is to help a system designer to select an efficient implementation for FIR in terms of power consumption without having to implement and synthesize the different possible solutions. The proposed method is applied for STMicroelectronics libraries 90 nm and 65 nm low power then validated with a use case of multistandard receiver designing.

1. Introduction

Currently, wireless technologies are widespread because of their flexibility of use. However, many different standards are used, and a new challenge for the communication-embedded system designer is to implement multiple communication standard devices in order to provide easy access to information everywhere with low hardware complexity. In general, such devices include communication chains with decimation and selector filters. Typically, multiple standards communication chains are composed of an RF front end, an over-sampler analogue to digital converter, and a cascade of decimation filters (Figure 1) [1]. The power consumption is an important constraint during embedded system design because the design of decimation filters has a substantial impact on the power consumption in multistandard receivers. This work focuses on FIR filter power consumption estimation in direct form (Figure 2). The polyphase form of FIR filters is widely recommended for reducing power consumption in comparison with all possible implementation forms of these filters. To the best of our knowledge, there has not been a clear study based on experimental results showing how much power FIR filters consume. In fact, the work by Dumonteix et al. [2] is widely mentioned, and it deals only with the power consumption, area and critical path of a particular implementation of a comb filter. It was shown in this work that appropriate filter decomposition in association with polyphase decomposition could lead to an important significant consumption of the CIC filter.

The main objective of this work is to provide models that evaluate FIR decimator filter power consumption in direct form. These models consider the main filter parameters, which are the filter order, the input wordlength, and the coefficient wordlength. This model was given for STM90 nm and STM60 nm low power. The operation conditions are specified in paragraph 2.

The second objective of this paper is to study the impact of the polyphase form on the power consumption of FIR filters. Some tips regarding the best decomposition to

Table 1: Dynamic and static power contribution using a 90 nm library of STMicroelectronics.

Filter order	Dynamic power (uW)	Dynamic power (%)	Static power (uW)	Static power (%)
13	300,9546	48,84	315,2013	51,16
22	481,3064	48,26	515,8453	51,74
31	632	46,71	721	53,29
40	722	43,54	936	56,46
49	840,64	42,63	1131,1	57,37
58	973,3182	41,87	1351,3	58,13
64	1054,6	41,62	1478,9	58,38

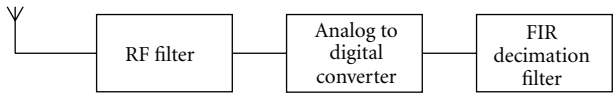


Figure 1: General communication chain in a multistandard receiver.

perform to save power are extracted from the synthesis results in this work.

The ultimate aim of this study is to help filter designers decide the best way to decompose the filter processing into stages to guarantee optimal power consumption.

This paper is organized as follows. Section 2 presents the specifications of the symmetric FIR filter that is used to estimate the power consumption of a decimation FIR filter. In Section 3, we present the power consumption models that are obtained for the direct implementation form. In Section 4, the models are validated by using synthesis results. In Section 5, we present the implementation results for the polyphase form usage. A use case based on established models is presented in Section 6. Finally, conclusions are given in Section 7.

2. Characterization of Decimator FIR Filter Parameters

When designing FIR channel selection and a decimation filter, the filter designer faces a trade-off between channel selection efficiency and filter complexity. Indeed, the design must guarantee channel selection with the minimum complexity in terms of occupied area and power consumption. Depending on the communication standard, filter designers choose the optimal filter order and coefficient length to guarantee the required signal-to-noise ratio. The input data wordlength is typically deduced from the analogue to digital converter input signal dynamic range. Hence, for the hardware performance evaluation, the designer should consider the three following parameters in FIR filter power consumption: input wordlength, filter order, and coefficient wordlength.

To evaluate the impact of these parameters on the power consumption of an FIR filter, we performed a large number of FIR filter architecture syntheses. We considered filters orders from 7 to 64 (with a step of three). We chose to implement both direct and polyphase forms. To reduce the hardware implementation complexity of the filter, we used a Wallace adder tree [3, 4] to perform the addition of all multiplication results (Figure 2).

For the polyphase decomposition, each subfilter was implemented as a FIR filter in the direct form. For simplicity, we chose to run the syntheses with the input wordlength equal to 4 bits, 8 bits, 16 bits, and 32 bits. In the next section, we will show that, for intermediate values, it is possible to interpolate the power consumption of the filter.

In the same way, we choose three possible values of filter coefficients: 4 bits, 8 bits, and 16 bits. In fact, both the filter order and coefficient size depend on the filter's mask. We estimate that 16 bits offer enough accuracy for quantization process.

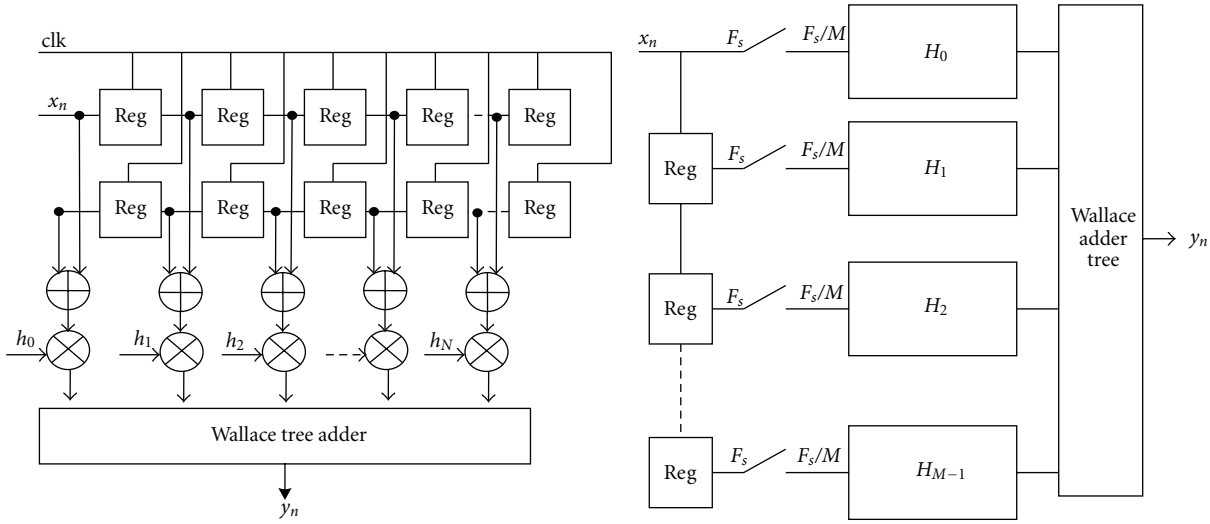
The performance estimation of FIR filters was done on STM90 nm process technology. In this 90 nm process library, the static power has almost the same proportions as the dynamic power consumption (see Table 1). For this reason we proposed two separate models for the dynamic and static power consumption. The performance estimation of FIR filters was also done on ASIC 65 nm process technology using a STMicroelectronics low-power library. Using this library, the power consumption of the FIR filters is reduced to the dynamic contribution because the static power is very low (see Table 2). Using the STM 65 nm low-power technology permits to deduce the dynamic power consumption model that can be verified later with the STM90 nm library process.

Design Vision of Synopsys was used to extract the performances on ASIC technology. From the experimental results, we were able to build a power consumption model of FIR filters depending on the three main filter parameters: input wordlength, coefficient wordlength, and filter order. Because the dynamic power consumption depends on frequency, the model for dynamic power consumption obtained is also frequency dependent.

3. Power Consumption Estimation Models

In this section, we introduce the power consumption estimation models for STM65 nm and STM90 nm process technologies. The STM65 nm library is low-power and operates at 0.9 V and used in nominal case with junction temperature of 25°C. The STM90 nm library operates at 1.26 V and is used in the best case with a junction temperature of 40°C.

3.1. Dynamic Power Consumption Model. For the dynamic power consumption model we used STM 65 nm low-power



(a) Direct form (b) Polyphase form

Figure 2: Implementation architecture of the direct form and polyphase form of FIR filter.

Table 2: Dynamic and static power contribution using a 65-nm low power library.

Filter order	Dynamic Power (uW)	Dynamic power (%)	Static Power (uW)	Static power (%)
13	169,6857	98,9	1,9	1,1
22	245,1208	98,8	3,1	1,2
31	356,7201	98,9	4,2	1,1
40	424,5542	98,7	5,7	1,3
49	497,6389	98,6	7,2	1,4
58	578,6831	98,7	8	1,3
64	628,4149	98,6	9,1	1,4

technology. In this technology, the dynamic power consumption is assumed to be the total power because the static power is very low (see Table 2). The Dynamic power consumption is assumed to be proportional to frequency. To verify this assumption, we performed several experiments to evaluate the impact of the frequency constraint, provided by the logical synthesis tool, on the occupied area and the power consumption of the generated design for a fixed filter order. Figure 3 confirms that the resources that are used to build the architecture for a given filter are the same regardless of the specified frequency constraint. Hence, the logical synthesis tool does not introduce any area or power optimization regardless of the working frequency. As a consequence, the dynamic power consumption is considered linear given the constrained frequency (Figure 3).

Following this observation, we concentrated all syntheses efforts at a fixed frequency equal to 80 MHz, which is sufficient for the requirements of both GSM and UMTS standards [5, 6].

Figure 4 gives the evolution of the power consumption versus the filter order for a direct form FIR filter for different values of coefficients and different inputs wordlength. This figure shows that power consumption is quite linear to the order of the filter for fixed input (4, 8, and 16 bits) and coefficient wordlength (4, 8, and 16 bits).

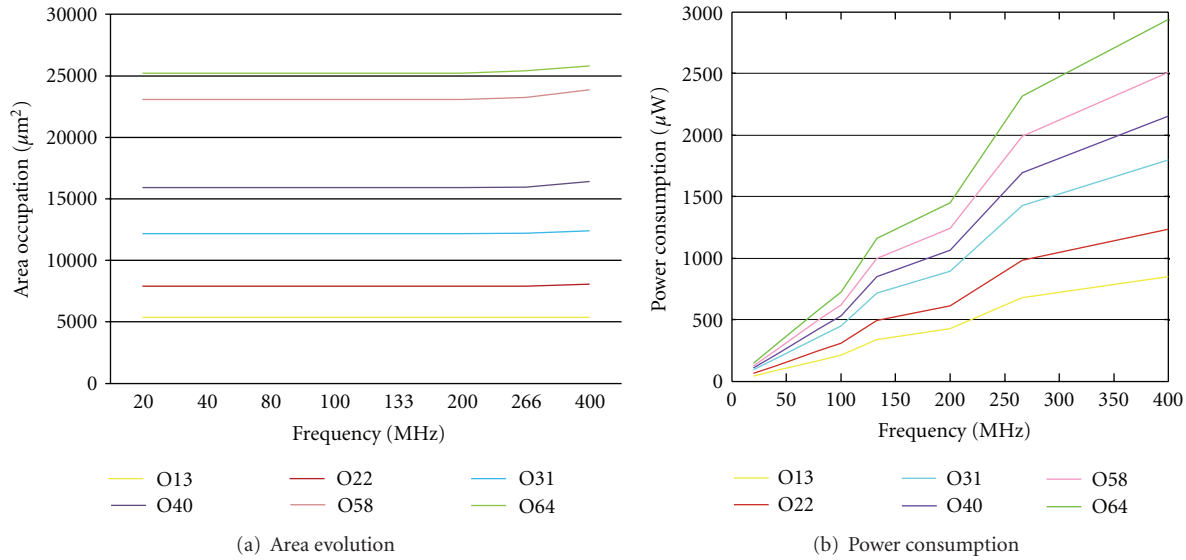
Figure 5 illustrates the relationship between the power consumption and input wordlength for four chosen orders and for a fixed coefficient value. For all other orders and for the different coefficient wordlengths, the evolution of the power consumption has the same trends. According to these curves, the power consumption evolution versus the input wordlength is not linear.

However, Figure 6 shows that the natural logarithm of the power consumption is almost linear as compared with the logarithm of the input wordlength. Hence, (1) gives the expression of the natural logarithm of the dynamic power consumption versus the natural logarithm of the input wordlength. This calculation leads to relation (2), which gives the power consumption general expression.

$$LN(P) = LN(\alpha) + \beta \times LN(I), \tag{1}$$

$$P = \alpha \times (I)^{\beta}, \tag{2}$$

where I is the input wordlength, β is the slope of the curves in Figure 6, and $LN(\alpha)$ represents the origin value of the



(a) Area evolution (b) Power consumption

Figure 3: Area occupation and power consumption of FIR filters in the direct form depending on frequency.

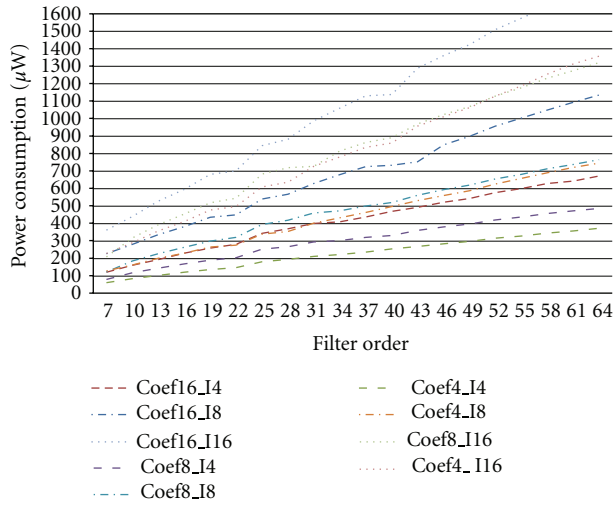


Figure 4: Power consumption of FIR filters in the direct form depending on input wordlength and coefficient wordlength.

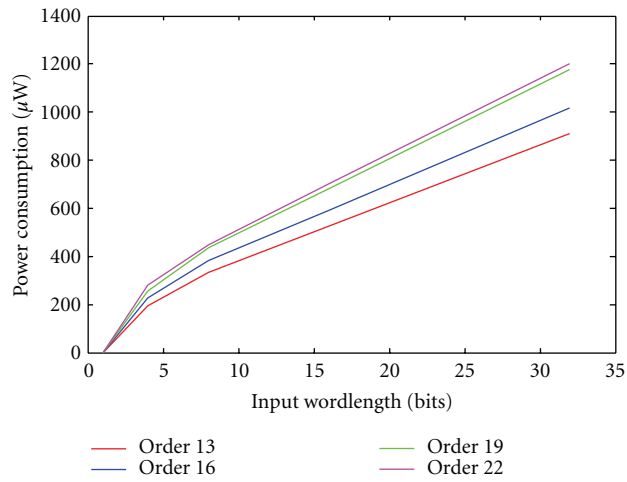


Figure 5: Power consumption of FIR filters in the direct form depending on input wordlength for the 65 nm technology for a fixed coef size of 4 bits.

same curves. Evaluating the α and β expressions depending on the filter parameters and on the basis of the different experiments and curves will lead to the establishment of the dynamic power consumption model. The following section demonstrates how each parameter of expression (2) is obtained.

We evaluate first whether parameter β is independent of the filter versus expression of the parameter β. Hence, according to Figure 6, all given curves *LN(P)* versus *LN(I)* are almost parallels for a fixed coefficient value. This property was verified for all other orders and for all coefficient wordlength considered in this work. Consequently, the slope (β) of all these curves is the same regardless of the filter order. Thus, this slope (β) does not depend on the filter order and is only dependent on the coefficient wordlength.

To evaluate the expression of β as a function of coef (filter coefficient wordlength), we plot the curves given β depending on coef for fixed filter orders in Figure 7. These curves are obtained from data illustrated in Figure 6.

Figure 7 confirms the β filter order (N). Moreover, the curves show no linear evolution of β versus coefficient wordlength (coef). The expression of β as a function of coef is given in (3).

$$\beta = g(\text{coef}) = a \times \exp(-b \times \text{coef}) + c, \tag{3}$$

where a, b, and c are technology-dependent terms. The exponential term is explained by the curves in Figure 7. In fact, the curves converge around $(a + c)$ when the coefficient wordlength is close to 0 and then decreases very fast when the coefficient wordlength increases to converge to the c value.

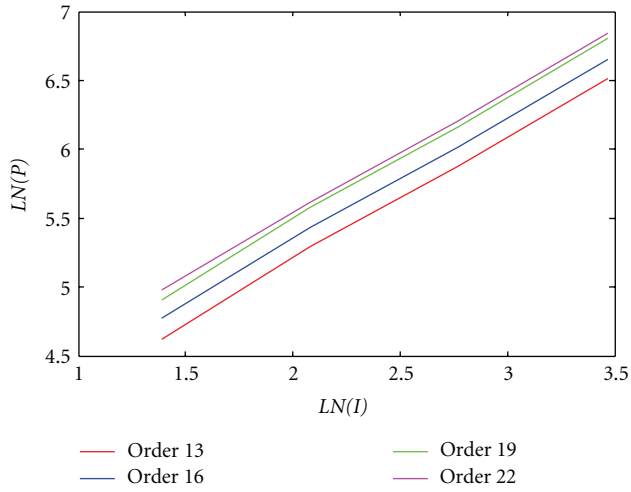


Figure 6: Natural logarithm of the power consumption of FIR filters in the direct form depending on the natural logarithm of the input wordlength for the 65 nm technology for a fixed coef size of 4 bits.

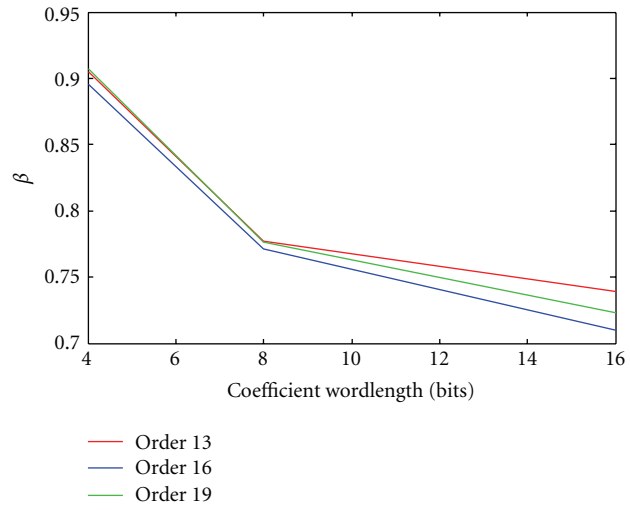


Figure 7: Evolution of β depending on coefficient wordlength for a fixed input size of 4 bits and fixed filter orders for the 65 nm technology.

The line of equation $y = c$ is a horizontal asymptote to the curves in Figure 7, which explain the additive c term in (3). To set numerical value of a, b, c for this given technology we used "*Matlab curve fitting toolbox*" [7] and we found that, in this technology, parameters a, b, and c are 0.6, 0.3, and -0.7, respectively. Of course, these values are useful only for the given technology; however, if we are using another technology the method presented above should be repeated at least one time as we have performed with the STM90 nm later.

To evaluate parameter α of (2), we used the same approach used for β expression extraction. Hence, first, the dependency of α versus filter order (N) was evaluated. Then,

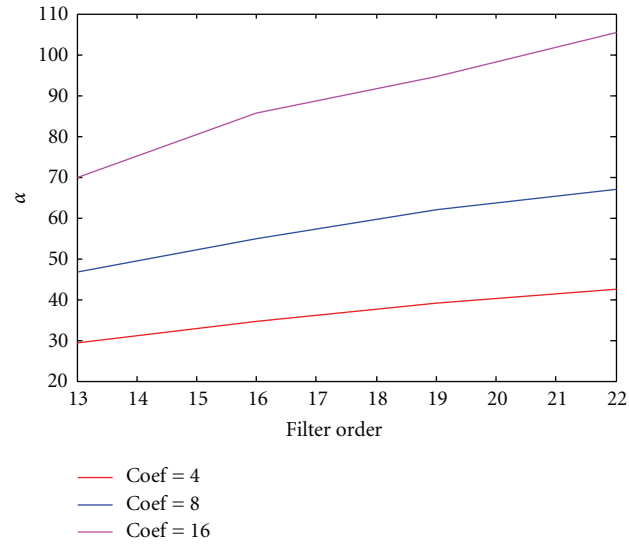


Figure 8: Evolution of α depending on filter order fixed coefficient wordlength for the 65 nm technology.

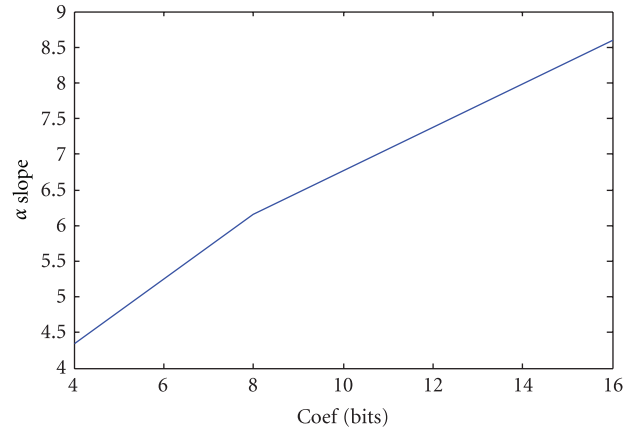


Figure 9: Evolution of α slope depending on the coefficient wordlength for fixed filter order for the 65 nm technology.

the dependency versus coefficient wordlength (coef) was found.

Curves in Figure 8 illustrate the dependency of parameter α on the filter orders and for three fixed coefficient wordlengths. To obtain these curves, we extracted the origin values from Figure 7. These origin values represent the normal logarithm of the parameter α for each fixed coefficient size. Then, we plotted the exponential of these values. The curves in Figure 8 show that α is almost linear versus filter order for any coefficient size. Hence, we next evaluated the dependency of the slope of α depending on coefficient wordlength, as shown in Figure 9. According to this figure, the slope of α is also linear versus coef. As a consequence, the relation (4) is deduced to model the evolution of α versus N and coef.

$$\alpha = N \times (\gamma \times \text{coef} + \omega), \tag{4}$$

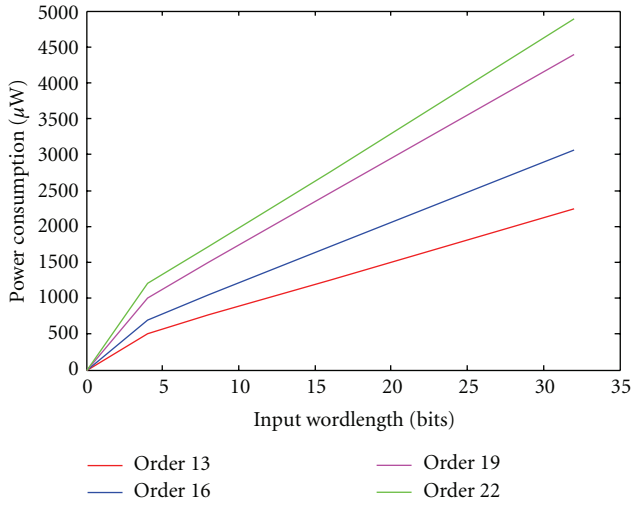


Figure 10: Relationship between the dynamic power consumption and input wordlength for the 90 nm technology for a fixed coefficient wordlength.

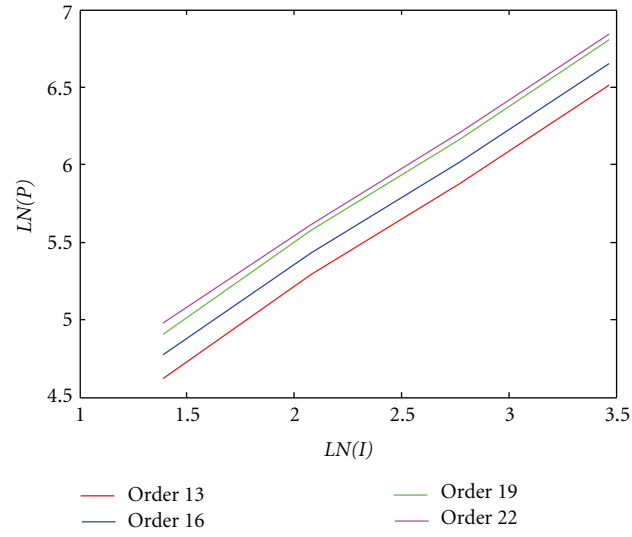


Figure 11: Relationship between the logarithm of dynamic power consumption and the logarithm of the input wordlength for the 90 nm technology for a fixed coefficient wordlength.

where γ and ω are constants and depend on the technology considered. Experiments show that in the 65 nm technology, the values of γ, ω are 0.2 and 1, respectively.

Hence, using (3) and (4), equation (5) gives the dynamic power consumption evolution versus filter parameters and normalized frequency for a direct form (Figure 2).

$$P(N, \text{coef}, I, f) = (N \times (\gamma \times \text{coef} + \omega)) \times \left(I^{(a \times \exp(-b \times \text{coef}) + c)}\right) \times \frac{f}{f_0}, \quad (5)$$

where f_0 is the frequency used during the syntheses process and is equal to 80 MHz. When the filter order is zero, the power consumption should be zero as well. This condition is guaranteed because power is directly proportional to filter order (N).

On the other hand, when the coefficient length (coef) is zero, the FIR filter is composed of N registers and $N/2$ adders. In this case, the power consumption becomes a constant multiplying the filter order (N).

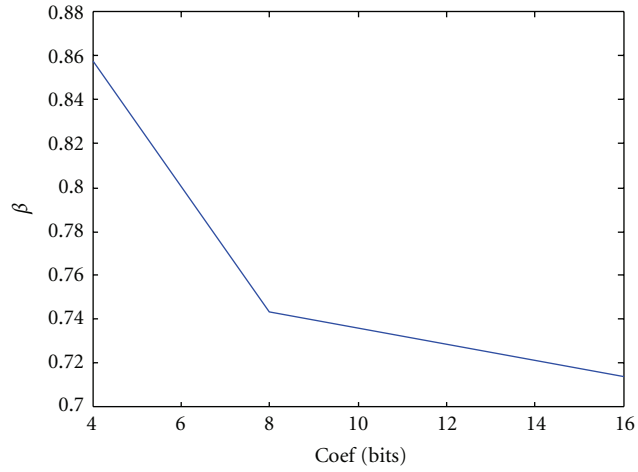


Figure 12: Evolution of β depending on the coefficient wordlength for fixed filter orders.

3.1.1. Verification of the Dynamic Power Consumption for STM90-nm. To verify whether the model of (4) is compliant with the amount of dynamic power consumption in the STM90 nm technology, we repeated all of the syntheses using the new library. The same parameter values were used for the experiments. The same conclusions regarding power consumption versus input wordlength were noticed. As shown in Figure 10, we also verified that the natural logarithm of the power is proportional to the natural logarithm of the input wordlength regardless of the filter order and coefficient size. Hence, (1) and (2) are still true for the 90 nm technology.

According to Figure 11, the independency of the parameter β in (2) regarding filter order N is still verified. Figure 12 shows the relationship between β and the coefficient wordlength. The same trends observed in Figure 7 are observed in Figure 13. Hence, the expression of β given in (3) is verified in the 90 nm technology.

Figures 13 and 14 give the evolution of parameter α (given in (3)) regarding filter order and the slope of α versus coefficient wordlength, respectively. According to the two figures, (4), which gives expression of α regarding filter order and coefficient wordlength, is verified.

As a consequence, the model of (5) is applicable for the evaluation of the dynamic power consumption in the 90 nm technology. With the same manner, to set numerical value of a, b, c for this given technology we used "*Matlab curve fitting toolbox*" [7] and we found that for the 90 nm process, parameters γ, ω, a, b, and c were equal to 0.5, 4.5, 0.5, 0.2, and 0.6, respectively.

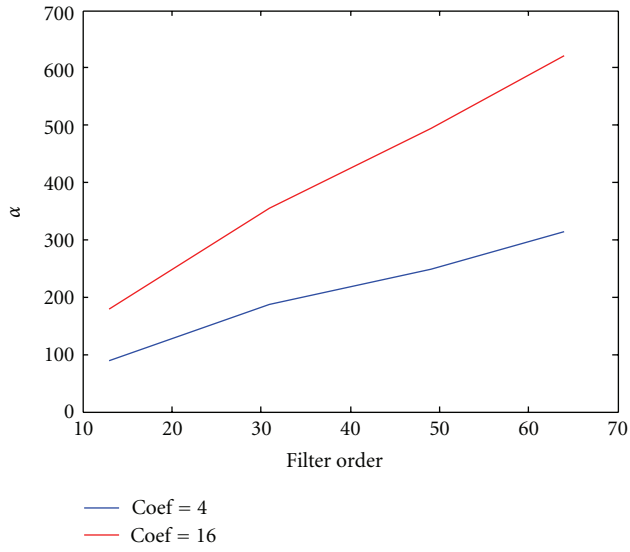


Figure 13: Evolution of α depending on filter orders for a fixed coefficient wordlength.

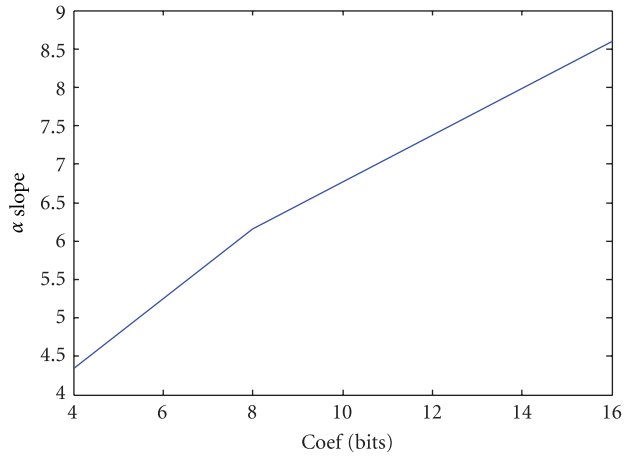


Figure 14: Evolution of the slope of α depending on the coefficient wordlength.

3.2. Static Power Consumption Model. The Table 1 illustrates the dynamic and static power contribution of the power consumption for the 90-nm process technology. It is clear from the table that static power cannot be neglected in this technology.

To establish a static power consumption model, we evaluated the evolution of the static power regarding filter parameters. In particular, we noticed a nonlinear relationship between the static power and input wordlength (see Figure 15). The second observation concerns the linearity of the natural logarithm of the static power versus the natural logarithm of the input wordlength (Figure 16). Hence, for the dynamic power consumption contribution, (1) and (2) could be used as general equation forms for the static power consumption of symmetric FIR filters.

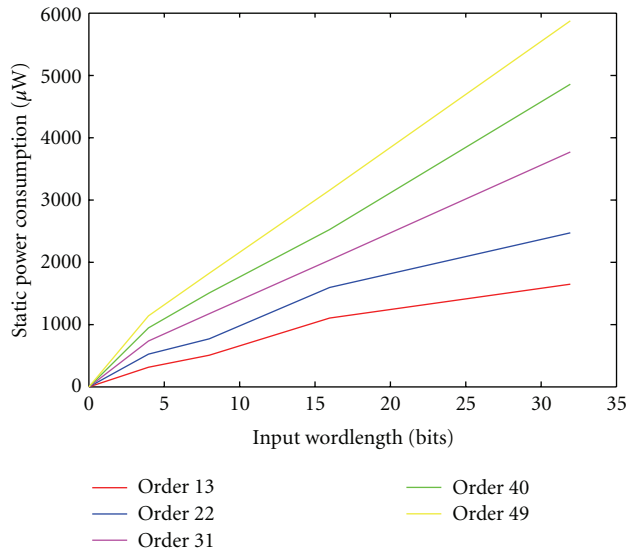


Figure 15: Relationship between the static power consumption and input wordlength for the 90 nm technology for a fixed coefficient wordlength.

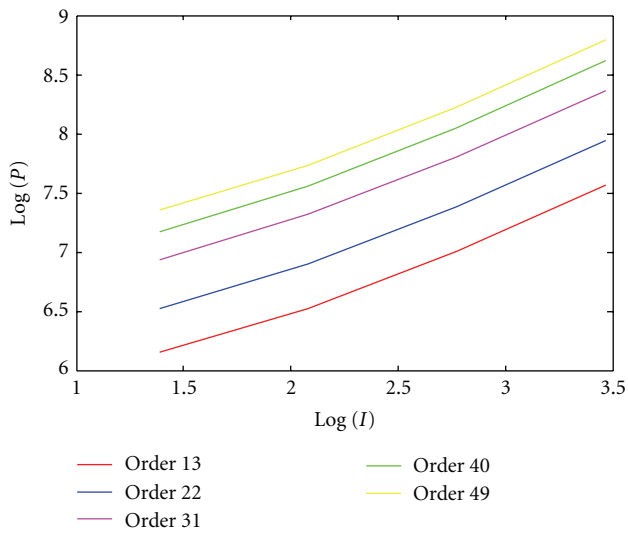


Figure 16: Relationship between the logarithm of the static power consumption and the logarithm of the input wordlength for the 90-nm technology for a fixed coefficient wordlength.

To establish expressions of the and β terms (in (2)), the dynamic power consumption modeling was followed. Hence, starting from the fact that β is filter order independent (since the curves are parallel in Figure 16), we plot the evolution of β versus coefficient wordlength in Figure 17.

After analyzing the curves in Figure 17, (6) fits the best curve evolution. Indeed, the exponential term is explained by the rapid decrease when small coefficient values are considered. The linear term is added because of the very slow increase when the coefficient wordlength increases.

$$\beta = \exp(-d \times \text{coef}) + e \times \text{coef} + g, \tag{6}$$

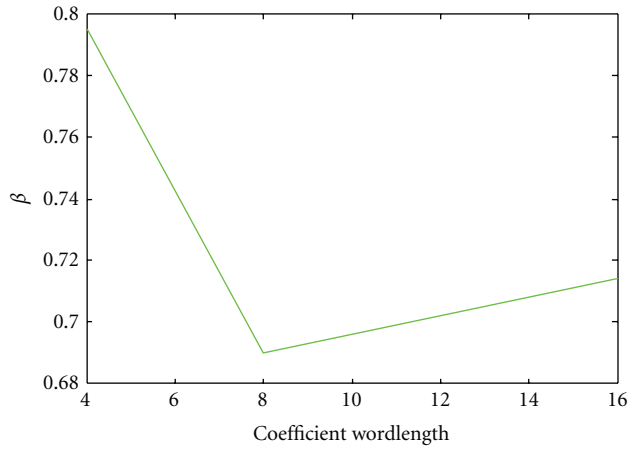


Figure 17: Evolution of β depending on the coefficient wordlength for fixed filter orders.

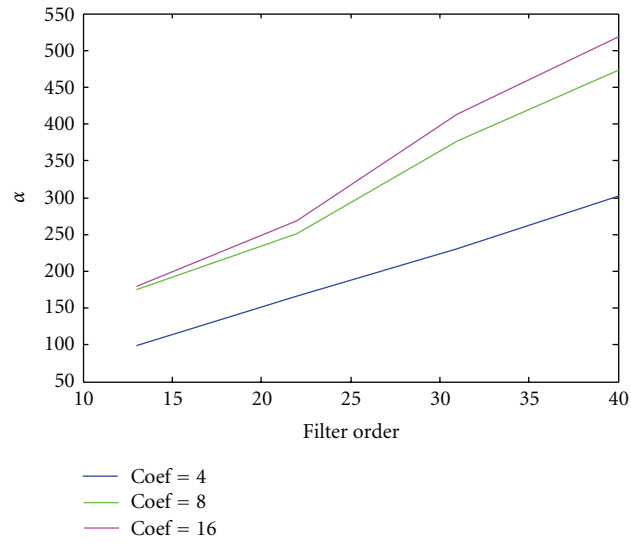


Figure 18: Evolution of α depending on filter orders for a fixed coefficient wordlength.

where d, e, and g are technology dependent parameters. Using Matlab, we shown that, when using a STMicroelectronics library, the parameters are equal to 0.11, 0.03, and 0.04.

Parameter α was then evaluated depending first on the filter order and then on the coefficient wordlength. Figure 18 shows the relationship between α and N. Hence, we found an almost linear relationship of this parameter versus filter order. In the second step, the slope of α was analyzed according to coefficient wordlength (see Figure 19). Matlab verified that (7) fits the evolution of the slope of α illustrated in Figure 18.

$$\text{slope} = \frac{\gamma}{(a \times \exp(-b \times \text{coef}) + c)}, \tag{7}$$

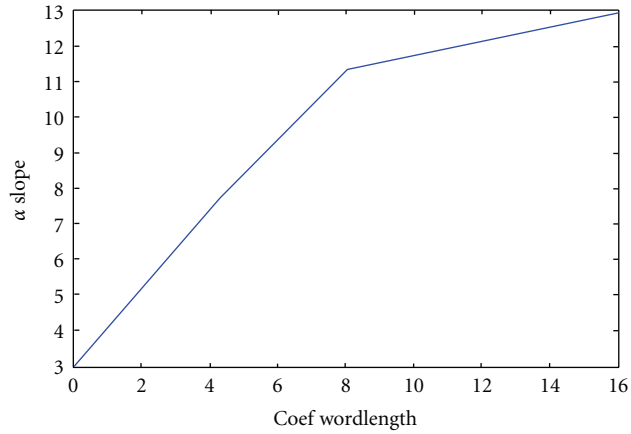


Figure 19: Evolution of the α slope depending on filter orders for a fixed coefficient wordlength.

where a, b, c, and γ are parameters depending on the technology. According to Matlab, when using the 90 nm STMicroelectronics library, the parameters are equal to 0.29, 0.4, 0.077, and 1, respectively. Finally, α is expressed in (8):

$$\alpha = \omega' + \left(\frac{\gamma}{(a \times \exp(-b \times \text{coef}) + c)}\right) \times N. \tag{8}$$

As a consequence, the general expression of the static power consumption for a direct form FIR filter (Figure 2) in the 90 nm technology could be written as shown in (9).

$$P_{\text{stat}}(N, \text{coef}, I) = N \times \left(\frac{\gamma}{a \times \exp(-b \times \text{coef}) + c} + \omega\right) \times I^{(\exp(-d \times \text{coef}) + e \times \text{coef} + g)}, \tag{9}$$

where technology dependent parameters a, b, c, d, e, and g are equal to 0.29, 0.4, 0.077, 0.11, 0.03, and 0.04, respectively. Parameters γ and ω depend also on the technology and are equal to 1.2 and -1.2, respectively, for the current model.

4. Validation of the Power Consumption Models

This part presents the validation of the models obtained. Hence, we present different figures comparing the syntheses values of power consumption and the results deduced from established models. The aim of the comparison is to demonstrate first that the power consumption trends are respected by the models. In fact, the power estimation value for a given design is not really important and the objective is to prove that decisions concerning the choice of the suitable parameters, which reduces the power consumption, are not modified using the models. For this purpose, a parameter called the "deviation" is calculated. This parameter gives the difference for two fixed filter parameters (which are the filter order and coefficient wordlength or input wordlength) between the power consumption value when varying the third filter parameter (which is the coefficient wordlength or input wordlength). Indeed, having a similar "deviation"

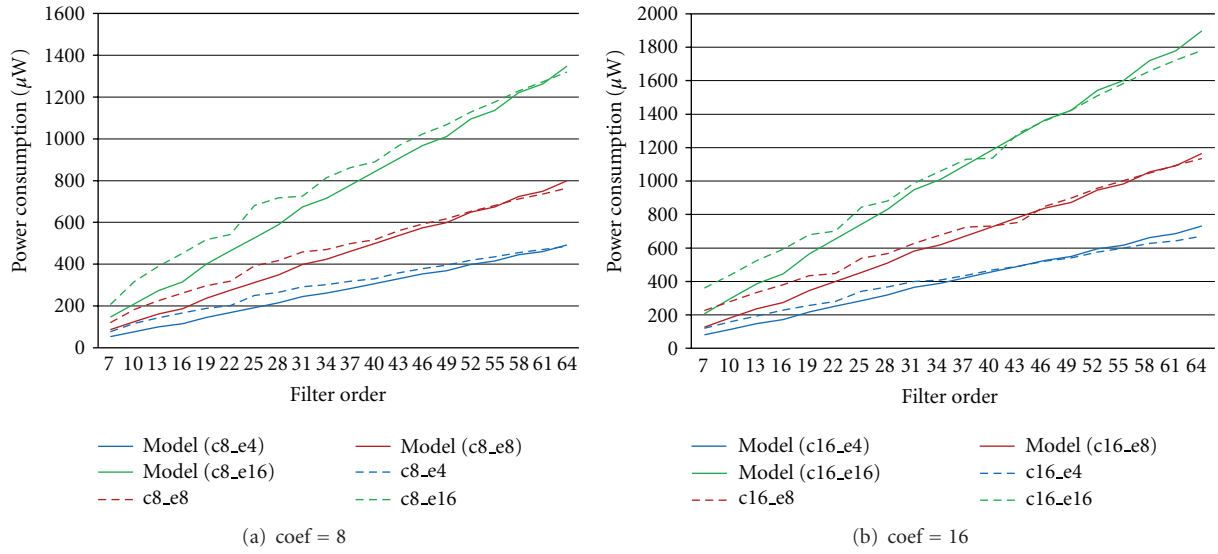


Figure 20: Comparison of model power results with experimental values for the 65 nm technology for a fixed coefficient wordlength.

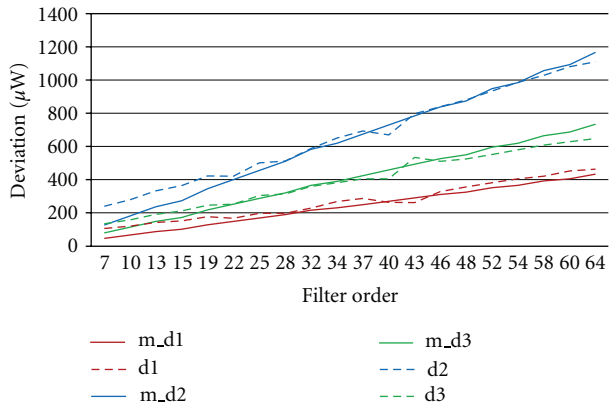


Figure 21: "Deviation" parameter for model and estimation tool power results for coef = 16.

for syntheses and model values means that models cannot change decisions concerning the best parameters to use for power consumption reduction.

4.1. Validation of Power Consumption Models in the 65 nm Process Technology. Figures 20(a) and 20(b) give the results of the comparison of power consumption for fixed coefficient wordlengths (8 bits and 16 bits, resp.) and variable input wordlength and filter order.

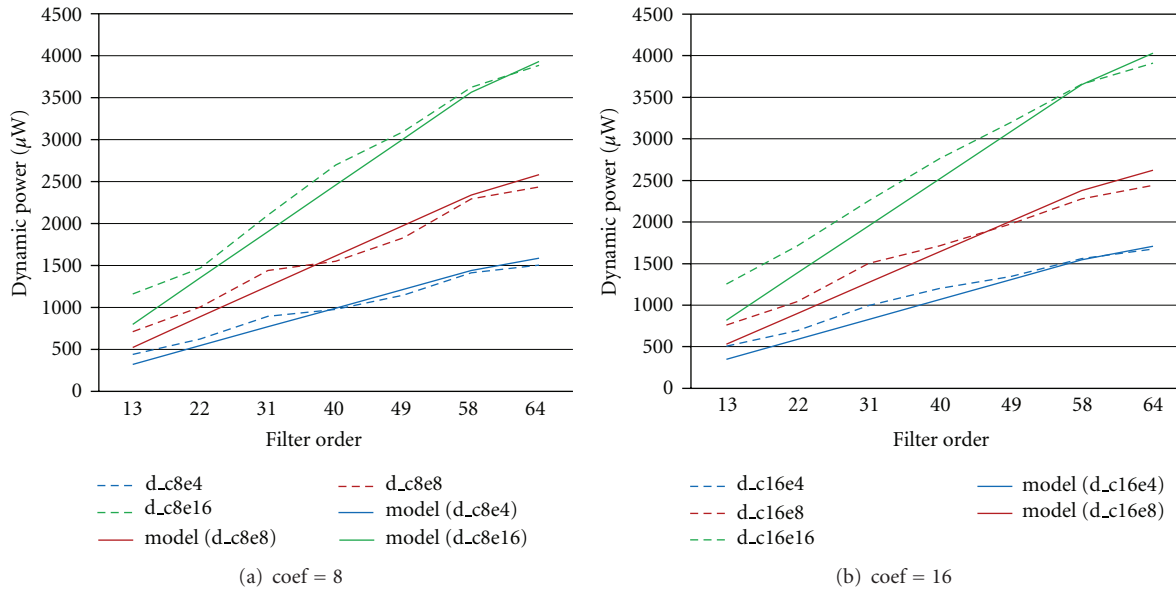
According to Figure 20, the power consumption trends are respected by the model. Moreover, for a fixed filter order and the deviation of the power consumption when varying the input size, it can be confirmed that deviations measured using the model and power estimation tool are very close. Figure 21 shows the deviations observed in Figure 20(b). In this figure, d1 corresponds to the difference in power consumption considering an input size equal to 8 bits and an input size equal to 4 bits according to experimental results. The model d1 parameter is the same information according to the given model. In the same way, d2 is the power difference considering an input size equal to 16 bits and an input size equal to 4 bits according to experimental results, and d3 is the power difference considering an input size equal to 16 bits and an input size equal to 8 bits. The model d2 and model d3 parameters are the same difference using the model.

In the same way, it can be confirmed that the deviations measured for a fixed filter order and a fixed coefficient size when varying the input wordlength are also close to the experimental deviations measured by the power estimation tool. Hence, we can conclude that, the model could be used for any filter order, input wordlength, or coefficient size not tested within experiments. In conclusion, the model obtained could help efficiently estimate the power consumption of a direct form symmetric FIR filter.

4.2. Validation of the Dynamic Power Consumption Model for the 90 nm Process Technology. In this part, we validate the dynamic power model given in (5). For this purpose, we ran several syntheses using different filter parameters. As in the case of the 65 nm technology, we used the design compiler of Synopsys for the syntheses and the "Primepower" tool for power estimation. The frequency was fixed to 80 MHz for the syntheses.

Figure 22 gives a comparison of the results extracted from the model and the experimental values for dynamic power consumption and for fixed values of the coefficient wordlength.

The same conclusions for the 65 nm technology could be formulated. In fact, if we analyze each curve in Figure 22, we can confirm that using a model does not modify the trends of dynamic power consumption of an FIR filter depending on its parameters. Moreover, for a fixed filter order, the dynamic power deviation due to the input wordlength increase is



(a) coef = 8 (b) coef = 16

Figure 22: Comparison of model dynamic power results to experimental values for the 90 nm technology for a fixed coefficient wordlength.

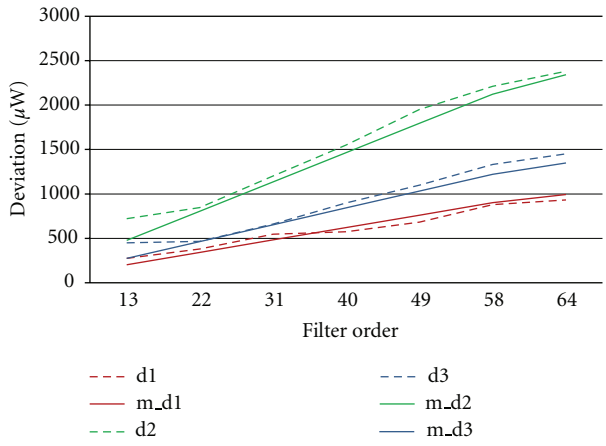


Figure 23: "Deviation" parameter for model and estimation tool power results for coef = 8.

almost equal to the deviation measured for the experimental dynamic power estimation (Figure 23).

4.3. Validation of the Static Power Consumption Model for the 90 nm Process Technology. For the dynamic contribution, we validated the static power model as given in (9). Figure 24 shows the comparison between the results extracted from the model and the experimental values for static power consumption and for fixed input wordlength.

In the same way, we confirm that the proposed model for static power consumption gives static power values very close to the values given by the estimation tool. The model also does not modify trends of power and does not modify decisions depending on the filter parameters because the deviations are almost equal (see Figure 25).

5. Power Consumption Estimation Tips for Polyphase Implementation Form of FIR Filters

5.1. Results of Polyphase Implementation Form in the 65 nm Technology. Polyphase implementations of FIR filters were performed with different input wordlengths, coefficient sizes, and filter orders, which lead to the same observations illustrated in Figure 26. This figure compares the direct form implementation of different FIR filter orders versus their polyphase decomposition.

The comparison is made for an input size equal to 4 bits and includes decimation values of 2, 4, 8, and 16. According to Figure 26, it is clear that polyphase decomposition reduces the power consumption for any decimation factor.

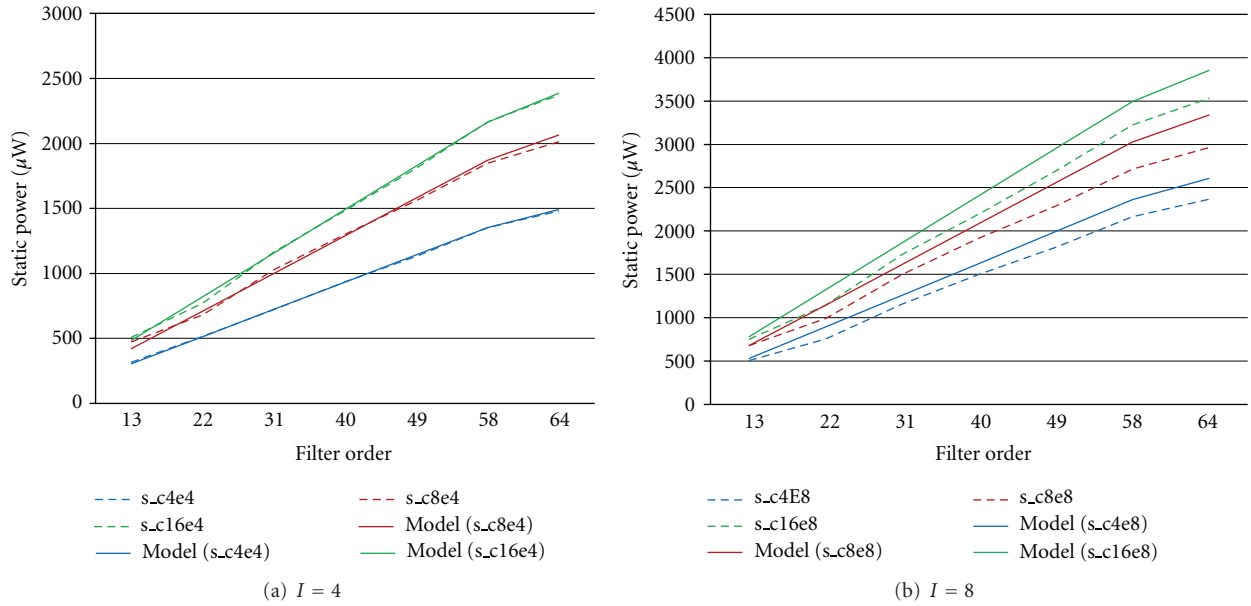
In fact, the following observations are clear.

(i) Decomposition into two stages allows power reduction rates that increase with filter order. The reduction rates are between 20% and 40% in comparison with the direct form implementation.

(ii) Decomposition into four stages allows reduction rates between 45% and 60% compared with the direct form implementation.

(iii) Decomposition into 8, 16, and 32 stages allows reduction rates between 60% and 70% for 8 stages based architecture, 60% and 75% for the 16 stages, and 55% and 75% for the 32 stages.

As a consequence, polyphase decomposition is always beneficial in terms of power consumption reduction. However, the reduction average depends on the decimation factor that is used for a given filter order. Equation (10) gives the relation between the decimation factor and filter order up to 128 to



(a) $I = 4$ (b) $I = 8$

FIGURE 24: Comparison of the model static power results with experimental values for the 90 nm technology for a fixed Input wordlength.

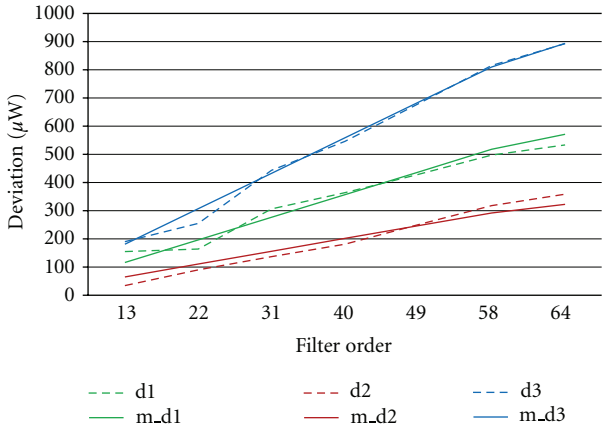


FIGURE 25: "Deviation" parameter for model and estimation tool power results for an input size of 8 bits.

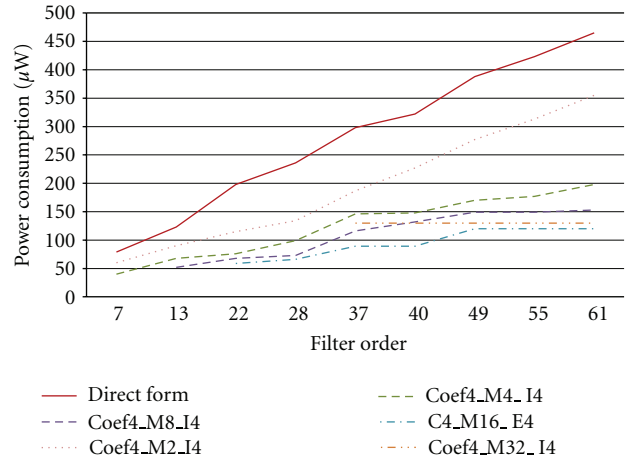


FIGURE 26: Comparison between direct form and polyphase form power consumption for different decimation factor values.

offer the best power consumption reduction. It is estimated that order 128 is sufficiently large for channel selection filters.

$$M_{\text{opt}} = \begin{cases} 8 \text{ si } 8 < \text{filter order} \leq 16, \\ 16 \text{ si } 16 < \text{filter order} \leq 64, \\ 32 \text{ si } 64 < \text{filter order} \leq 128. \end{cases} \quad (10)$$

5.2. Results of Polyphase Implementation Form in the 90-nm Technology. To verify the dynamic power reduction rates measured in the 65 nm technology when performing polyphase decomposition of a symmetric FIR filter, some syntheses were run in the 90 nm technology. Figure 24 illustrates the impact of each decimation factor on the dynamic power consumption for a fixed input size (4 bits) and fixed coefficient size (4 bits). It was verified that trends for other parameters are the same as those represented in Figure 27. A comparison to the direct form confirms the benefit of the polyphase decomposition.

The same conclusions extracted in the 65 nm technology can be formulated for dynamic power consumption in the 90 nm technology. However, the static power in this technology could not be neglected. Hence, the comparison should also concern the static power contribution (see Figure 28). According to Figure 28, static power and dynamic power consumption have the opposite trends. In fact, the static power consumption increases with the increase of the decimation factor. Indeed, when applying polyphase decomposition, the occupied area increases, which increase the static power consumption.

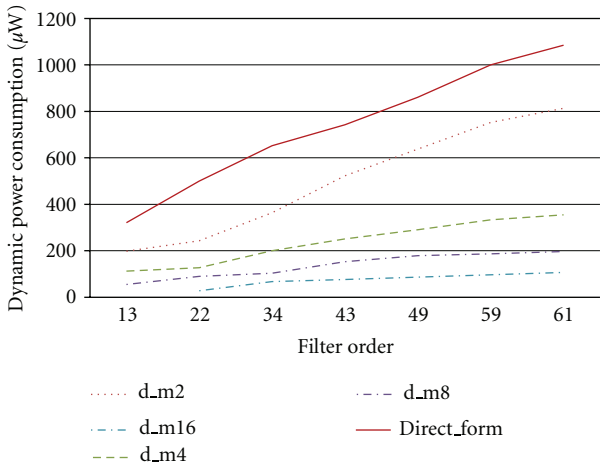


Figure 27: Dynamic power evolution of poly phase form implementations of a symmetric FIR filter for the 90 nm technology.

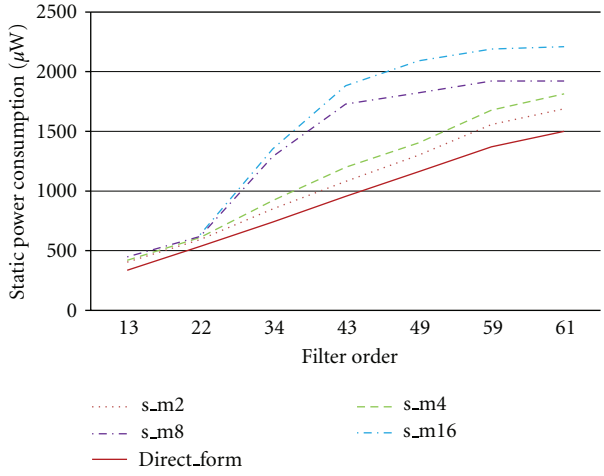


Figure 28: Static power evolution of poly phase form implementations of a symmetric FIR filter for the 90 nm technology.

In Figure 29, the total power consumption results for the FIR filter considered are plotted and compared to the power consumption results of the direct form. According to this figure and considering the total power consumption criteria, the polyphase implementation could lead in some cases to a power consumption increase in comparison with the direct form power consumption results.

However, decimation by a factor of 2 and 4 always reduces the total power consumption in comparison with the direct form, with advantageous reduction for a decimation factor of 4.

6. Comparison of Power Consumption of Filtering Architectures for GSM and UMTS Standards Using Established Models.

In this section, we examine the power consumption estimation of different decimation filter solutions for multistandard receiver supporting UMTS and GSM standards. The aim

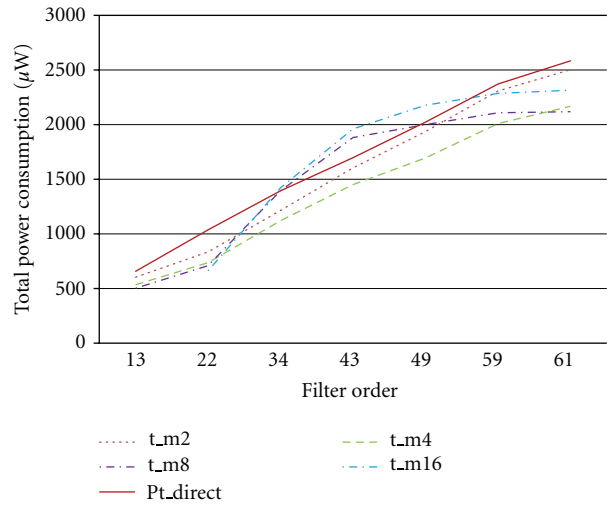


Figure 29: Total power evolution of poly phase form implementations of a symmetric FIR filter for the 90 nm technology.

of this section is to demonstrate how the models could help filter designer make the correct choice regarding filter architecture to reduce the power consumption.

For this purpose, three different architectures suitable for multistandard receivers are analyzed (Figure 30).

The parameters of the different filters building each architecture are obtained using the specification methodology described in [8]. The multistandard receiver architecture and parameters considered in this work are calculated in [9]. Because filtering performances are out of the scope of this paper, only the description of the filter parameters is given.

6.1. Filtering Solution Description for UMTS and GSM Standards. This paragraph gives the details of the different architectures for both standards. Tables 3 and 4 summarize the parameters of the different stages in the case of UMTS and GSM standards, respectively.

6.1.1. 2-Stage Architecture (Arch0). For both UMTS and GSM standards, the first filter is composed of a cascade of 5-stage CIC filters. It performs decimation by a factor of 12 or 4 for GSM and UMTS standards, respectively. The recursive architecture of the CIC filter allows the programmability of the filter depending on the selected standard. The input wordlength of the filter is equal to 6 bits at its output; the wordlength is 16 bits for the UMTS standard and is equal to 26 bits in the case of the GSM standard. The second filter of the 2-stage architecture is a symmetric FIR filter of order 45 for UMTS, where 12 bits are necessary for the quantification of the filter coefficients for this standard. In the case of GSM, the required filter order is 83, and 12 bits are sufficient for the quantization of coefficients.

6.1.2. 3-Stage Architecture (Arch1). In this architecture, the last filter of the 2-stage architecture is split into a half-band filter and an FIR filter performing, each one a decimation by a factor of 2. The half-band filter has an order equal

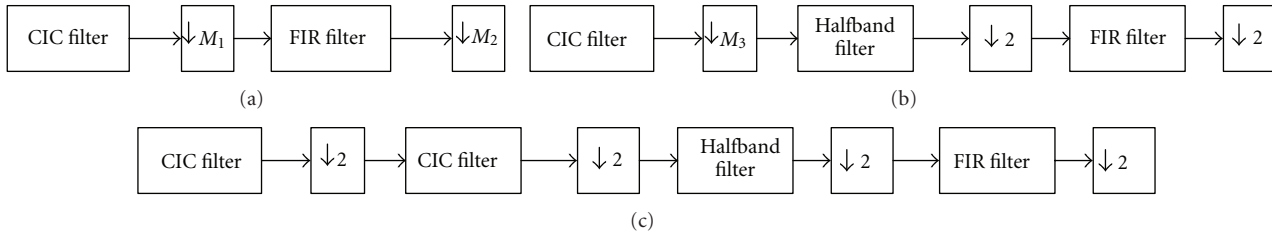


Figure 30: Decimation filter architectures for GSM and UMTS standards.

Table 3: Specification of the filters in proposed architectures for the GSM standard.

Architecture Specification	CIC filter	HB1filter	HB2 filter	FIR filter
2 stages				
Order	5	*N/A*	*N/A*	83
Input wordlength	6	*N/A*	*N/A*	17
Coefficients wordlength	*N/A*	*N/A*	*N/A*	12
Decimation factor	12	*N/A*	*N/A*	4
3 stages				
Order	5	20	*N/A*	45
Input word-length	6	17	*N/A*	17
Coefficients wordlength	*N/A*	11	*N/A*	11
Decimation factor	12	2	*N/A*	2
4 stages				
Order	6	8	10	34
Input wordlength	6	17	17	17
Coefficients wordlength	*N/A*	10	11	10
Decimation factor	6	2	2	2

Table 4: Specification of the filters in proposed architectures for the UMTS standard.

Architecture Specification	CIC filter	CIC filter	HB filter	FIR filter
2 stages				
Order	5	*N/A*	*N/A*	57
Input wordlength	6	*N/A*	*N/A*	15
Coefficients wordlength	*N/A*	*N/A*	*N/A*	12
Decimati on factor	4	*N/A*	*N/A*	4
3 stages				
Order	5	*N/A*	14	20
Input wordlength	6	*N/A*	15	15
Coefficients wordlength	*N/A*	*N/A*	10	12
Decimation factor	4	*N/A*	2	2
4 stages				
Order	5	4	10	20
Input wordlength	6	11	15	15
Coefficients wordlength	*N/A*	*N/A*	9	12
Decimation factor	2	2	2	2

to 14 for UMTS and 20 for GSM. The filters coefficients length is fixed to 10 bits and 11 bits for UMTS and GSM standards, respectively. The last filter has an order of 20 for UMTS and an order equal to 45 for GSM. For both standards, coefficients are quantized with 12 bits.

6.1.3. 4-Stage Architecture (Arch2). In the case of the GSM standard, the 4-stage architecture is based on a CIC filter of order 6 in its recursive form, followed by a cascade of 2 half-band filters, each one performing a decimation by a factor of 2. The last stage is composed of a symmetric FIR filter,

Table 5: Power consumption evaluation according to models in the 65 nm low-power technology.

		2-stage Arch power (μW)	3-stage Arch power (μW)	4-stage Arch power (μW)
GSM standard	Filter 1 (cic filter)	226	226	264,3
	Filter 2 (halfband)	*N/A*	42,5	32,6
	Filter 3 (halfband)	*N/A*	*N/A*	20,4
	Filter 4 (FIR)	208,6	47,8	34,7
	Total GSM	434,6	336,3	352
UMTS standard	Filter 1 (cic filter)	*N/A*	*N/A*	*N/A*
	Filter 2 (cic filter)	*N/A*	*N/A*	*N/A*
	Filter 3 (halfband)	*N/A*	54	52
	Filter 4 (FIR)	303	56	56
	Total UMTS	303	110	108

Table 6: Power consumption evaluation according to models for the 90 nm technology.

		2-stage Archpower (μW)	3-stage Archpower (μW)	4-stage Archpower (μW)
GSM standard	Filter 1 (cic filter)	967,8	967,8	2796,8
	Filter 2 (halfband)	*N/A*	119,5 + 1818,8	93,7 + 720
	Filter 3 (halfband)	*N/A*	*N/A*	59,7 + 909,4
	Filter 4 (FIR)	575,3 + 8660	134,4 + 4092,4	99,5 + 3060,75
	Total GSM	10203,1	7132	7739,8
UMTS standard	Filter 1 (cic filter)	*N/A*	*N/A*	*N/A*
	Filter 2 (cic filter)	*N/A*	*N/A*	*N/A*
	Filter 3 (halfband)	*N/A*	145 + 827	142 + 818
	Filter 4 (FIR)	823 + 4000	147 + 1600	147 + 1600
	Total UMTS	4823	2719	2707

completing the decimation by 2 and the channel selection. The two half-band filters have orders equal to 6 and 10, and their coefficients are quantized on 10 and 11 bits, respectively. The last FIR filter has an order equal to 34, and we found that 10 bits are required for the quantization of coefficients.

For the UMTS standard, the CIC filter used in the 3-stage architecture is split into a cascade of 2 CIC filters. The number of required stages is 5 for the first CIC filter and 4 for the second. Each filter performs decimation by a factor of 2. The output size of the first CIC filter is equal to 11 bits. The third filter is a half-band filter of order 10 and has a coefficient size equal to 9 bits. Finally, a symmetric FIR filter of order 20 with coefficients quantized on 12 bits completes the selection. As explained before, the input wordlength is considered equal to 15 bits, unless for the first or second stage, for which the input wordlength is equal to 6 bits or 11 bits, respectively.

6.2. Comparison to Implementation Results. On the basis of the filter parameters and the established models given in (5) and (10), we estimated the power consumption of each sub-block in the filtering architectures for both the 65 nm and 90 nm technologies. Table 5 gives power consumption estimation results for GSM and UMTS standards for the 65 nm technology. The results for both standards for the 90 nm technology are given in Table 6.

For the particular case of CIC filters, the power consumption comparison is performed following the work in [2]. Indeed, the authors in [2] studied the power consumption of fixed order CIC filters depending on their implementation architecture. In the case of the GSM standard and because the power consumption of CIC filters of different orders is not included in the work in [2], the power consumption results concern the FIR implementation form of the CIC filters.

According to Tables 3 and 4, it is clear that, for both technologies (65 nm and 90 nm), the architecture that presents the optimal power consumption in the case of the GSM standard is the 3-stage architecture.

For the UMTS standard, the power consumption values obtained from models are very similar in the case of 3-stage

Table 7: Power consumption evaluation according to the "prime-power" tool for the 65-nm technology for UMTS and GSM standards.

	2 stages	3 stages	4 stages
GSM standard	280,6	231	260
UMTS standard	171	157	149

Table 8: Power consumption evaluation according to the "prime-power" tool for the 90 nm technology for UMTS and GSM standards.

		2 stages	3 stages	4 stages
90 nm	GSM	751	640	731
	UMTS	5353	4231	4441
65 nm	GSM	280,6	231	260
	UMTS	171	157	149

and 4-stage architectures. If the estimation of the power is considered, which is done in [2], for the comparison of the power consumption of the CIC filters, the power values increase, but the trends are not modified.

It is, however, important to notice that the power due to the clock tree and stage connection is not considered in the evaluation. Thus, the power consumption for 4 stages should be considered because it is composed of more stages and should increase compared to the 3-stage architectures, in particular for the 90 nm technology. Hence, it can be concluded that the 3-stage architecture is also more advantageous in terms of power consumption for the UMTS standard.

To validate these results experimentally, VHDL code of all architectures considered was built. The results of the power estimated after logic syntheses are given in Tables 7 and 8 for the 65 nm and 90 nm technologies, respectively. These tables confirm the conclusions obtained from established models.

7. Conclusion

This paper presented power consumption evaluation models of direct form FIR filters (Figure 2) used for decimation. Both models for dynamic and static power contributions are proposed in this work in STM65 nm low-power and STM90 nm ASIC technology. The proposed models are high level models, which estimate the dynamic and static power consumption of the FIR filter depending on three filter parameters, which are the input wordlength, coefficient wordlength, and filter order. The aim of these models is to help the system designer compare, at the system level, different filter architectures in terms of power consumption without having to implement the different filters and perform syntheses. However, this method should be verified for any new library different form the used ones in this work. In the second step, the effect of polyphase decomposition for FIR decimator filters was evaluated in two different technologies. We found that polyphase decomposition allows good dynamic power reduction regarding the direct form implementation. We observed that the dynamic power reduction could reach 75% in some cases. To help the system designer choose the best decimation factor in terms of power consumption, relation (3) was given between the decimation factor and the filter order. However, the static power consumption increases when performing polyphase decomposition. When the static contribution is important (as with the 90 nm technology library considered), the total power consumption can increase in some cases in comparison with the direct form power consumption.

Finally, a case study concerning the UMTS and GSM standards was presented. We performed a comparison between three filter architectures. The power estimation based on proposed models helped choose the suitable architecture for power consumption optimization, and the result was confirmed by filter implementations.

References

[1] Y. S. Poberezhskiy and G. Y. Poberezhskiy, "Flexible analog front ends of reconfigurable radios based on sampling and reconstruction with internal filtering," *Eurasip Journal on Wireless Communications and Networking*, vol. 2005, no. 3, pp. 364–381, 2005.

[2] Y. Dumonteix, H. Aboushady, H. Mehrez, and M. Louerat, "Low power comb decimation filter using polyphase decomposition for mono-bit analog-to-digital converters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 10, pp. 898–903, 2001.

[3] M. O. Lakshmanan and M. Ali, "High performance parallel multiplier using Wallace-Booth algorithm," in *Proceedings of the IEEE International Conference on Semiconductor Electronics*, pp. 433–436, 2002.

[4] H. Lee, "A power-aware scalable pipelined booth multiplier," in *Proceedings of the IEEE International Systems-On-Chip Conference*, pp. 123–126, September 2004.

[5] Universal Mobile Telecommunication System (UMTS), UE Radio Transmission and Reception (FDD), 3GPP TS.101 version 5.2.0 Release 5. ETSI, 2002.

[6] Radio Transmission and Reception GSM 05.05. ETSI, 1996.

[7] Matlab curve fitting toolbox data sheet, http://www.mathworks.com/products/curvefitting/.

[8] R. Barrak, A. Ghazel, and F. Ghannouchi, "Design of sampling-based downconversion stage for multistandard RF subsampling receiver," in *Proceedings of the 13th IEEE International Conference on Electronics, Circuits and Systems (ICECS '06)*, pp. 577–580, December 2006.

[9] A. Ghazel, L. Naviner, and K. Grati, "On design and implementation of a decimation filter for multistandard wireless transceivers," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 558–562, 2002.

Hindawi Publishing Corporation
VLSI Design
Volume 2012, Article ID 716476, 13 pages
doi:10.1155/2012/716476

Review Article

Cognitive Radio RF: Overview and Challenges

Van Tam Nguyen,[1] Frederic Villain,[2] and Yann Le Guillou[3]

[1] *Institut Mines-Telecom, Telecom ParisTech, LTCI CNRS UMR 5141, 75634 Paris, France*
[2] *BL TV Front End, NXP, 14460 Colombelles, France*
[3] *Renesas Mobile Corporation, 35517 Cesson Sevigne Cedex, France*

Correspondence should be addressed to Van Tam Nguyen, vtnguyen@telecom-paristech.fr

Received 4 November 2011; Revised 6 February 2012; Accepted 7 February 2012

Academic Editor: Christophe Moy

Cognitive radio system (CRS) is a radio system which is aware of its operational and geographical environment, established policies, and its internal state. It is able to dynamically and autonomously adapt its operational parameters and protocols and to learn from its previous experience. Based on software-defined radio (SDR), CRS provides additional flexibility and offers improved efficiency to overall spectrum use. CRS is a disruptive technology targeting very high spectral efficiency. This paper presents an overview and challenges of CRS with focus on radio frequency (RF) section. We summarize the status of the related regulation and standardization activities which are very important for the success of any emerging technology. We point out some key research challenges, especially implementation challenges of cognitive radio (CR). A particular focus is on RF front-end, transceiver, and analog-to-digital and digital-to-analog interfaces which are still a key bottleneck in CRS development.

1. Introduction

The information and communication technology industry is today faced with a global challenge: develop new services with improved quality of service (QoS) and at the same time reduce its environmental impact. Clearly, there is a deep need of global efficiency not only in the energy domain, but also in the spectral domain.

Indeed, the gap between wireless supply and demands widens. It is feared that an imminent spectrum crisis, in which exploding demand from smart phones will soon overwhelm the wireless capacity, will occur. The problem is the lack of new spectrum available to wireless data carriers. Smartphone data traffic is growing so fast that if nothing is done they will use up the available spectrum. This just affects not only smart phones but also all wireless devices. A huge amount of spectrum is required for broadband use in the future as suggested in National Broadband Plan (NBP) in USA and the digital agenda in Europe.

In order to address the problem of spectrum usage efficiency, the cognitive radio (CR) concept was proposed [1, 2]. The detailed definition of cognitive radio systems (CRSs) will be given in Section 2. Cognitive radio technology has the potential of being a disruptive force within spectrum management.

A very popular example is opportunistic radio or opportunistic spectrum access whose principle is temporal, spatial, and geographic "reuse" of licensed spectrum as shown in Figure 1 where an "unlicensed" secondary user (SU) can be permitted to use licensed spectrum, provided that it does not interfere with any primary users (PUs). In that way, the efficiency of spectrum usage is significantly improved. Various measurements of spectrum utilization have shown that spectrum is underutilized, in the sense that the typical duty cycle of spectrum usage at a fixed frequency and at a random geographical location is low. This means that there are many "holes" in the radio spectrum that could be exploited [3]. Opportunistic radio system should be able to exploit these spectrum holes by detecting them and using them in an opportunistic manner. Because of the outstanding propagation characteristic in the television (TV) bands with strong wall and floor penetration capability, long

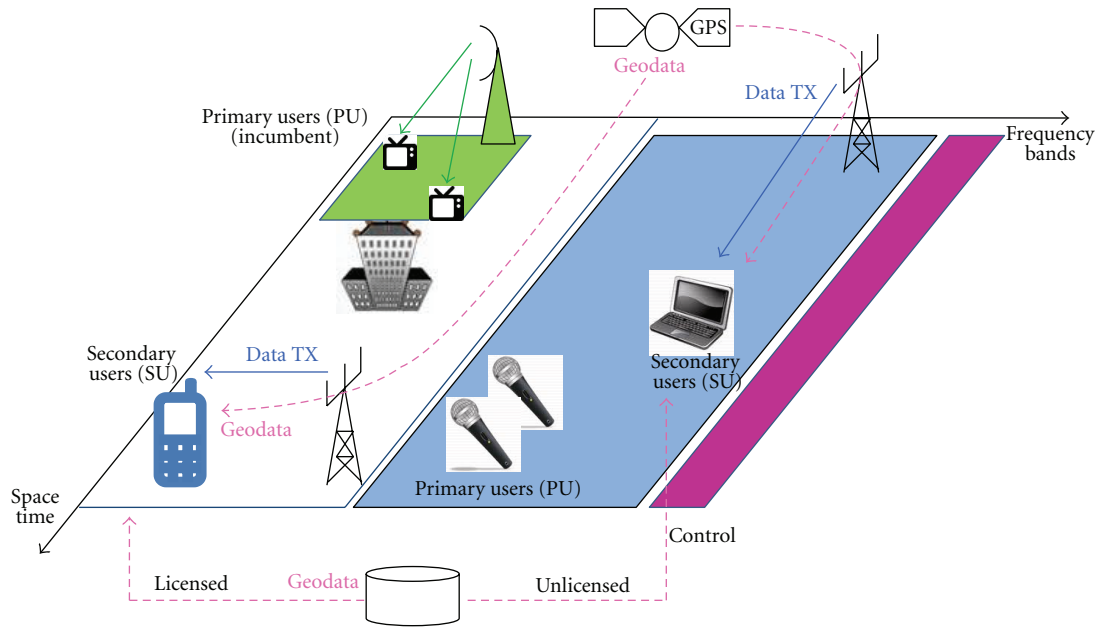


Figure 1: Opportunistic spectrum access overview.

range, and flexible bandwidth, it could be used to allow a brand-new class of services and increase the limited capacity of existing systems.

But CRS is not only limited to opportunistic spectrum access. Also, it includes heterogeneous networks where a heterogeneous radio framework management is performed. Current spectrum allocation approaches, fixed by nature, do not allow for the allocation of frequency bands to different radio access technologies (RATs) dynamically. However, the coexistence and cooperation of diverse technologies, which form part of a heterogeneous infrastructure, have brought about the possibility of flexibly managing the spectrum in a dynamic manner. No longer are fixed frequency bands guaranteed to apply to specific RATs, but conversely, through intelligent management mechanisms, bands can be allocated to RATs dynamically in a way such that the capacity of each RAT is maximized and interference is minimized. on long terms, will be considered also the flexibility in spectrum management where network operator may employ different RATs dynamically over time/frequency/location and acquire or exchange the spectrum usage rights. The devices may autonomously and dynamically adapt to the diverse heterogeneous radio networks [4].

This new technology opens up many potential applications and exciting opportunities [5]. For example, the rural connectivity, content distribution networks, city and campus wide coverage, and giant wireless hotspots could benefit from the new spectrum access and management.

This paper is organized as follows: after this introduction, the definition and high level concept of CRS is presented in Section 2. In Section 3, the status on regulation and standardization activities are described. Section 4 will present the research challenged related to CRS. Section 5 will describe the implementation challenges of CR devices. The conclusion of the paper is drawn in Section 6.

2. Definition and High Level Concept

There are different definitions of CRS, from many authors and organizations. The definition giving the common understanding about CRS and now adopted for most is from International Telecommunication Union (ITU) [6]. CRS is a radio system employing technology that allows the system:

(i) to obtain knowledge of its operational and geographical environment, established policies, and its internal state (*cognitive capability*);

(ii) to dynamically and autonomously adjust its operational parameters and protocols according to its obtained knowledge in order to achieve predefined objectives (*reconfigurable capability*);

(iii) to learn from the results obtained (*learning capability*).

At high level concept presented in Figure 2, the main components of the CRS are the intelligent management system and reconfigurable radios [7, 8]. CRS is also able to take action including obtaining knowledge, making decision, reconfiguration, and learning. The knowledge used by the CRS includes knowledge about operational radio and geographical environment, internal state, established policies, usage patterns, and users' needs.

The methods to obtain this knowledge include getting information from component of the CRS, spectrum sensing, access to the cognitive pilot channel (CPC), geolocation, and database access. Using the obtained knowledge, the CRS

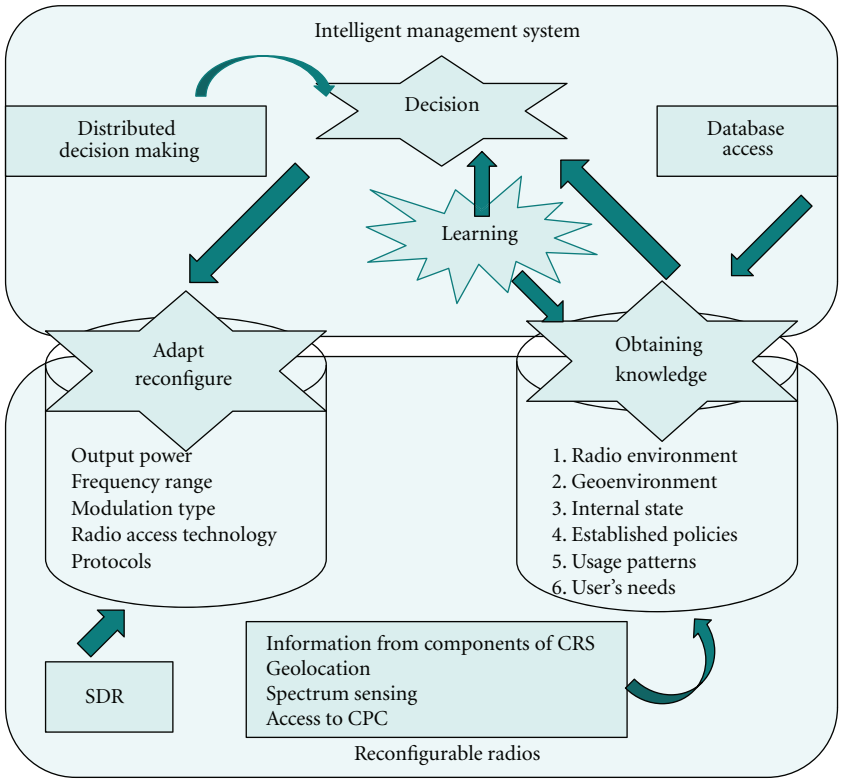


Figure 2: High-level CRS concept.

dynamically and autonomously makes reconfiguration decisions according to some predefined objectives, for example, in order to improve efficiency of spectrum usage. Based on the decisions made, the CRS adjusts operational parameters and protocols of its reconfigurable radios. Such parameters include output power, frequency range, modulation type, and radio access technology (RAT) protocols.

Software-defined radio (SDR) approach is used to implement the reconfigurations. Also, the CRS can learn from its decisions to improve its future decisions. The results of learning contribute to both obtaining knowledge and decision making.

CRS can be classified into two types: heterogeneous CRS and spectrum sharing CRS. The first type uses the network centric approach where one or several operators operate several radio access networks (RANs) using the same or different RATs. Frequency bands allocated to these RANs are fixed. Cognitive network optimizes radio resources and improves the QoS. The second type of CRS is sharing CRS, where several RANs using the same or different RATs can share the same frequency band by using the unoccupied subbands in an intelligent and coordinated way. Most of standardization activities are related to this type of CRS.

3. Status in Regulations and Standardizations

3.1. Regulations. The major regulatory agencies are developing rules for the unlicensed use of TVWS such as the FCC in the United States, Ofcom in the UK, and the Electronic Communications Committee (ECC) of CEPT in Europe.

The FCC provided the final rules for TVWS in 2010 [9]. There is ongoing proceeding for secondary use of the 2.36 GHz to 2.4 GHz band for medical area networks. Other opportunistic spectrum access beyond the already completed TVWS proceedings and cognitive techniques to better utilize the radio spectrum are currently under investigation.

Ofcom has also made significant progress in developing regulations for the TVWS with a first public consultation in 2009 [10]. The statement on white spaces devices and implementation of geolocation databases was released on September 1st, 2011 [11]. The detailed rules will be released in the future.

The ECC studied the technical and operational requirements for the operation of CRS in the WS of the UHF broadcasting band (470–790 MHz) [12]. This work is used as the starting point for regulatory activities within the ECC.

3.2. Standardizations. Currently international standardization of CRS is performed at all levels (ITU, IEEE, ETSI, and ECMA) [7, 13]. They are considering multiple deployment scenarios and business directions.

In ITU, Working Party (WP) 1B has worked on the definition of SDR and CRS and their relationship and summarized the technical and operational studies, and relevant recommendations. It has considered the SDR and CRS usage scenarios in different radio services and regulation

implications. The WP 5A is currently addressing the definition, description, and application of CRS in the land mobile service.

IEEE is very active in CRS. In 802 WGs (LAN/MAN), the activity to define CRSs is currently performed in the 802.11 and 802.22, while the activity to specify components of a CRS is currently performed in 802.19, 802.21, and 802.22. 802.11y is an amendment for 3650–3700 MHz operation in USA defining new regulatory classes, transmit power control, and dynamic frequency selection for 802.11 to share frequency bands with other users. Draft standard P802.11af is an amendment for TVWS operation defining standardized modifications to both the 802.11 physical (PHY) layers and medium access control (MAC) sublayer to meet the legal requirements for channel access and coexistence in the TVWS. Draft standard P802.19.1 concerns TVWS coexistence methods. IEEE 802.21 focuses on media independent handover services enabling the optimization of handover between heterogeneous IEEE 802 networks, and facilitating handover between IEEE 802 networks and cellular networks. The draft standard P802.22 is on policies and procedures for operation in the TV bands. It specifies the air interface, including the cognitive MAC and PHY, of point-to-multipoint wireless regional area networks, operating in the unlicensed TV bands between 54 MHz and 862 MHz. Draft standard P802.22.1 is to enhance harmful interference protection for PUs operating in TV bands.

The IEEE DySPAN standards committee, whose predecessor is the IEEE P1900 standards committee, develops IEEE standards for radio and spectrum management with focus on improved use of spectrum. Its working groups and resulting standards numbered in the 1900 range propose new techniques and methods of dynamic spectrum access (DSA), which requires managing interference and coordination of wireless technologies and includes network management and information sharing. The IEEE 1900.1 standard related to terminology and concepts was published in 2008. The IEEE 1900.4 published in 2009, defines management system to support network-terminal distributed optimization of radio resource usage and improvement in QoS in heterogeneous wireless networks. The IEEE 1900.4a working group is defining architecture and interfaces for DSA networks in TVWS. The IEEE 1900.5 standard is dedicated exclusively to policy language requirement and architecture. The IEEE 1900.6 standard deals with spectrum sensing interfaces and data structures. IEEE P1900.7 is a draft standard on radio interface for white space dynamic spectrum access radio systems supporting fixed and mobile operation. It specifies a radio interface including MAC sublayer and PHY layer of white space dynamic spectrum access radio systems supporting fixed and mobile operation in white space frequency bands, while avoiding causing harmful interference to incumbent users in these frequency bands. The standard provides means to support P1900.4a for white space management and P1900.6 to obtain and exchange sensing-related information (spectrum sensing and geolocation information). Recently the ad hoc on dynamic spectrum access in vehicular environments was created. The purpose is to consider interest in, feasibility of, and necessity of developing a standard radio interface for vehicular communications.

Table 1: Summary of CRS activities in standardization and open issues.

Functionality	Covering organization
Definitions	IEEE SCC 41, ETSI, ITU-R
Coexistence	IEEE 802.19, IEEE SCC 41
SDR	IEEE SCC 41, SDR Forum, ITU-R, OMG, ETSI
Radio interfaces	IEEE 802.22, 3GPP, ECMA
Heterogeneous access	ETSI, IEEE SCC 41
Spectrum sensing	IEEE 802.22, IEEE SCC 41
Testing	—
Networking	—
Security	—

The ETSI Reconfigurable Radio Systems (RRS) Technical Committee (TC) is also active in standardizing SDR and CRS [14]. TC RRS main responsibility is to carry out standardization activities related to reconfigurable radio systems (RRS) encompassing both SDR and CR with a focus on specific needs of the European Regulatory Framework, and CR/SDR TV white space standards adapted to the digital TV signal characteristics in Europe. Two out of the four working groups within ETSI RRS have activities resulting in standardization of potential regulatory aspects of CRS and SDR. Working group 3 has proposed and investigated the feasibility of standardizing a functional architecture for management and control of reconfigurable radio systems and cognitive pilot channel. SDR-related standardization is considered for both base station and mobile device. Working group 2 relies mainly on mobile device SDR related interface standardization. ETSI RRS is also working on operation in WS frequency bands and coexistence architecture for cognitive radio and investigating security and threats issues.

ECMA-392 released in 2009, specifies MAC and PHY for personal/portable cognitive wireless networks operating in TVWS, a MUX sublayer for higher layer protocols and a number of incumbent protection mechanisms.

3GPP is also interested in standardizing CR-like features in its future releases. For example, the idea of a cognitive reference signal is proposed through which each RAN can broadcast the interference level, frequency bands, and RATs of other networks, and other information that can help newly joined user equipment to choose the best RAN.

We summarize different aspects of standardization for CRS in Table 1. There are a lot of standardization activities for CRS with a lot of overlapping. However, many open issues have not been covered at all until now, especially the security issues.

4. Research Challenges

CRS covers multidisciplinary areas attracting a large number of research works with many interesting obtained results.

The challenges remain numerous, namely, intelligence distribution and implementation, delay/protocol overhead, cross-layer design, security, sensing algorithms, and flexible hardware design. Due to the huge amount of published papers and the interdisciplinary nature of the topic, it is very difficult to provide an exhaustive analysis of all research works available on CRS. The purpose of this section is therefore to briefly describe challenges which are yet open and current under debate in the framework of research on CRS.

4.1. Decision Making. As CRS is driven by a decision making, the first relevant research challenge is where and how the decision (e.g., the decision on spectrum availability, strategy for selecting channel for sensing or access, or how to optimize radio performance) should be taken. The first issue is directly related to whether the cognitive process should be implemented in a centralized or distributed fashion. This aspect is more critical not only for cognitive networks, where intelligence is more likely to be distributed, but also for cognitive radios, as decision making could be influenced by collaboration between them and also with other devices. The second issue is the choice of the decision algorithms (e.g., neural networks, genetic algorithms, ant-colony optimization, etc.) which should be customized to fulfill the CRS requirements.

4.2. Learning Process. Research in machine learning has grown dramatically recently, with significant amount of progress. One of the important aspects of the learning mechanisms is whether the learning performed is supervised or unsupervised. In the context of a CRS, either technique may be applied. The first challenge of learning is to avoid wrong choices before a feasible decision, especially in autonomous or unsupervised learning process. The second issue is to concretely define learning process in the context of CRS, its objectives and contributions.

In terms of implementation and algorithm design, the cognitive functionalities, which are related to enabling devices or networks to learn from past decisions to improve their behavior, are too much complex. The design of the learning algorithm represents by itself a challenge, and measurements which should be employed by learning open new issues related to which measurements to use and how to perform them.

4.3. Cross-Layers. While the aspect of interprotocol interaction is included in the concept of cognitive network as means to support user and applications requirement, no relevant and comprehensive analysis is available to address the performance and, in general, the behavior of applications and networks based on CRS technology. The design of cognitive or self-organized network is itself a challenging task, in particular, the outer and inner loops coordination, the networking middleware for knowledge exchange, and intersystem networking for sharing and cooperation.

Challenge is also in the design of high layers including MAC sublayer and network layer, spectrum management functions integrated at the different layers of the network protocol stack, cognitive radio resource management and coordination, various protocols and routings. Many technologies will be using multiple frequency bands. As a result, challenges in interoperability, including coexistence, cooperation and collaboration for devices, and networks signaling with cross-layer interfaces and interlayer signaling are to be solved.

Although many papers deal with the cross-layer design, the issues addressed are still specific and tailored to a given technology with the lack of its general. We need to provide cross-layer designs for more general classes of communications schemes.

4.4. Security. The challenges of employing CRS include that of ensuring secure devices operations. Security in this context includes enforcement of rules. Enforcement for static systems is already a challenge due to the amount of resources necessary to authorize equipment, the requirement of obtaining proof that violations have occurred, and the determination of the violator' identities. As the systems become more dynamic, there is an increase in the number of potential interaction that can lead to a violation. Additionally, this leads to a decrease of the time and special scales of these interactions. Both of these changes will amplify the enforcement challenges.

The first issue is on equipment authorization, especially on evaluation criteria and security certification. It becomes even more problematic with the employment of self-learning mechanisms. Software and hardware certification will not provide sufficient assurances that the device conforms to the operational envelopes.

Software certification and the security of the software are also challenging area, especially when software provides the control of dynamic systems. The security of that software is critical to ensure that rogue behavior is not programmed into the devices.

The number of combination of interactions is high and the mobility and the agility of CRS is great, so the monitoring mechanisms are challenging tasks.

Also, security is to deal at the protocol layer with key exchanges which are very adapted to a highly reliable physical layer and centralized network. However, in CRS with distributed cognitive networks, traditional cryptographic schemes are not adapted.

4.5. Sensing. Following challenge is about spectrum sensing, especially on the accuracy on spectrum occupancy decision, sensing time, and malicious adversary, taking into account the fundamental limits of spectrum sensing algorithms due to noise uncertainty multipath fading and shadowing [15]. In order to solve hidden PU problem and mitigate the impact of these issues, cooperative spectrum sensing has been shown to be an effective method to improve the detection performance by exploiting spatial diversity in the observations of spatially located CRs [16]. Challenges of cooperative sensing include reducing cooperation overhead, developing efficient information sharing algorithms. The

coordination algorithm for cooperation should be robust to changes and failures in the network, and introduce a minimum amount of delay.

The most prominent hardware trial for spectrum sensing thus far has been the FCC field trial conducted in 2008 by the office of engineering and technology [17]. Although the spectrum sensing approach exhibited good sensitivities satisfying stringent regulation requirements, the future spectrum sensing hardware should improve the receiver selectivity and receiver desensitization, especially when the adjacent channels have high powers. The geolocation database-based approach is able to identify occupied channels with 100% accuracy. However, for identification of unoccupied channels, it did not exhibit the best performance, presumably due to incomplete information in the database. This shows that the spectrum sensing alone works to some degree, but the performance could be further enhanced especially in the identification of occupied channels. Combining a geolocation database with spectrum sensing may be a better option provided that the CR device cost and power dissipation are decreased.

4.6. Geolocation. Geolocation is an important CR enabling technology due to the wide range of applications that may results from a radio being aware of its current location and possibility being aware of its planned path and destination. When CRS uses the geolocation technology for location determination combined with a database look-up, each access point (AP) may be connected to one, or multiple databases which provide information on the unused TVWS channels that are available at the location of AP and they provide also information on maximum transmit power levels usable in each channel. Furthermore, the use of master-slave technology is encouraged so that the necessary functionalities for database lookup and channel selection need to be implemented only in APs. It keeps the complexity and cost of end-user devices to a minimum.

However, the challenges in this area are who and how to implement the data base, how to feed it. Providing incumbent (or PUs) databases requires knowledge of the locations of CR devices whose precision should be specified. If global positioning service (GPS) is equipped and CR devices are outdoor, obtaining their geolocations may still be less a technical challenge. If no GPS is available or if CR devices are indoor, then obtaining geolocations becomes a challenging task.

5. Implementation Challenges

Although the theoretical research for CRS is blooming, with many interesting results, hardware implementation and system development are progressing at a slower pace, because of the complexities involved in designing and developing CRS [17]. In this section, we present the implementation challenges of CR in the system on chip (SoC) integration's perspective.

The first SDR architecture was proposed by Mitola and Maguire [1], in which the RF and analog processing are reduced to only a pair of data converters, thus providing the maximum flexibility and programmability through the digital processing block. This idealistic approach, however, suffers from the poor tolerance of to the interferers. In many wireless applications, a small desired signal could be accompanied by several large in-band signals created by nearby transmitters of the same communication standard or out-of-band blockers caused by any transmitter. At times, these blockers could be as much as about 100 dB larger than the desirable received signal, which, due to the lack of any filtering in this idealistic approach would demand an impractical dynamic range of about 100 dB on the ADC. This long-term requirement is very far beyond the limits of the technology available as will be shown in ADC/DAC challenges subsection. Research on high-performances ADC/DAC is going on with significant progress, especially with hybrid-filter bank and time-interleaving architectures.

For short- and midterm implementation, the block diagram of a CR may be presented in Figure 6. In order to operate on very wide band or multibands simultaneously, parallel processing is employed from antennas to analog to digital interfaces as shown in Figure 3. Multiantennas are necessary for multi-input-multioutput (MIMO) operation and/or multibands operation. After antennas, a passive module is used for switching or duplexing, RF filtering, and impedance matching between antennas and power amplifiers (PAs). This module is composed of a range of submodules in order to cover a wide bandwidth or enable simultaneous communications. Then multireceiver (Rx) and multitransmitter (Tx) are followed before a multi-ADC/DACs module. A high performance and very flexible digital baseband carries out not only all conventional processing for modulation and demodulation, coding and encoding, and so forth, but also digital filtering, dc offset cancellation, digital automatic gain control, calibration and correction of analog errors and non linearities. Combining with control plane and sensor, a feedback from baseband to RF front-end and transceiver are necessary to boost the performance of the analog part. The challenges of RF front-end and transceiver in the short-/midterm are to reduce the off-chip and passive components, increase their frequency-agility, minimize the power dissipation, and reduce area. These challenges will be discussed more in detail in the next two subsections.

CR needs to adapt transmission and receiver parameters to avoid causing interference to PUs and maximize spectral efficiency. To avoid causing interference, numerous techniques can be used and combined such as frequency tuning (adaptive frequency hopping, dynamic frequency selection and RF band switching), OFDM subchannelization, channel aggregation, time multiplexing, power control, modulation and coding for QoS adaptability, beam-forming, and space-time coding for MIMO. To maintain link in adverse conditions, wide dynamic range especially for analog-to-digital converter (ADC) and high sensitive receiver with rapid adaptation to changes in interference temperature are required.

CR will be also based on strong cross-layer interactions. For example, the cognitive spectrum management involves

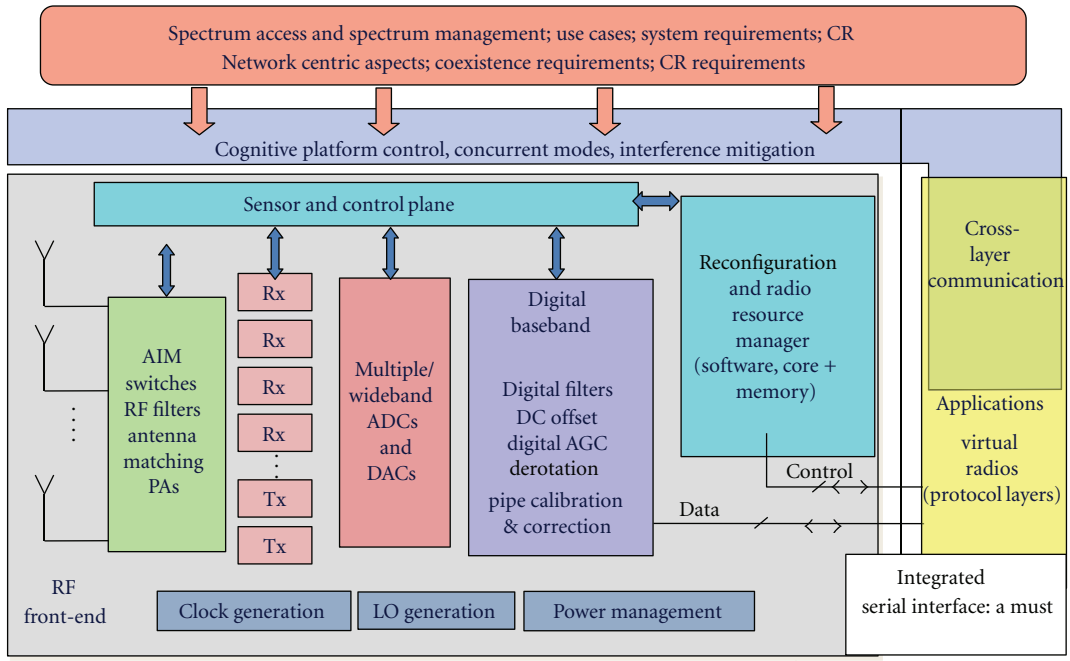


Figure 3: Full cognitive radio grail quest.

intelligent use of spectrum based on anticipating the demand for spectrum by the user and previous observation of user behavior. Another cognitive behavior is to monitor the environment in which the CR is operating and then manage the resource intelligently based on expectations or experiences.

5.1. RF Front-Ends Transceiver Challenges. A key bottleneck in CR has always been and continues to be the frequency-agile RF front-ends that can easily be coupled with the parts of the CR that carry out the digital processing—be they pure software systems or a mix of hardware and software. This subsection presents short-/midterm solution/requirements for RF front-end transceiver of CR devices. The state of the art related to this topic is described and design challenges are also presented.

CR transceiver should be able to use any available band, adapt to multiple access methods and adaptive modulation scheme, switch quickly between links, and communicate with 2 or more points at a time. Therefore, the RF section needs to be particularly flexible. In addition, CR receiver should be able to sense unused frequency bands if necessary.

One of the most difficult engineering concerns in the RF portion of SDR/CR, much more difficult than as in traditional OFDM modems (e.g., in simultaneous communications of diverse RATs), is handling very large peak-to-average power ratios (PAPRs) [18]. The first issue related to high PAPRs is the efficiency of the PA which suffers dramatically because the PA must be oversized in terms of its average power requirement. In addition to creating design problems for the PA, the high PAPR also requires highly linear upconverters leading to high power consumption.

OFDM modulation and simultaneous usage of multi-RATs are also very sensitive to the intermodulation distortion (IMD) that results from mild nonlinearity in the RF. For the RF components in the transmitter, the linearity requirement must be met at reasonably high power levels. With the associated requirement for high efficiency, this places extreme design challenges upon the architectural realization. For the RF components in the receiver, the power levels are much lower but the linearity requirement is even more challenging. This arises from the possible presence of adjacent channel interference from other radio systems, independent but closely located.

Resulting from the possibility of the interfering transmitter location being nearer than the desired transmitter, as well as the possibility of channel shadowing on the desired signal, the adjacent channel blocker is often received with a significantly stronger signal power level than the desired channel. The receiver selectivity becomes therefore more and more important. The linearity requirement for the receiver components must include this anticipated level of blocker power above the desired channel. CR receiver should have a wide dynamic range, that is with the ability to handle a large interferer whilst simultaneously receiving a small wanted signal. It means also that it is able to handle large amplitude signal whilst also having a low noise floor. Two direct consequences of this are difficulty in achieving acceptable noise figure performance for the overall receiver, and difficulty in achieving sufficient dynamic range for the ADCs.

As in OFDM system, phase noise and distortion should be taken carefully in SDR/CR, the problematic is almost the same. In terms of receiver requirements, it should exhibit a good sensitivity, leading to low noise amplifier (LNA)

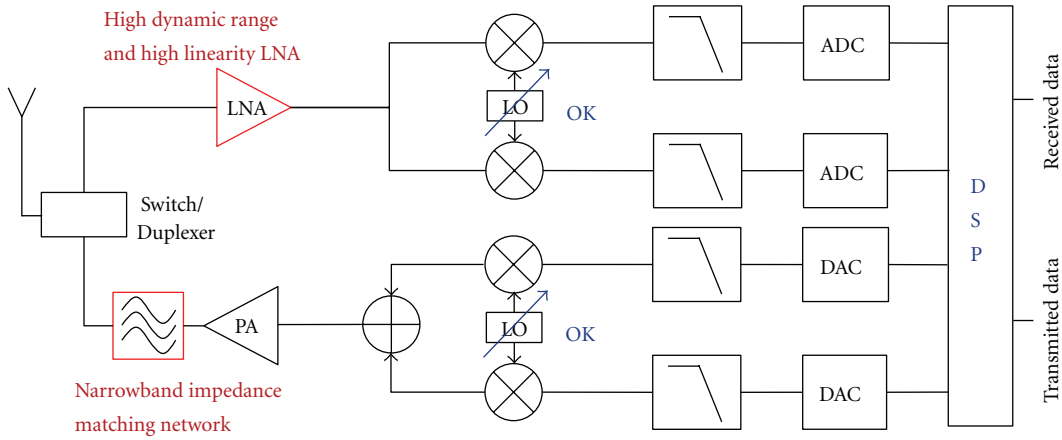


Figure 4: CR transceiver architectures—wideband radio.

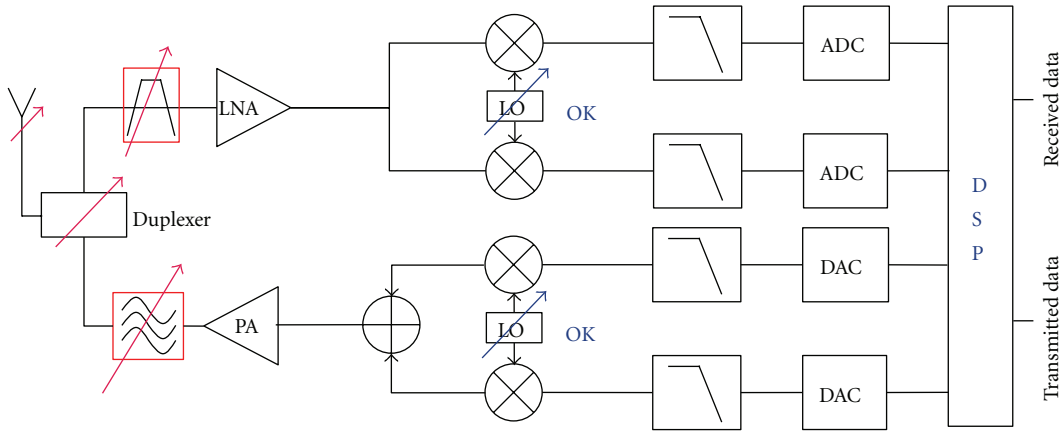


Figure 5: CR transceiver architectures—tunable radio.

achieving low noise factor (NF) (<3 dB) and filter design resulting in low insertion losses (<1 dB). It needs to achieve a high LNA and mixer linearity (IP3, IP1), adequate filtering, and low local oscillator (LO) phase noise, and spurious to have a good blocker immunity.

In transmitter, the key requirements are high PA linearity as explained above and low PA noise necessary for low adjacent channel leakage power (ACLP), high PA efficiency and heat removal, and low filter insertion loss to reduce power consumption.

Traditional front-end technology cannot handle these requirements because they are generally band limited, both for the form of modulation used and the frequency band in which they operate. Even so called multimode/multiband and wide band transceivers have limitations and generally operate by switching front ends as required [19].

Typically linearity, selectivity, phase distortion, and phase noise issues are addressed through the addition of costly and power-hungry external components including surface acoustic wave (SAW) filters and crystal oscillators that contribute to a higher system bill of materials (BOMs) and increased power consumption and lack of flexibility required for SDR/CR. The combination of innovative architectures and on-chip filters reduce BOM costs and power dissipation; however, the signal corruptions previously discussed can become problematic.

Solutions for improved performance in SoC's perspective include process selection, innovative architectures, the use of on-chip passive components when possible. Although there are some advantages to using a GaAs process for higher linearity or a bipolar process to reduce noise for RF signal processing, the all CMOS radio is the preferred choice for cost, especially when designing radios with significant digital subsections as the case in SDR/CR. Innovative architectures use complex domain RF and digital signal processing techniques to mitigate the effects of analog imperfections and smart calibration sequences at adequate timeframe to improve performance of the system. In the following, we will present some research directions in all CMOS radio for SDR/CR and associated challenges.

As shown in Figures 4 and 5, CR transceiver architecture can be wideband radio or tunable radio. Considering wideband radio, the challenges are not only on ADC/DAC which is presented in the next subsection, but also the high dynamic range and high linearity LNA, and narrowband impedance matching network. Considering tunable radio, the key issues

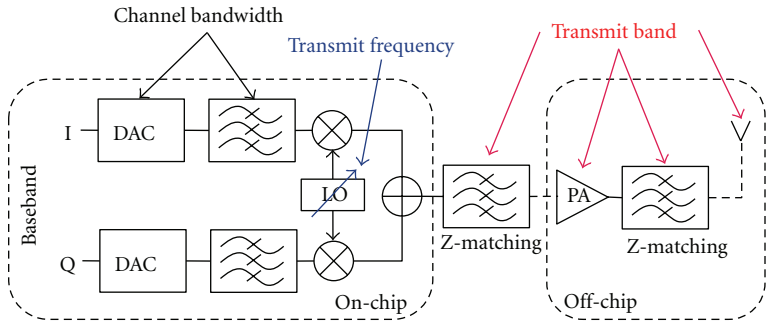


Figure 6: RF transmitter limitations.

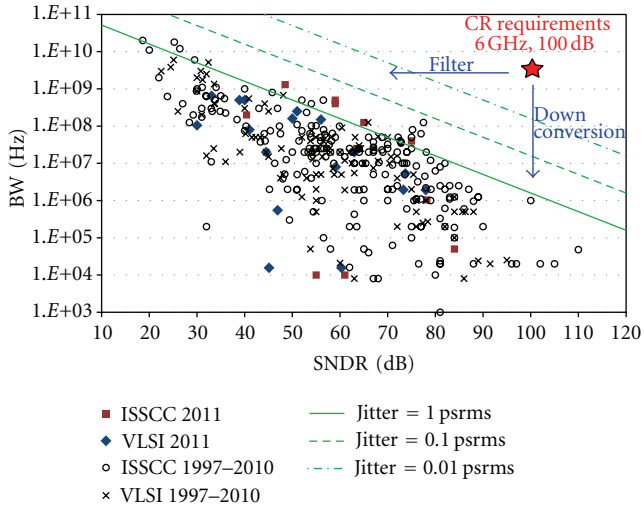


Figure 7: ADC SNDR as a function of bandwidth.

are related to the frequency-agile antenna, duplexer, and passive filters. In this direction, some solutions have been proposed, for example, a tunable integrated duplexer in [20], tunable RF BAW filters in [21], using metamaterials as BST in [22], or RF MEMS [23]. However the performances are not good enough and the tuning range is quite small. The challenge is still a low-loss pass-band filter with a small size which has an electrically tunable center frequency with a wide tuning range (of 20% would be truly disruptive).

An alternative solution for tunable receiver performing blocker rejection is based on circuit architecture using translational loops with feedforward structure [24]. However, it requires very high linearity LNA while sacrificing the NF and introducing an LO leakage issues. The feedback structure proposed in [25] achieves higher out of band (OOB) linearity while maintaining a lower NF. Another solution improving the OOB linearity and the receiver flexibility has been proposed in [26], using both direct conversion receiver and continuous-time delta sigma converter with upconverted RF feedback loop but for a single band and at the cost of a substantial increase of noise figure due to the feedback loop. The OOB filtering improvement is limited to 15 dB in practice which limits the OOB linearity enhancement. The future design challenge for [25, 26] is related to higher bandwidth operation (up to 40 MHz). In addition concerning [26], future design challenges are also related to multiband capability, OOB filtering enhancement without compromising the noise figure performances and the power consumption.

Receiver architectures particularly suited to partial SDR make use of impedance reflection from baseband to RF port. Connecting the antenna directly to a passive mixer without an RF LNA provides very increased tuning range and linearity [27]. Complex mixer-first, LNA-less architecture [28], provides very good OOB linearity and low NF with baseband programmable RF band-pass filter, wide tunable frequency range, and complex impedance matching. The design challenge for this approach is related to the extension of frequency range (up to 6 GHz) without degradation of performances.

The RF transmitter limitations are presented in Figure 6. With the success of the all-digital techniques for RF frequency synthesis in all-digital PLL (ADPLL) [29] and the apparition of digital PA concept showing direct reconstruction of the signal envelope in the RF domain, mostly digital transmitters have been demonstrated such as digital polar transmitter [30] and digital transmitter using direct modulation [31]. Although the design of almost digital transmitter is not as challenging than that of frequency-agile receiver, some issues are still remaining, especially decreased ACLP, increased PA efficiency using digital predistortion technique.

5.2. ADC and DAC Challenges. In ideal SDR and CR, the RF signals are converted into the digital domain as close to the antenna as possible. In this way all the processing is handled by the digital signal processing. In this case, not only must the ADC and DAC have a 100 dB dynamic range (with NF = 4 dB, noise floor in 10 MHz is at –100 dBm, interference and maximum blocker are supposed at 0 dBm), and be able to operate over a very wide range, extending up to 6 GHz at least, but the transmitter must be able to handle significant levels of power. The power consumption of the ADC could be 1 kW if the 1 pJ/conversion is assumed.

As illustrated in Figure 7, these requirements are very far beyond the limits of the technology available [32]. Therefore, as mentioned earlier, realistic CR receivers for short-/midterm are based on downconversion and filtering

Table 2: CPU-ARM perforamnces.

Technology	2007-8	2009-10	2011-12	Improvement
ARM Processor	ARM11 470–700 DMIPS	Cortex-A8 1,200–2,000 DMIPS	Dual Cortex-A9 5,000 + DMIPS	10× + SMP
Ext. display	VGA	XGA	WUXGA + HDMI	8× + HDMI
Video	VGA–30 fps	720–30 fps	1080p-30 fps	7×
3D graphics	2 Mtri/s OpenGL ES1.1	10 + Mtri/s OpenGL ES2.0	20+ Mtri/s OpenGL ES2.0	10× + Pgm. shaders
Imaging	3–5 MP	8–12 MP	16–20 MP	7×
Audio	15 hrs	40 hrs	140+ hrs	10×
DDR memory	128–256 MB	256–512 MB	1–2 GB	8×
Mass storage	8–16 GB	16–32 GB	64–128 GB	8×
Process	90 nm	65/45 nm	45 nm/beyond	3+ nodes

in front of the ADC to reduce both the required dynamic range and the conversion bandwidth.

For long-term research, wideband radio with very few amount of off-chip and passive components reduce analog signal processing and is still a desired solution. In this case, the challenge is mainly on ADC/DAC. In this optic, parallelism can be used in order to widen the conversion bandwidth. Parallel continuous-time $\Delta\Sigma$ ADC presented in [33] needs very complex digital synthesis filters which become even more complex when the channel number is increased. Hybrid filter bank based ADC [34] could be a promising solution for CR applications. However it is really sensitive to analog filter errors and imperfections, necessitating high resource calibration technique. The most natural solution employs time-interleaving [35] giving outstanding performance in terms of speed. But it lacks the resolution and dynamic range. The challenges are background calibration of frequency-dependent channel mismatch and time skew errors corrections without the need for any special calibration signal or postproduction trimming. Digital postlinearization is desired to suppress low order nonlinearities of parallel ADCs and nonlinearities caused by preADC analog components. Another technique is to combine time multiplexing and frequency multiplexing by using band-pass charge sampling filters as analysis filters in hybrid filter banks architecture [36, 37]. This leads to the reduced complexity of analog analysis filters, and at the same time the sensitivity to analog errors and imperfections are reduced. However, a deeper investigation on practical implementation to widen the bandwidth and sensitivity is really needed.

Almost digital RF transmitters employ an RF-DAC [30, 31]. Although some improvements are still encouraged, the design is less challenging than that of ADCs. The key issues in IC design are rather in Rx and especially with spectrum sensing algorithms implementation: broadband, high linearity, high dynamic range, and low noise.

5.3. Baseband Challenges. Looking at the trend of the digital core processor embedded in expensive smart phone devices, as shown in Table 2, the embedded CPU hosts are more and more powerful. As a result, a part of the digital filters in addition to the demodulation functions are implemented in the software domain. This is already the case in low latency systems as GPS and digital broadcast radio system. This trend is expected to continue paving the way of "true" SDR receiver.

In SDR/CR, the baseband architecture requires dynamic reconfigurations, high computational demand, up to 1 TOPS (100 GOPS are now required for 4G mobile terminal), and low power (roughly 500 mW) under real time constraints. In order to fulfill the unique requirements of embedded applications (real-time, low-power, and multitasking) and improve the flexibility, multiprocessor system-on-chip (MPSoC) has been introduced [38]. network-on-chip (NoC) based architecture can be used to enable fast run-time reconfiguration and reduce the power consumption with distributed power management [39]. 3D NoC is a promising solution for increased modularity and scalability [40]. 3D stacking is a very interesting technology for SoC, however, the matching between 3D technology and multicore is not straightforward due to some limitations of 3D technology. Dynamic task management has been proposed in [41] to solve MPSoC scalability and programmability problem, and improve resource allocation and energy efficiency.

With real-time constraints, the challenge in baseband architecture and design is to enable flexibility with reduced overhead in terms of performances, powerconsumption, and silicon area by optimizing the tradeoff between performances (dynamic reconfigurations, computational resources) and power efficiency, implementing efficient power management techniques and reducing the run-time management overhead with flexible/dynamic task management.

5.4. Spectrum Sensing Algorithm Implementation. Although, regulatory agencies beginning with FCC and then recently with Ofcom do not require the implementation of spectrum sensing as mandatory but only optional for white space devices operating in the TV bands as a secondary user, spectrum sensing stays a challenging task for CRS and very useful for a better spectrum management and would be a key feature for CR devices in the future. In terms of algorithms, the challenge is on the accuracy on spectrum occupancy decision, sensing time and malicious adversary, taking into account the fundamental limits of spectrum

Table 3: Spectrum sensing—regulation requirements.

	FCC 2010	Ofcom 09	IEEE 802.22
Max. power adjacent to TV/other [dBm]	16/17	4/17	36
Adjacent-channel emission	−72 dBc*	−46 dBc	−49 dBc
Digital TV sensing threshold [dBm/MHz]	−114/6	−120/8	116/6
Wireless microphone sensing threshold [dBm/MHz]	−107/0.2	−126/0.2	−10/0.2
$\text{Probability}_{\text{detection}}/\text{probability}_{\text{FalseAlarm}}$			90%/10%
Backoff time(s)	2	<1	2
Sensing frequency (1/minutes)	1	60	30
Bandwidth	6	8	6, 7, 8
Transmit power control	Yes	Yes	Yes
Modulation	Free	Free	OFDM

* Measured in 100 kHz with reference to total power in 6 MHz.

sensing algorithms due to noise uncertainty multipath fading and shadowing and hidden PU problem as mentioned above.

A spectrum sensing algorithm is characterized by its detection, false alarm, and miss detection probabilities but also SNR regime, sensing time and frequency, and especially its implementation complexity.

In a practical implementation, it is interesting to build this function with a very low power receiver chain parallel to the main receiver path. Because of the SNR wall issue [15] and particularly the hidden node problem that should be taking and mitigated with additional margin in the threshold detection level, this function requires a very high quality RF receiver in terms of NF and linearity. Spectrum sensing algorithms are implemented in the baseband. In terms of implementation complexity and power consumption, it depends on the used algorithms, but it is quite simpler compared to conventional baseband signal processing for demodulation. The most challenging task is located in the RF front-end design, in particular low NF, high linearity, and wide dynamic range:

$$\begin{aligned}\text{SNR}_{\text{min}}[\text{dB}] = {} & P_{\text{min}}[\text{dBm}] - \text{NF}[\text{dB}] + 174[\text{dBm/Hz}] \\ & - 10\log_{10}\text{B}[\text{dB}\cdot\text{Hz}].\end{aligned} \quad (1)$$

The equation above represents the relation between the minimal SNR value (SNR_{min}) for a given algorithm and the minimal sensing sensitivity or threshold (P_{min}), the NF of the receiver (NF) where B is the channel bandwidth to be sensed.

As shown in Table 3, the spectrum sensing requirements are very stringent with very low sensing threshold. This sensitivity level is even lower in some cases in CEPT-SE43 (−169 dBm in the worst case). It leads to a very low NF requirement (less than 7 dB) and very high out of band linearity. Baseband implementation for algorithm itself must be optimized in terms of computational resources but also power consumption.

6. Conclusion and Discussion

This paper presented an overview and challenges of CRS with focus on implementation of cognitive radio. We summarized the status of the related regulation and standardization activity and then pointed out some key research challenges, especially implementation challenges of cognitive radio.

Having three key capabilities such as cognitive capability, reconfigurable capability, and learning capability, CRS has the potential of being a disruptive force within spectrum management. Using SDR for implementing reconfigurable radios, CRS provides additional flexibility and offers improved efficiency to overall spectrum use. It can exploit the spectrum holes or white spaces of licensed spectrum bands provided that it does not cause harmful interference to any primary users in order to significantly improve the efficiency of spectrum usage.

CRS offers also the possibility of flexibly managing the spectrum in a dynamic manner in heterogeneous radio access networks. Through intelligent management mechanisms, frequency bands can be allocated to RATs dynamically in a way such that the capacity of each RAT is maximized and interference is minimized. Network operator may employ different RATs dynamically over time/frequency/location and acquire or exchange the spectrum usage rights. The cognitive devices may autonomously and dynamically adapt to the diverse heterogeneous radio access networks.

CRS covers multidisciplinary areas attracting a large number of researches with many interesting obtained results. The challenges remain numerous, namely, intelligence distribution and implementation, security, delay/protocol overhead, cross-layer design, flexible hardware design, and so forth, CRS will always be limited by physically possible bounds. Limitations depend on the usage model, future standardization and highly on the carrier to noise ratio needed to decode the signal and the signal bandwidth. It is anticipated that semiconductor manufacturers will make steps to improve the selectivity, linearity, and agility performances. But these performances will lead to additional costs that could be offset by new design techniques, and

for the digital portions, denser CMOS node. The timeframe to make this happen is more linked to the economical and business model for CRS than real technological issues.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under Grant agreement SACRA no 249060, from French industry ministry and the ENIAC Joint Undertaking in ARTEMOS Project, and from Institut Telecom in Green RAN ICT-Asia Project.

References

[1] J. Mitola III and G. Q. Maguire Jr., "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[2] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[3] R. Tandra, S. M. Mishra, and A. Sahai, "What is a spectrum hole and what does it take to recognize one?" *Proceedings of the IEEE*, vol. 97, no. 5, pp. 824–848, 2009.

[4] M. Mueck, "ETSI TC RRS (Reconfigurable Radio Systems)—building standards for SDR and CRS," in *Proceedings of the Workshop on Software Defined Radio and Cognitive Radio standardization*, Ispra, Italy, 2011.

[5] J. Wang, M. Ghosh, and K. Challapali, "Emerging cognitive radio applications: a survey," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 74–81, 2011.

[6] "Definitions of Software Defined Radio (SDR) and Cognitive Radio System (CRS)," ITU-R Report SM 2152, 2009.

[7] S. Filin, H. Harada, H. Murakami, and K. Ishizu, "International standardization of cognitive radio systems," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 82–89, 2011.

[8] C. Moy, "High-level design approach for the specification of cognitive radio equipments management APIs," *Journal of Network and Systems Management*, vol. 18, no. 1, pp. 64–96, 2010.

[9] "In the matter of unlicensed operation in the TV broadcast bands: second memorandum opinion and order," FCC 10-174, Federal Communications Commission, 2010.

[10] Ofcom, "Digital dividend: cognitive access, statement on licence-exempting cognitive devices using interleaved spectrum," 2009.

[11] Ofcom, "Implementing geolocation: Statement," 2011.

[12] "Technical and operational requirements for the possible operation of cognitive radio systems in the white spaces of the frequency band 470–790 MHz," ECC Report 159, 2010.

[13] F. Granelli, P. Pawelczak, R. Venkatesha Prasad et al., "Standardization and research in cognitive and dynamic spectrum access networks: IEEE SCC41 efforts and other activities," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 71–79, 2010.

[14] M. Mueck, A. Piipponen, K. Kalliojärvi et al., "ETSI reconfigurable radio systems: status and future directions on software defined radio and cognitive radio standards," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 78–86, 2010.

[15] R. Tandra and A. Sahai, "SNR walls for signal detection," *IEEE Journal on Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 4–17, 2008.

[16] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: a survey," *Physical Communication*, vol. 4, no. 1, pp. 40–62, 2011.

[17] P. Pawelczak, K. Nolan, L. Doyle, S. Oh, and D. Cabric, "Cognitive radio: ten years of experimentation and development," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 90–100, 2011.

[18] S. Zabre, J. Palicot, Y. Louet, C. Moy, and C. Lereau, "Carrier per carrier analysis of SDR signals power ratio," in *Proceedings of the SDR Forum Technical Conference*, Orlando, Fla, USA, 2006.

[19] T. Sowlati, B. Agarwal, J. Cho et al., "Single-chip multiband WCDMA/HSDPA/HSUPA/EGPRS transceiver with diversity receiver and 3G digRF interface without SAW filters in transmitter/3G receiver paths," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC '09)*, pp. 116–117, February 2009.

[20] M. Mikhemar, H. Darabi, and A. Abidi, "A tunable integrated duplexer with 50dB isolation in 40nm CMOS," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC '09)*, pp. 386–387, February 2009.

[21] S. Razafimandimby, C. Tilhac, A. Cathelin, A. Kaiser, and D. Belot, "Digital tuning of an analog tunable bandpass BAW-filter at GHz frequency," in *Proceedings of the 33rd European Solid-State Circuits Conference (ESSCIRC '07)*, pp. 218–221, September 2007.

[22] K. B. Kim, "Tunable dual-mode filter using varactor and variable ring resonator on integrated BST/TiO2/Si substrate," *Electronics Letters*, vol. 46, no. 7, pp. 509–511, 2010.

[23] T. Oita, "RF MEMS: focusing on the next step," in *Proceedings of the IEEE Ultrasonics Symposium*, pp. 1173–1178, 2009.

[24] H. Darabi, "A blocker filtering technique for wireless receivers," in *Proceedings of the 54th IEEE International Solid-State Circuits Conference (ISSCC '07)*, pp. 77–588, February 2007.

[25] X. He and H. Kundur, "A compact SAW-less multiband WCDMA/GPS receiver front-end with translational loop for input matching," *ISSCC Digest of Technical Papers*, pp. 372–373, 2011.

[26] K. Koli, S. Kallioinen, J. Jussila, P. Sivonen, and A. Parssinen, "A 900-MHz direct delta-sigma receiver in 65-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 12, pp. 2807–2818, 2010.

[27] M. C. M. Soer, E. A. M. Klumperink, Z. Ru, F. E. Van Vliet, and B. Nauta, "A 0.2-to-2.0GHz 65nm CMOS receiver without LNA achieving >11dBm IIP3 and <6.5 dB NF," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC '09)*, February 2009.

[28] C. Andrews and A. C. Molnar, "A passive mixer-first receiver with digitally controlled and widely tunable RF interface," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 12, pp. 2696–2708, 2010.

[29] R. B. Staszewski, "State-of-the-art and future directions of high-performance all-digital frequency synthesis in nanometer CMOS," *IEEE Transactions on Circuits and Systems I*, vol. 58, no. 7, pp. 1497–1510, 2011.

[30] Z. Boos, A. Menkhoff, F. Kuttner et al., "A fully digital multimode polar transmitter employing 17b RF DAC in 3G mode," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 376–377, 2011.

[31] A. Pozsgay, T. Zounes, R. Hossain, M. Boulemnakher, V. Knopik, and S. Grange, "A fully digital 65nm CMOS transmitter for the 2.4-to-2.7GHz WiFi/WiMAX bands using 5.4GHz

ΔΣ RF DACs," in *Proceedings of the IEEE International Solid State Circuits Conference (ISSCC '08)*, pp. 355–619, February 2008.

[32] B. Murmann, "ADC Performance Survey 1997–2011," http://www.stanford.edu/~murmann/adcsurvey.html.

[33] J. Arias, L. Quintanilla, J. Segundo, L. Enríquez, J. Vicente, and J. M. Hernández-Mangas, "Parallel continuous-time ΔΣ ADC for OFDM UWB receivers," *IEEE Transactions on Circuits and Systems I*, vol. 56, no. 7, pp. 1478–1487, 2009.

[34] C. Lelandais-Perrault, T. Petrescu, D. Poulton, P. Duhamel, and J. Oksman, "Wideband, bandpass, and versatile hybrid filter bank A/D conversion for software radio," *IEEE Transactions on Circuits and Systems I*, vol. 56, no. 8, pp. 1772–1782, 2009.

[35] C.-C. Huang, C.-Y. Wang, and J.-T. Wu, "A CMOS 6-Bit 16-GS/s time-interleaved ADC using digital background calibration techniques," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 4, pp. 848–858, 2011.

[36] A. Gruget, M. Roger, V. T. Nguyen, C. Lelandais-Perrault, P. Bénabès, and P. Loumeau, "Wide-band multipath A to D converter for cognitive radio applications," in *Proceedings of the IEEE International Microwave Workshop Series on RF Front-Ends for Software Defined and Cognitive Radio Solutions (IMWS '10)*, pp. 73–76, Aveiro, Portugal, 2010.

[37] A. Gruget, M. Roger, V. T. Nguyen, C. Lelandais-Perrault, P. Bénabès, and P. Loumeau, "Optimization of bandpass charge sampling filters in hybrid filter banks converters for cognitive radio applications," in *Proceedings of the 20th European Conference on Circuit Theory and Design (ECCTD '11)*, pp. 785–788, Linpöping, Sweden, 2011.

[38] W. Wolf, A. A. Jerraya, and G. Martin, "Multiprocessor system-on-chip (MPSoC) technology," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 10, pp. 1701–1713, 2008.

[39] F. Clermidy, C. Bernard, R. Lemaire et al., "A 477mW NoC-based digital baseband for MIMO 4G SDR," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC '10)*, pp. 278–279, February 2010.

[40] F. Clermidy, F. Darve, D. Dutoit, W. Lafi, and P. Vivet, "3D Embedded multi-core: some perspectives," in *Proceedings of the Design, Automation and Test in Europe (DATE '11)*, pp. 1327–1332, 2011.

[41] C. Jalier, D. Lattard, G. Sassatelli, P. Benoit, and L. Torres, "A homogeneous MPSoC with dynamic task mapping for software defined radio," in *Proceedings of the IEEE Annual Symposium on VLSI*, pp. 345–350, July 2010.

Hindawi Publishing Corporation
VLSI Design
Volume 2012, Article ID 826350, 14 pages
doi:10.1155/2012/826350

Research Article

A System View on Iterative MIMO Detection: Dynamic Sphere Detection versus Fixed Effort List Detection

Christina Gimmler-Dumont,[1] Frank Kienle,[1] Bin Wu,[2] and Guido Masera[2]

[1] *Microelectronic Systems Design Research Group, University of Kaiserslautern, Erwin-Schroedinger-Straβe, 67663 Kaiserslautern, Germany*
[2] *Dipartimento di Elettronica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy*

Correspondence should be addressed to Christina Gimmler-Dumont, gimmler@eit.uni-kl.de

Received 2 December 2011; Accepted 27 January 2012

Academic Editor: Christophe Moy

Multiple-antenna systems are a promising approach to increase the data rate of wireless communication systems. One efficient possibility is spatial multiplexing of the transmitted symbols over several antennas. Many different MIMO detector algorithms exist for this spatial multiplexing. The major difference between different MIMO detectors is the resulting communications performance and implementation complexity, respectively. Particularly closed-loop MIMO systems have attained a lot of attention in the last years. In a closed-loop system, reliability information is fed back from the channel decoder to the MIMO detector. In this paper, we derive a basic framework to compare different soft-input soft-output MIMO detectors in open- and closed-loop systems. Within this framework, we analyze a depth-first sphere detector and a breadth-first fixed effort detector for different application scenarios and their effects on area and energy efficiency on the whole system. We present all system components under open- and closed-loop system aspects and determine the overall implementation cost for changing an open-loop system in a closed-loop system.

1. Introduction

Multiple-antenna (MIMO) systems are a promising approach to increase the data rate of wireless communication systems in rich-scattering environments. Spatial multiplexing is a spectrally efficient way to exploit the diversity of the MIMO channel while an outer error correction code ensures the desired quality of service for a given data rate. This setting is called a Bit Interleaved Coded Modulation (BICM) system (see Section 3). Particularly iterative MIMO detection attained a high attention in the last years. In an iterative receiver, reliability information is fed back from the outer channel decoder to the MIMO detector and vice versa. The resulting communications performance is improved by 3 dB and more compared to open-loop decoding [1, 2].

This improvement is gained at the cost of a highly complex signal detection (Section 4). Optimal detection by exhaustive search is infeasible for realistic scenarios (4×4 antennas, 16- or 64-QAM). Finding the right trade-off between communications performance and implementation complexity and understanding the implications on the whole receiver is one of the major challenges in the design of iterative MIMO receivers. MIMO detection algorithms and their implementations have been extensively studied in the literature (Section 2). They can be divided into classes with similar characteristics, for example, linear filters or breadth-first tree search algorithms.

The fixed effort list detector (breadth-first search, Section 4.2) and the sphere detector (depth-first search, Section 4.1) are among the most promising approaches to obtain a good communications performance in iterative systems at reasonable implementation complexity. The fixed effort detector processes the MIMO vectors at a constant throughput whereas the sphere detector has a dynamic throughput due to the nature of the depth-first search. However, the sphere detector is able to approach the optimum detection while the communications performance of the fixed effort detector is restricted by the storage requirements of the generated lists.

In this paper, we explore the design space for iterative MIMO detection from a system perspective comparing fixed effort and sphere detection. We start with an investigation of the system communications performance for both algorithms (Section 5) and continue with an architectural analysis of the complete receiver system. Not only the implementation of the MIMO detectors but also of the other building blocks in the iterative receiver (channel preprocessing and channel decoding) needs to be studied to analyze the whole system (Section 6). Therefore, it is mandatory to fix some shared design constraints. We introduce a generic architecture framework which connects the building blocks by system memories in order to be able to exchange individual blocks easily (Section 7.1). Characteristics of each block are analyzed in a system context (Section 7.2); for example, the channel decoder can employ different algorithms for open loop decoding and closed loop decoding.

A fair comparison of different MIMO detectors is only possible as a part of an iterative receiver. Different architectures have advantages for different system constraints, thus we compare fixed effort and sphere detector in several throughput centric and communication centric scenarios (Section 7.3). Eventually, we investigate the system cost in terms of throughput, area, and power when moving from an open-loop to a closed-loop system (Section 7.4). The corresponding area and energy efficiency numbers drop by more than a factor of 2 for closed-loop decoding with one iteration.

2. Review of State-of-the-Art Detection Algorithms and Their Implementations

Multiple-antenna systems employing spatial multiplexing increase the spectral efficiency. However, this improvement comes at the cost of an increased receiver complexity. Finding the right trade-off between communications performance and implementation complexity in MIMO detection is one of the key challenges in the receiver design.

In order to optimally solve the MIMO detection problem, an exhaustive search for the best solutions can be done over all signal constellations. The number of possible signal constellations increases exponentially with the number of antennas and the number of bits per modulation symbol. For a 4×4 antenna system employing 16-QAM, more than 65000 constellations exist. For 64-QAM, this number rises to more than 16000000. This makes an exhaustive search infeasible for a hardware implementation [9].

As the optimal exhaustive search is far too complex for hardware implementations, many suboptimal detection algorithms exist with a big range in communications performance and complexity. They can be divided into the following classes.

2.1. Linear MIMO Detection. Zero-Forcing (ZF) and minimum mean square error (MMSE) filters apply an inverse of the channel to the received signal in order to restore the transmitted signal [10]. These linear filters can be implemented at a low complexity; however, their communications performance is very low as well. The MMSE filter considers the noise power in the interference cancellation and therefore shows a slightly better performance.

2.2. Successive Interference Cancellation. The successive interference cancellation (SIC) technique was initially adopted by the vertical Bell Laboratories layered space-time (V-BLAST) system [11]. In contrast to the basic ZF and MMSE filters, SIC detects the transmitted streams sequentially. It chooses the substream with largest signal-to-noise ratio and removes the interference of each detected stream before continuing the detection process. The performance of the SIC algorithm is generally better than ZF and MMSE filters.

2.3. Breadth-First Tree Search Algorithms. For further improvement of the communications performance, the MIMO detection problem can be mapped on a tree search. The tree search algorithms can be divided into breadth-first and depth-first search algorithms.

Breadth-first algorithms offer a constant throughput with a small loss in communications performance compared to an optimal detection. Among the best known techniques are the K-best algorithm [12, 13] and the fixed-complexity detector [14]. While traversing the tree, the K-best detector keeps the K best nodes in each level. This requires sorting operations which result in a high implementation cost. The fixed-effort detector follows a regular tree traversal path which is determined at design time. This regularization enables the design of highly-efficient parallel architectures [14], however, at slightly lower communications performance than the K-best algorithm. In general, the communications performance of breadth-first algorithms depends on the number of nodes visited in each layer of the tree.

2.4. Depth-First Tree Search Algorithms. Depth-first detectors apply pruning criteria to remove parts of the tree in the search to reduce the computational complexity [15]. They approach the ML solution for hard output and the MAP solution for soft output. Sphere detectors achieve the best communications performance among the different detection techniques, but due to the nature of the depth-first search, their throughput is variable. The sequential tree search order makes it difficult to parallelize the detection. There exist many sub-optimal variants regarding enumeration technique, pruning criterion, or simplified metric calculations, for example, [3, 16].

The hardware implementation of sphere detection has been extensively explored for hard- and soft-output versions, for example, [17, 18]. Different forms of pipelining have been proposed to increase the architecture parallelism [3, 19].

2.5. Iterative MIMO Detection and Channel Decoding. In this paper, we investigate iterative receivers where MIMO detector and channel decoder exchange reliability information to increase the communications performance. Therefore, the aforementioned algorithms have to be adjusted to utilize the given soft-input information. Studer et al. implemented a soft-input soft-output extension of the linear MMSE filter (called MMSE-PIC) in [8]. Breadth-first algorithms have

been extended to list detectors. Thereby, the breadth-first algorithm generates a number of candidate vectors which are stored in a list. The iterative detection process is only based on the available vectors in the list. In contrast to breadth-first algorithms, soft information can be directly included in depth-first sphere detection algorithms, for example, [1, 2]. Witte et al. presented the first implementation of such a soft-input soft-output sphere detector in [5] based on the single-tree-search algorithm (STS) of [20].

2.6. State-of-the-Art MIMO Detection Architectures. Architectures for MIMO detection have been extensively studied in the literature for all kind of algorithms. Several silicon implementation results of the proposed MIMO detection architectures are listed in Table 1.

A fundamental one-node-per-cycle hardware architecture for the hard-output depth-first sphere decoder is introduced in [3] together with the l^{∞}-norm approximation for complexity reduction. This architecture has been firstly extended to a soft-output version in [4] by applying techniques including single-tree-search, sorted QR decomposition and LLR clipping, and further enhanced to be soft-input soft-output in [5], to perform iterative MIMO decoding. Other architectural improvements, such as the modified best first with fast descent (MBF-FD) MIMO detection [6], and the parallel and scalable architecture for modified metric first (MMF) list sphere detection (LSD), have been proposed to enhance detection efficiency and performance. The basic architectural considerations for implementing the depth-first sphere decoders are generalized in [21], from high-level architecture and enumeration strategy to approximations and pipeline interleaving.

The architecture for K-best algorithm is modified in [22] by applying bidirectional partial tree search and hybrid two-step scheme to reduce complexity. Another similar approach, namely, the early pruned technique, is applied to reduce the complexity of the K-best algorithm [7].

Besides the sphere decoders, several other MIMO detection algorithms have been investigated. In [23], the Markov chain Monte Carlo (MCMC) simulation techniques are reported to achieve comparable performance to LSD. The MMSE-SIC algorithm has also been improved to be soft-input soft-output and achieve very high throughput by applying parallel architecture [8].

3. System Model

In this paper, we focus on a bit interleaved coded modulation (BICM) scheme like that shown in Figure 1. The source generates a random infoword $\mathbf{u}$ of length K_c which is encoded by the channel encoder. The interleaved codeword $\mathbf{X}^N$ consists of N_c bits which are linearly grouped into N subblocks $\mathbf{x}_n$:

$$\mathbf{X}^N = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N). \tag{1}$$

Each subblock $\mathbf{x}_n$ consists of Q coded bits:

$$\mathbf{x}_n = \left(x_{1,n}, x_{2,n}, \ldots, x_{q,n}, \ldots, x_{Q,n}\right), \quad x_{q,n} \in \{-1, +1\}. \tag{2}$$

Each $\mathbf{x}_n$ is mapped directly to a complex symbol $s = \text{map}(\mathbf{x}_n)$ chosen from a 2^Q-ary QAM modulation scheme. M_T symbols are combined in one transmission vector $\mathbf{s}_t$. M_T is the number of transmit antennas:

$$\mathbf{s}_t = \left(s_{1,t}, s_{2,t}, \ldots, s_{m,t}, \ldots, s_{M_T,t}\right). \tag{3}$$

The whole modulated sequence is represented by

$$\mathbf{S}^T = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t, \ldots, \mathbf{s}_T). \tag{4}$$

T time slots are needed to transmit all symbols of one codeword. The transmission of vector $\mathbf{s}_t$ in time step t is modeled by

$$\mathbf{y}_t = \mathbf{H}_t \cdot \mathbf{s}_t + \mathbf{n}_t \tag{5}$$

with $\mathbf{H}_t$ the channel matrix of dimension $M_T \times M_R$ and $\mathbf{n}_t$ the noise vector of dimension M_R whose entries are zero-mean and unit variance Gaussian variables. The elements of $\mathbf{H}_t$ are modeled as independent, complex, zero-mean, Gaussian random variables. Real and imaginary part are independent variables each with variance $\sigma^2 = N_0/2$. It is assumed that $\mathbf{H}_t$ is ergodic, that is, its entries change independently after each channel use. Furthermore, $\mathbf{H}_t$ is perfectly known by the MIMO detector and all employed antenna constellations are symmetric with $M_T = M_R = M$. The received vectors $\mathbf{y}_t$ are gathered in the matrix $\mathbf{Y}^T$

$$\mathbf{Y}^T = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t, \ldots, \mathbf{y}_T) \tag{6}$$

with

$$\mathbf{y}_t = \left(y_{1,t}, y_{2,t}, \ldots, y_{m,t}, \ldots, y_{M_R,t}\right). \tag{7}$$

Before the decoding starts, the channel preprocessing applies a QR decomposition on $\mathbf{Y}^T$ and $\mathbf{H}_t$ (for details see Section 4). This results in the transformed received vectors $\hat{\mathbf{Y}}^T$ and updated channel matrices $\mathbf{R}_t$. The decoding process is iterative between MIMO detector and channel decoder. They exchange probability information on the codeword. The soft-in-soft-out MIMO detector determines the likelihood of the bits for each received vector $\hat{\mathbf{y}}_t$ using the a priori information $\mathbf{L}_t^a$ from the channel decoder. Only the extrinsic information $\boldsymbol{\lambda}^e = \boldsymbol{\lambda} - \mathbf{L}^a$ is passed on to the channel decoder.

The channel decoder processes the whole codeword at a time. It uses the interleaved a priori information $\boldsymbol{\lambda}^a$ from the MIMO detector for the calculation of the estimated information bit sequence $\hat{\mathbf{u}}$ and the a posteriori logarithmic likelihood ratios (LLRs) $\boldsymbol{\Lambda}$ of the codeword. The extrinsic information $\mathbf{L}^e = \boldsymbol{\Lambda} - \boldsymbol{\lambda}^a$ is returned to the MIMO detector thus closing the iterative loop.

4. MIMO Detection

A received symbol vector $\mathbf{y}_t$ can be seen as a weighted superposition of the entries of $\mathbf{s}_t$ disturbed by Gaussian noise. The task of the MIMO detector is the equalization and separation of the originally sent symbols $\mathbf{s}_t$. The MIMO detector works on one received vector $\mathbf{y}_t$ at a time.

Table 1: ASIC implementations of recently reported MIMO detectors.

Publication	[3]	[4]	[5]	[6]	[7]	[8]
Algorithm	Hard-output sphere decoder	Soft-output STS-SD	SISO STS-SD	Soft-output MBF-FD	Hard-output K-best	SISO MMSE-PIC
Antenna	4×4	4×4	4×4	8×8	4×4	4×4
Modulation	16-QAM	16-QAM	16-QAM	64-QAM	64-QAM	64-QAM
Iterative decoding	no	no	yes	no	no	yes
Constant throughput	no	no	no	no	yes	yes
Technology	250 nm	250 nm	90 nm	130 nm	130 nm	90 nm
Clock frequency	71 MHz	71 MHz	250 MHz	198 MHz	138 MHz	568 MHz
Core area	50 KG	57 KG	96 KG	350 KG	491 KG	410 KG
Max. throughput	169 Mbit/s	70 Mbit/s	72 Mbit/s	429 Mbit/s	1200 Mbit/s	757 Mbit/s
Power consumption	—	—	—	58.2 mW	185 mW	189.1 mW

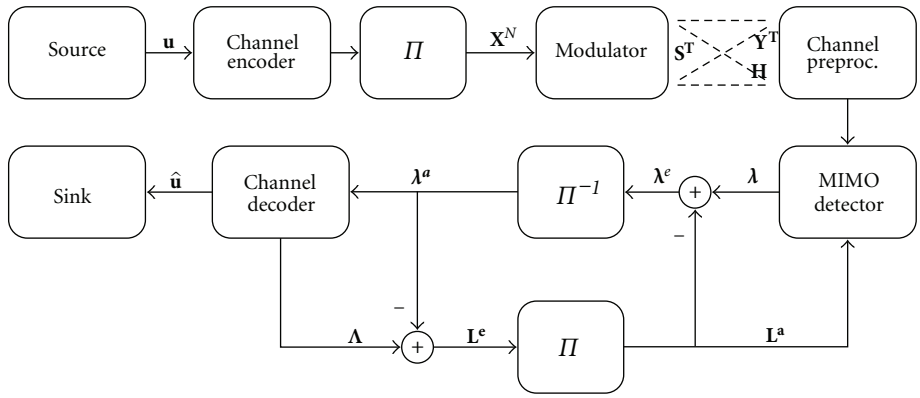


Figure 1: System model of bit interleaved coded modulation scheme with iterative MIMO detection and channel decoding in the receiver.

For all detection-related explanations, the time indices of y, H, and $\mathbf{s}$ are dropped for ease of notation. Even if not mentioned specifically for each equation, the vectors $\mathbf{s}$ and $\mathbf{x}$ are always connected via $\mathbf{s} = \text{map}(\mathbf{x})$. $x_{q,m}$ denotes the qth bit of the mth symbol in $\mathbf{s}$.

For iterative detection and decoding the MIMO detector computes logarithmic likelihood values (LLRs) on each bit

$$\lambda\left(x_{q,m}\right) = \ln \frac{P\left(x_{q,m} = +1 \mid \mathbf{y}\right)}{P\left(x_{q,m} = -1 \mid \mathbf{y}\right)}. \tag{8}$$

For independent $x_{q,m}$, the probability $P(x_{q,m} = +1 \mid \mathbf{y})$ is obtained by summing up the probabilities of all possible symbol vectors $\mathbf{s}$ which contain $x_{q,m} = +1$:

$$P\left(x_{q,m} = +1 \mid \mathbf{y}\right) = \sum_{\forall \mathbf{s} \mid x_{q,m} = +1} P(\mathbf{s} \mid \mathbf{y}). \tag{9}$$

Using *Bayes theorem*, $P(\mathbf{s} \mid \mathbf{y})$ can be expressed as

$$P(\mathbf{s} \mid \mathbf{y}) = \frac{P(\mathbf{s}) \cdot P(\mathbf{y} \mid \mathbf{s})}{P(\mathbf{y})}. \tag{10}$$

We can observe that the analyzed probability consists of three parts. $P(\mathbf{s})$ takes into account that not every $\mathbf{s}$ is equally likely given the a priori information $\mathbf{L}^a$ from the channel decoder. As the codeword is interleaved before the QAM mapping the bits $x_{q,m}$ are assumed independent from each other. Therefore, $P(\mathbf{s})$ is the product of its bits' probabilities:

$$P(\mathbf{s}) = \prod_i P\left(x_{q,m}\right). \tag{11}$$

The conditional probability $P(\mathbf{y} \mid \mathbf{s})$ illustrates how likely it is to receive the signal $\mathbf{y}$ when $\mathbf{s}$ has been sent. It equals the probability of the noise needed to receive $\mathbf{y}$ when $\mathbf{s}$ is sent over the channel $\mathbf{H}$. As the noise $\mathbf{n}$ is additive white Gaussian with variance N_0, $P(\mathbf{y} \mid \mathbf{s})$ can be written as

$$P(\mathbf{y} \mid \mathbf{s}) = P(\mathbf{n} = \mathbf{y} - H\mathbf{s}) = \frac{1}{\sqrt{2\pi}} e^{-\|\mathbf{y} - H\mathbf{s}\|^2 / N_0}. \tag{12}$$

The third part $P(\mathbf{y})$ is constant during the detection of $\mathbf{y}$ and is cancelled out when applying (10) and (12) to (8):

$$\lambda\left(x_{q,m}\right) = \ln \frac{\sum_{\forall \mathbf{s} \mid x_{q,m} = -1} P(\mathbf{s}) \cdot e^{-\|\mathbf{y} - H\mathbf{s}\|^2 / N_0}}{\sum_{\forall \mathbf{s} \mid x_{q,m} = +1} P(\mathbf{s}) \cdot e^{-\|\mathbf{y} - H\mathbf{s}\|^2 / N_0}}. \tag{13}$$

The large number of multiplications and the exponential function involved in the computation of (13) make it less attractive for implementation. Therefore, it is transformed

into the logarithmic domain where the exponential function disappears and the multiplications become additions. Hereby, a problem is posed by the additions. The *Jacobian logarithm* is used to formulate them as

$$\ln(e^x + e^y) = \max^*(x, y), \tag{14}$$

with

$$\max^*(x, y) = \max(x, y) + \ln\left(1 + e^{-|x-y|}\right). \tag{15}$$

The max*-operation can be approximated by the normal max-operation. This leads to the Max-Log-Map approximation [1]:

$$\begin{aligned}\lambda\left(x_{q,m}\right) &\approx \max_{\forall \mathbf{s}|x_{q,m}=-1}\{\ln P(\mathbf{y} \mid \mathbf{s}) + \ln P(\mathbf{s})\} \\ &\quad - \max_{\forall \mathbf{s}|x_{q,m}=+1}\{\ln P(\mathbf{y} \mid \mathbf{s}) + \ln P(\mathbf{s})\}, \\ \lambda\left(x_{q,m}\right) &\approx \max_{\forall \mathbf{s}|x_{q,m}=-1}\left\{-\frac{||\mathbf{y} - H\mathbf{s}||^2}{N_0} + \sum_{\forall q,m} \ln P\left(x_{q,m}\right)\right\} \\ &\quad - \max_{\forall \mathbf{s}|x_{q,m}=+1}\left\{-\frac{||\mathbf{y} - H\mathbf{s}||^2}{N_0} + \sum_{\forall q,m} \ln P\left(x_{q,m}\right)\right\}.\end{aligned} \tag{16}$$

Exchanging maximum by minimum operations the next equation is obtained:

$$\begin{aligned}\lambda\left(x_{q,m}\right) &\approx \min_{\forall \mathbf{s}|x_{q,m}=+1}\left\{||\mathbf{y} - H\mathbf{s}||^2 - N_0 \sum_{\forall q,m} \ln P\left(x_{q,m}\right)\right\} \\ &\quad - \min_{\forall \mathbf{s}|x_{q,m}=-1}\left\{||\mathbf{y} - H\mathbf{s}||^2 - N_0 \sum_{\forall q,m} \ln P\left(x_{q,m}\right)\right\}.\end{aligned} \tag{17}$$

An interpretation for (17) is that we derive the LLR value $\lambda(x_{q,m})$ from the most likely symbol vectors $\mathbf{s}$ with $x_{q,m}$ being +1 or −1, respectively. The metric $d(\mathbf{s})$ measures the likelihood that a specific vector $\mathbf{s}$ has been sent:

$$d(\mathbf{s}) = ||\mathbf{y} - \mathbf{H}\mathbf{s}||^2 - N_0 \sum_{\forall q,m} \ln P\left(x_{q,m}\right). \tag{18}$$

Small metrics $d(\mathbf{s})$ relate to a high probability of $\mathbf{s}$ having been sent.

Calculating all possible $d(\mathbf{s})$ to determine (17) becomes quickly infeasible for higher antenna constellations and/or higher-order modulations as the complexity grows with 2^{QM}. Therefore, many sub-optimal algorithms with lower complexity exist. Most of them are based on a tree search. In order to map the metric calculations (18) on a tree, the channel matrix $\mathbf{H}$ is decomposed into a unitary matrix $\mathbf{Q}$ and an upper-triangular matrix $\mathbf{R}$. The Euclidean distance is rewritten as

$$||\mathbf{y} - \mathbf{H}\mathbf{s}||^2 = ||\mathbf{y}' - \mathbf{R}\mathbf{s}||^2 \tag{19}$$

with $\mathbf{y}' = \mathbf{Q}^H\mathbf{y}$. Equation (18) is replaced by the equivalent metric

$$d(\mathbf{s}) = ||\mathbf{y}' - \mathbf{R}\mathbf{s}||^2 - N_0 \sum_{\forall q,m} \ln P\left(x_{q,m}\right). \tag{20}$$

The triangular structure of $\mathbf{R}$ allows the recursive calculation of $d(\mathbf{s})$

$$d_m = d_{m+1} + \gamma_m\left(s^{(m)}\right) \tag{21}$$

with the starting point $d_{M+1} = 0$ and $d(\mathbf{s}) = d_1$. The metric update $\gamma_m(s^{(m)})$ depends on the partial symbol vector $s^{(m)} = (s_m, s_{m+1}, \ldots, s_M)$:

$$\gamma_m\left(s^{(m)}\right) = \left| y'_m - \sum_{j=m}^{M} R_{m,j} s_j \right|^2 - N_0 \sum_{q=1}^{Q} \ln P\left(x_{q,m}\right). \tag{22}$$

This recursive structure can be represented by a tree with $M+1$ levels as shown in Figure 2 for the modulation alphabet $\{-1, +1\}$. The root node corresponds to d_{M+1} and each leaf node corresponds to the metric $d(\mathbf{s})$ of one possible vector $\mathbf{s}$. Each level corresponds to the detection of one symbol s_m. Branches are labeled with an element of the modulation alphabet. When advancing from a parent to a child node, the metric of the child node d_m is calculated from the metric of the parent node d_{m+1} and the branch metric γ_m.

Based on this tree search, many different MIMO detection algorithms exist. The main differences between the algorithms can be described by how they traverse the tree, for example, breadth-first, depth-first, or metric-first, and how branches of the tree are excluded. In general, those algorithms result in different communications performance and implementation complexities. In the next sections, we will present two different algorithms and show the trade-offs between them.

4.1. Sphere Detector. The sphere detector is a depth-first search which considers all symbol vectors $\mathbf{s}$ in the computation of (17) which lie inside a sphere of radius r around the received vector $\mathbf{y}$, that is, for which $d(\mathbf{s}) < r$. The radius r is determined before the search starts. The choice of the radius offers a trade-off between very good communications performance and throughput. For a high radius, many nodes are visited and the resulting communications performance is close to the optimum. For a low radius, the search is very fast but the communications performance is degraded.

During the search, the sphere detector may visit many leaf nodes but only stores the data relevant for the computation of the LLR values (17). Furthermore, sorted QR decomposition [24] and MMSE preprocessing [10] are used as additional techniques for complexity reduction.

4.2. Fixed Effort List Detector. A fixed effort detector [25] generates a list $\mathbf{L}$ of leaf nodes and their according Euclidean distances. It is based on a breadth-first search in which the number of child nodes is predetermined for every layer of the tree. Thus, the number of visited nodes is constant for one so-called *node distribution*. Typically, in the beginning

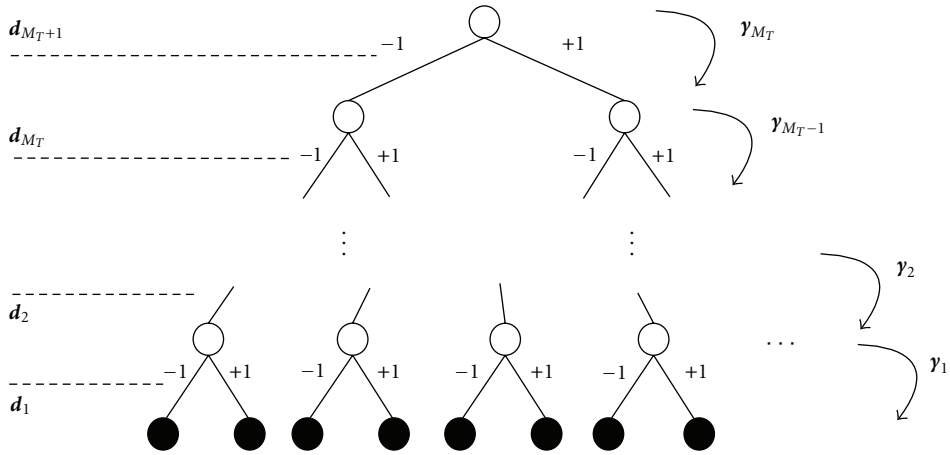


Figure 2: Detection problem represented by a tree for the modulation alphabet $\{-1, +1\}$ (BPSK) and M_T antennas.

of the tree search, many children nodes are visited while, in lower layers, only one or two nodes are expanded. Therefore, the use of a sorted QR decomposition which moves the unreliable layers to the top of the tree is mandatory [14, 24]. Each candidate in the list consists of a bit vector $\mathbf{x}$ and the corresponding Euclidean distance d_E.

In order to obtain soft-output LLRs and to be able to process a priori information, the fixed effort MIMO detector has to be followed by an LLR generator. In the LLR generator, the a posteriori LLRs are approximated by (17) but the minimum search only runs over those vectors $\mathbf{s}$ which have been stored in the list $\mathbf{L}$. Also, the Euclidean distance has been stored in $\mathbf{L}$ and does not have to be recalculated.

5. Results Communications Performance

The design space for iterative MIMO detection and channel decoding is enormous considering all the possibilities for sub-optimal algorithms, the choice of the channel code, scheduling between detector and decoder, channel and modulation parameters, and so forth. Covering all these possibilities is out of scope of this paper. Therefore, we introduce the following restrictions on the design space. As channel code we employ a WiFi compliant 64-state non-systematic, nonrecursive convolutional code. The decoding of convolutional codes is noniterative thus removing the scheduling problem between inner and outer iterations. We use code rate 1/2 and code words of 2304 bits. This code length has been chosen to allow a comparison with existing LDPC codes of the WiMax, WiFi standards [26]. This in-depth comparison will be done in a future publication. The channel is modeled as Rayleigh fading with 4 transmit and 4 receive antennas.

As a first step of the design space exploration we compare the communications performance for two different MIMO detection algorithms, namely, the sphere detector and the fixed effort detector from Section 4. The two algorithms offer a trade-off between hardware efficiency and communications efficiency. Two modulation schemes are compared—16-QAM and 64-QAM—which pose different requirements to the MIMO detector in terms of complexity.

Figures 3 and 4 show the communications performance results for the two algorithms for 4×4 antennas, 16-QAM and 64-QAM, respectively. The frame error rate is measured after the convolutional decoder. The red curves show the results of the close-to-optimum sphere detector. The green and blue curves stem from the fixed effort detector with different list sizes L. We limited the number of outer iterations to 3. Currently, this is the highest number of iterations we assume in a hardware realization since the throughput will linearly decrease with the iterations. Anyway, additional iterations will not result in a further significant gain in communications performance [1, 2].

In both figures, we observe a similar behaviour of the different algorithms. Both, the sphere detector and the fixed effort detector have their largest gain within the first iteration (up to 4 dB for the sphere detector and around 3 dB for the fixed effort detector). Furthermore, the communications performance of the fixed effort detector depends significantly on the list size L. Particularly for small list sizes (green curves), more than one iteration does not significantly improve the performance anymore. Whereas the difference between small (green) and big list sizes (blue curves) is small in iteration 0, it is well known that, for the larger list sizes, the communications performance is better in successive iterations. When an extremely large list is adopted (e.g., 1024 for 16-QAM and 4096 for 64-QAM), the performance of the fixed effort list detector approaches the soft-output depth-first sphere detector.

Recapitulatory, the most important observations are listed in the following. After iteration 0, fixed effort and sphere detector based MIMO detection obtain a similar communications performance. Both achieve the biggest gain within the first iteration. The communications performance of the fixed effort detector depends heavily on the list size. For small list sizes, no more than one iteration is beneficial as the decoding process "gets stuck," that is, does not further

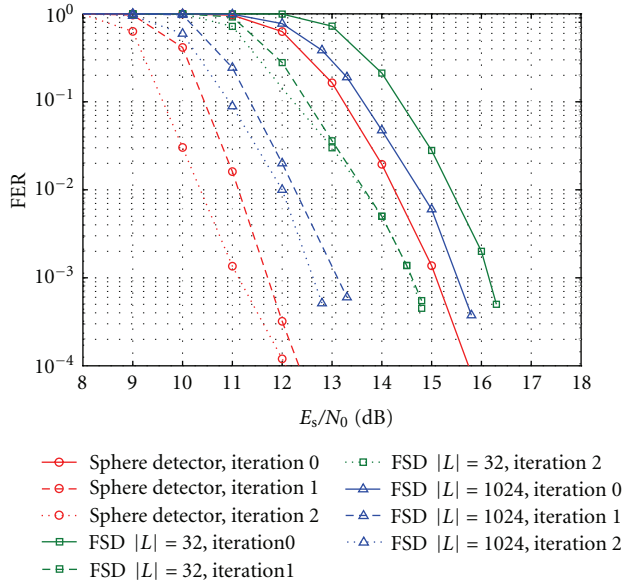


Figure 3: Communications performance of 4×4 antennas system, 16-QAM modulation for different MIMO detection algorithms.

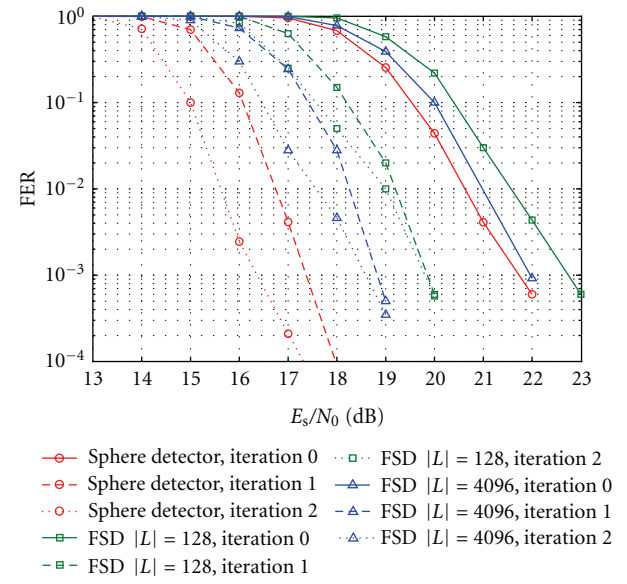


Figure 4: Communications performance of 4×4 antennas system, 64-QAM modulation for different MIMO detection algorithms.

improve. The best communications performance is achieved by sphere detection with several outer iterations.

6. Results VLSI Components

In this section, we will present the architectures and implementation results of the different VLSI components which will be combined and analyzed as an iterative receiver in Section 7.

All designs were synthesized in a 65 nm low-power bulk CMOS standard cell library. Target frequency after place & route is 300 MHz which is typically used for industrial designs. In order to ensure 300 MHz after place & route, synthesis was done with a target frequency of 360 MHz. We considered the following PVT parameters: Worst Case (WC, 1.1 V, 125°C), Nominal Case (NOM, 1.2 V, 25°C) and Best Case (BC, 1.3 V, −40°C). Synthesis was performed with Synopsis Design Compiler in topographical mode, place & route (P&R) with Synopsys IC Compiler. Synthesis as well as P&R were performed with Worst Case PVT settings of the 65 nm library.

6.1. QR Decomposition. From the bunch of existing algorithms, we chose the modified Gram-Schmidt process [27] to compute the QR decomposition due to its simplicity and stability when working with finite precision values. Input and output matrices are quantized with 12 bits for real and imaginary values, respectively. It has been shown that this quantization yields only a minor degradation in system communications performance [28]. The resulting architecture runs a sorted QR decomposition with MMSE preprocessing for a 4×4 channel matrix in 167 clock cycles. After P&R it has an area of 0.14 mm^2 and consumes a power of 12.0 mW in nominal case when running at 300 MHz.

6.2. Convolutional Decoder. In open-loop systems, convolutional codes can be decoded with the Viterbi algorithm [29] which provides the ML solution, that is, a sequence of hard output bits. In closed-loop MIMO systems, however, soft-output LLR values of the whole codeword are needed for the outer iterations. Thus, the BCJR algorithm [30] has to be applied to obtain the soft-output MAP solutions. Input and output LLR values are quantized with 6 bits each.

State-of-the art convolutional decoders process 1 bit per clock cycle. Consequently, they obtain a throughput of 300 Mbit/s at a clock frequency of 300 MHz. In [31], a 65 nm technology Viterbi decoder design has been presented which is able to run at a clock frequency of more than 300 MHz. It consumes an area of 0.11 mm^2 and has a power consumption of approximately 40 mW.

Implementations of the BCJR-algorithm for 64-state convolutional codes are not widely available in the literature. Therefore, we chose the 180 nm technology decoder design from [32]. We scaled the original implementation data down to 65 nm technology yielding an area of 0.31 mm^2 and a power consumption of approximately 240 mW (area scaling factor: $65^2/180^2$, power scaling factor: $65^{1.5}/180^{1.5}$).

6.3. Sphere Detector. The tree search for sphere detection can be separated into five basic operations: computing the interference reduced symbol, enumerating the most promising children nodes, computing the metrics, processing the results of the leaf nodes and storing intermediate results and choosing the next node. In the presented sphere decoder architecture, each of these operations has been implemented in a separate block, see Figure 5. The enumeration unit performs the enumeration of children nodes either based

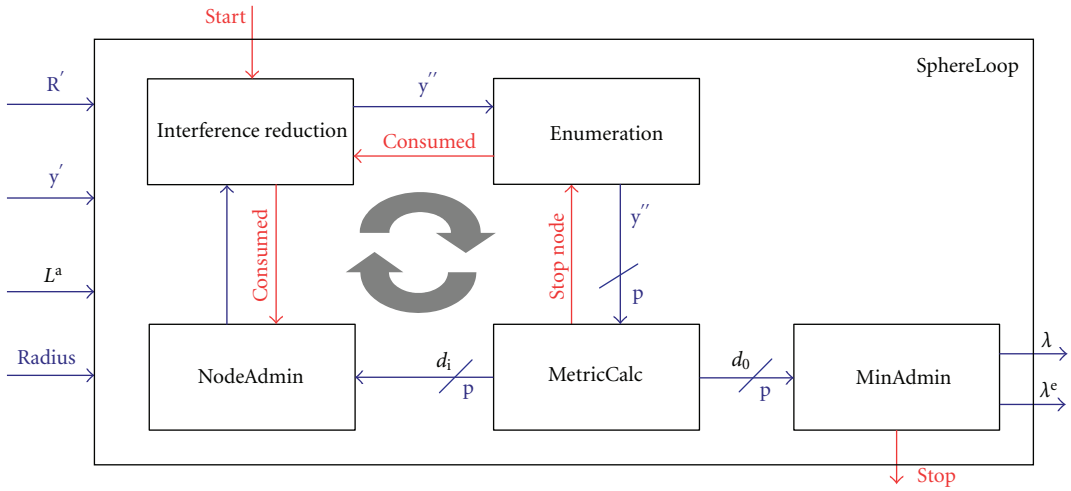


Figure 5: Sphere detector architecture.

Table 2: Implementation results of the sphere detector architecture after place & route for a clock frequency of 300 MHz.

At 300 MHz	Sphere Detector
Modulation	up to 64-QAM
Antennas	up to 4×4
Area	0.26 mm^2
Throughput	38–58 Mbit/s
Power consumption	15 mW

Table 3: Implementation results for the components of the fixed effort list detector architecture after place & route for a clock frequency of 300 MHz.

At 300 MHz	Fixed effort detector		LLR-unit	
Modulation	16-QAM	64-QAM	16-QAM	64-QAM
List size	32	128	32	128
Area	0.36 mm^2		0.14 mm^2	
Throughput	267 Mbit/s	109 Mbit/s	141 Mbit/s	55 Mbit/s
Power	103 mW	118 mW	21 mW	31 mW

on the interference reduced symbol or based on the a priori information.

The presented architecture computes two nodes per cycle in contrast to other depth-first sphere decoders (e.g., [3, 5]) which employ a one-node-per-cycle architecture. This is a new approach which doubles the throughput compared to state-of-the-art implementations. Its detailed architecture will be presented in a future publication since this paper deals with system analysis and the trade-off between communications performance versus implementation performance. The sphere detector works with antenna systems up to 4×4 antennas and QAM modulation schemes up to 64-QAM. During run-time, throughput can be traded off against communications performance by adjusting the radius. However, due to the nature of the depth-first search, the throughput is dynamic and varies with the channel conditions and the outer iterations. After place & route, the design has an area of 0.26 mm^2 and a power consumption of only 15 mW. The implementation data is summarized in Table 2.

6.4. Fixed Effort List Detector. The architecture of the fixed effort list detector supports 16-QAM and 64-QAM modulation. The list size is configurable to be 32 and 128 for 16-QAM and 64-QAM, respectively. It consists of a list generator (employing the fixed effort detection algorithm) and an individual LLR generator to generate soft-outputs, as shown in Figure 6.

The list generator is implemented by an eight-nodes-per-cycle parallel architecture, which processes 8 nodes in each clock cycle concurrently as a group, with the breadth-first tree search order. Eight identical units are employed for each of the main arithmetic tasks, such as enumeration and metric calculation. After the tree search, a candidate list is sent to the LLR generator, which receives the a priori data from channel decoder and computes the extrinsic data. The LLR generator is also implemented with highly parallel architecture. The throughput of both, the list generator and the LLR generator, depends highly on the list size. Implementation results after place & route are summarized in Table 3.

7. System Analysis

In this section we will investigate the cost for practical applications with respect to throughput, area, and power. Therefore, we first introduce a generic architecture framework supporting different MIMO detectors and channel decoders. After presenting each building block individually in the last section, we will analyze different aspects of the components regarding the complete iterative system. The major problem of MIMO iterative systems with the overall design decisions is the dynamic constraints for throughput

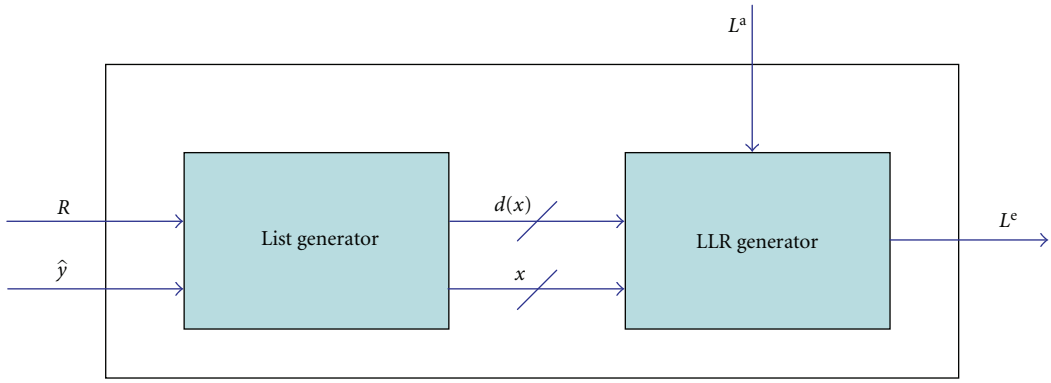


Figure 6: Fixed Effort list detector architecture.

and communications performance in different application scenarios. Thus, we will compare the sphere detector and fixed effort detector for different scenarios and SNR ranges. Eventually, we analyze the difference in implementation costs for open- and closed-loop systems.

7.1. Architecture Framework. We have mapped the iterative receiver structure from Figure 1 onto a general architecture framework which allows to plug in different MIMO detectors and channel decoders. The framework—shown in Figure 7—connects the main building blocks via several system memories. The area for each memory is shown in Figure 7. The total area of all system memories is 0.271 mm^2.

The iterative receiver structure from Figure 1 is mapped onto this generic framework. During the inner iterations of the channel decoder the values in DEC_IN might be updated. Thus, the original information is not on hand after decoding. The a posteriori LLR values $\boldsymbol{\lambda}$ have to be stored in DET_OUT in order to be able to extract the extrinsic information $\mathbf{L}^a$ for the next iteration of the detector. Interleaver and Deinterleaver tables are stored in INT and DE_INT and are read by interleaver unit Π and deinterleaver unit Π^{-1}, respectively. We assume that all complex values require 12 bits for real and imaginary part, respectively, and that all LLR values are quantized with 6 bits.

In the further analysis, we distinguish between the open-loop system without feedback between channel decoder and MIMO detector and the closed-loop with feedback. In closed-loop systems, all memories are mandatorily required. When the MIMO detector is processing a codeword, the decoder has to wait until it is finished and vice versa. Thus, MIMO detector and channel decoder are never active at the same time.

For an open-loop system, the architecture framework can be simplified. First of all, the memories related to the feedback loop—DET_OUT, DET_IN, and INT—are obviously not needed. But in addition, the QR decomposition can provide the data as needed for the MIMO detector so the memories Y_HAT and MAT_R are not required. While the channel decoder is working on one codeword the MIMO detector can already start the next one. In this way, MIMO detector and channel decoder can both be active at all times. The only additional requirement to enable full activity is the doubling of DEC_IN. In summary, in open-loop systems we need an area of 0.123 mm^2 for system memories and in closed-loop systems we need 0.271 mm^2.

7.2. Components in the System. In Section 6 the VLSI building blocks were introduced without any system considerations. In the following paragraphs, we will look at the dependencies between throughput, communications performance, and different system parameters for each component and what are requirements on the components in open-loop and closed-loop receivers. The observations from the next paragraphs are also summarized in Table 4. The units are shown in columns next to each other giving a good overview of individual design problems, throughputs, and constraints.

QR Decomposition. The presented design for QR decomposition processes matrices for 2×2 or 4×4 antennas including the sorting of layers and MMSE preprocessing. For 4×4 matrices, the unit processes $1.8 \cdot 10^6$ matrices per second consuming 6.68 nJ per matrix. Under the assumption of a truly ergodic channel, that is, the channel changes independently after each use, this relates to 28.8 Mbit/s for 16-QAM, or 43.2 Mbit/s for 64-QAM. In contrast to the MIMO detector, a higher constellation size is beneficial for the bit throughput of the QR decomposition because the processing time depends only on the size of the matrix. In a realistic channel, it is expected that the channel will stay constant for several channel uses. In this case, the QR decomposition only has to be done once for several MIMO vectors and the bit throughput increases. For the QR decomposition there is no difference between open-loop and closed-loop systems as the channel preprocessing is only done once for every channel matrix.

Sphere Detector. The sphere detector architecture detects MIMO vectors for systems with up to 4×4 antennas and QAM modulation schemes up to 64-QAM. Throughput and communications performance depend mainly on the number of visited nodes during the tree search. The sphere radius offers a good trade-off parameter which regulates the number of nodes which can be visited. For a low radius, a

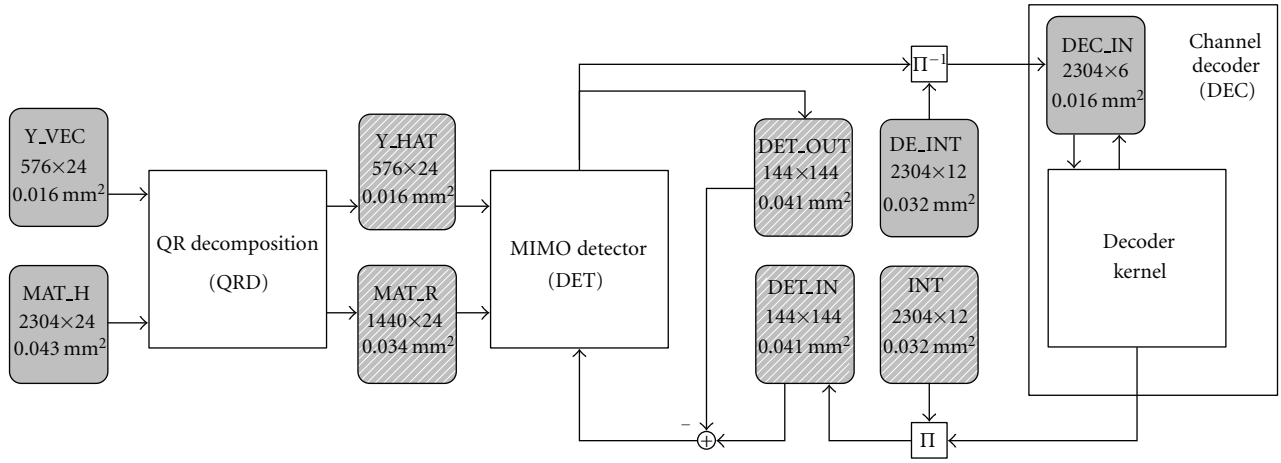


Figure 7: Generic architecture framework including main building blocks and system memories. In open-loop systems, the diagonally hatched memories are not needed while DEC_IN has to be doubled.

Table 4: Design overview for individual components in open-loop and closed-loop systems. Showing them in columns next to each other gives a good overview of individual design problems, throughputs, and constraints even if they are not put in a system yet.

Component	QR decomposition	MIMO sphere	MIMO fixed effort	Convolutional decoder
Flexibility	2×2 or 4×4 matrices	up to 4×4 antennas, up to 64-QAM	4×4 antennas, 16-QAM or 64-QAM	code rates 0.5–1
Throughput depends on	Number of antennas, sorting, MMSE or zero-forcing	Modulation, number of antennas, radius	Modulation, number of antennas, list size	Constant
Communications performance depends on	MMSE/zero-forcing, sorted/unsorted	radius	list size	—
Throughput range (4×4, 64-QAM)	≥43 Mbit/s (ergodic)	38–58 Mbit/s	109 Mbit/s (fixed effort det.), 55 Mbit/s (LLR)	300 Mbit/s
Open loop		Best communications performance	Good communications performance	Low complexity Viterbi algorithm with hard output
		Dynamic throughput over SNR	List storage not required	
Area	0.14 mm² (P&R)	0.26 mm² (P&R)	0.36 + 0.14 mm² (P&R)	0.11 mm² [31]
	+0.06 mm² memories	+0.032 mm² memories	+0.032 mm² memories	+0.032 mm² memories
Closed loop	No further processing necessary	Best communications performance	Gets stuck after 2nd iteration	BCJR algorithm with soft-output of the parity information
		Dynamic throughput over iterations	For one feedback loop good throughput	
Area	0.14 mm² (P&R)	0.26 mm² (P&R)	0.36 + 0.14 mm² (P&R)	0.31 mm² [32]
	+0.11 mm² memories	+0.146 mm² memories	+0.146 mm² memories	+0.016 mm² memories
			+0.32 mm² list storage	

high throughput is obtained at the cost of a reduced communications performance and vice versa for a high radius.

Particularly for iterative receivers, the sphere detector offers the best communications performance possible. Due to the depth-first search strategy, the processing time for one MIMO vector is not constant. In fact, it depends on the SNR of the current channel realization. So even for one SNR value, the throughput varies for different MIMO vectors. Generally, the number of nodes will decrease for higher SNR values. The throughput also changes over the outer iterations. This is problematic when a worst case throughput has to be ensured. Otherwise, there are no changes within the architecture for open- or closed-loop systems.

Fixed Effort List Detector. The fixed effort detector architecture is optimized for 4×4 antenna systems with two node distributions for 16-QAM and 64-QAM, respectively. This results in list sizes of 32 or 128 entries. The following LLR

generator is able to work with list sizes up to 128 entries. The node distributions determine the number of nodes which will be visited for one MIMO vector. The choice of the node distribution, however, varies according to the number of antennas, the modulation scheme, and the required list size. The communications performance of the fixed effort detector is directly influenced by the list size. For small list sizes, iterative detection and decoding obtain no more gain after the first iteration. Furthermore, it is mandatory to use a sorted QR decomposition which moves the least reliable layers to the top of the tree. Otherwise, the communications performance drops by several dB. In open-loop processing, the list which is generated in the FSD can be directly used as input for the LLR generator. List storage is not required. Like for the sphere detector, the memories DET_OUT, DET_IN, and INT are not needed. When moving to closed-loop receivers, the lists of all MIMO vectors have to be stored to be reused in the next iterations. The required memory is determined by the 64-QAM case with a list size of 128. For the whole block consisting of 2304 bits, 12288 list entries with 36 bits are needed. The resulting memory consumes approximately 0.32 mm^2. This shows already why bigger lists will not be feasible because already for a list size of 128 the list storage consumes almost the same area as the fixed effort detector core itself.

Convolutional Decoder. The chosen architecture for convolutional decoding processes all code rates ≥ 0.5. The throughput is fixed to 300 Mbit/s by the choice of the architecture independent of the code rate. In the open-loop system, no feedback information is required, thus hard-output bits of the information word are sufficient. In this case, the low-complexity Viterbi algorithm can be chosen which finds the optimal maximum likelihood (ML) solution. In the closed loop, however, soft-output LLR values of the complete codeword are needed as feedback for the MIMO detector. This requires an extended version of the BCJR algorithm which also produces LLR values of the parity information. The introduction of the BCJR algorithm increases the decoder area from approximately 0.11 mm^2 to 0.31 mm^2.

7.3. Scenario Analysis. In most publications, MIMO detectors are analyzed as an individual building block. However, the major problem of iterative MIMO systems are the dynamics of different system scenarios, for example, different throughput and communications performance requirements. The argumentation for one specific architecture is often misleading if it is only based on one specific scenario. Depending on quality of service or throughput requirements, different detection strategies will have advantages. Arguments for a specific realization can be reversed when changing the required flexibility or the multiplexing scheme.

In this section, we will analyze and compare sphere detector and fixed effort list detector in different scenarios. One part of the scenarios will be communication centric, that is, what is the cost to reach a certain frame error rate at a certain signal-to-noise ratio. Other scenarios concentrate on throughput exploring hardware units and power consumption in order to reach a certain throughput. The scenarios combined with the summarized result data are shown in Table 5. Typically, worst case constraints in systems are for the highest antenna/modulation system. Thus only in the 4×4 antennas, 64-QAM case is shown within the presented system examples. For the fixed effort list detector architecture two LLR units are employed to balance the throughput between list generation and LLR generation.

For all scenarios, it is assumed that the channel decoder processes one bit per clock cycle resulting in a throughput of 300 Mbit/s. This is a typical assumption for state-of-the-art convolutional decoder architectures. While the throughput of the channel decoder is fixed, the throughputs of the MIMO detectors vary depending on the chosen scenario leading to an unbalanced processing time for MIMO detection and channel decoding. In open-loop systems, MIMO detector and channel decoder work as a pipeline. The system throughput is determined by the component with the lowest throughput only, typically the MIMO detector.

For closed-loop systems, there are two alternatives. Either two code words are processed in parallel—one in the MIMO detector and one in the channel decoder—or only one codeword is processed at a time. Working on the same codeword in parallel is prevented by the channel interleaver because detector and decoder always have to wait until the other one has finished processing the whole codeword. In the first case all system memories have to be doubled to store the data of the two code words. Furthermore, for unbalanced processing the throughput is still determined by the slower component whereas the faster component is idle for the rest of the time. On the other hand, if only one code word is handled by the iterative receiver, every component has to wait until the other one has finished the current code word but the memories are not impacted. The system throughput T_{sys} in this case depends on the throughputs of MIMO detector T_{mimo} and channel decoder T_{dec} and the number of outer iterations iter (starting at 0) in the following nonlinear way:

$$\frac{1}{T_{\text{sys}} \cdot (\text{iter} + 1)} = \frac{1}{T_{\text{mimo}}} + \frac{1}{T_{\text{dec}}}. \tag{23}$$

The system throughput decreases linearly with the number of iterations. As the throughputs of the MIMO detectors largely vary for the different scenarios we chose the second case for our analysis; that is, only one codeword is processed at a time.

The scenarios in Table 5 either target a system frame-error rate of 10^{-3} at different signal-to-noise ratios or specific system throughputs ranging from 30 Mbit/s up to 300 Mbit/s. In the *communication centric scenarios*, the current architecture of the fixed effort list detector is able to achieve the target frame-error rate for the two highest SNR values at a good system throughput of 110 Mbit/s for open loop and 40 Mbit/s for closed loop. The average power consumption decreases for closed-loop systems because the list generator only runs in iteration 0. In the following iterations, only list storage and LLR unit are active. Theoretically, the fixed effort list detector can reach the frame-error rate of 10^{-3} at 18 dB with a list of size 4096 as shown in Figure 4. In that case, the list storage would increase by a factor of 128 to approximately 10.2 mm^2. The processing units would scale

Table 5: System perspective constraints for different scenarios for 4 × 4 antenna, 64-QAM systems. The resulting throughput, area, power, and communications performance are very dynamic. Two different types of scenarios are analyzed: communications centric and throughput centric. The fixed effort list detector consists of one fixed effort detector and two LLR units.

Scenario description	MIMO detector components	Detector throughput	Detector area	Power consumption detector	System throughput
Fixed Effort List Detector—Communication Centric					
FER = 10^{-3} at SNR = 22 dB	1 × Fixed Effort List Det., 0 iterations	110 Mbit/s	0.64 mm^2	180 mW	110 Mbit/s
FER = 10^{-3} at SNR = 20 dB	1 × Fixed Effort List Det. + list storage, 1 iteration	110 Mbit/s	0.96 mm^2	153 mW	40 Mbit/s
FER = 10^{-3} at SNR = 18 dB	theoretically with list size 4096, *not adequate*				
FER = 10^{-3} at SNR = 16 dB	not possible	—	—	—	—
Sphere Detector—Communication Centric					
FER = 10^{-3} at SNR = 22 dB	1 × Sphere Detector, 0 iterations	38 Mbit/s	0.26 mm^2	15 mW	38 Mbit/s
FER = 10^{-3} at SNR = 20 dB	1 × Sphere Detector, 1 iteration	58 Mbit/s	0.26 mm^2	15 mW	24 Mbit/s
FER = 10^{-3} at SNR = 18 dB	1 × Sphere Detector, 1 iteration		0.26 mm^2	15 mW	
FER = 10^{-3} at SNR = 16 dB	1 × Sphere Detector, 2 iterations	*Achievable but T very low (not adequate)*			
Fixed Effort List Detector—Throughput Centric					
T = 300 Mbit/s, 0 iterations	3 × Fixed Effort List Det.	330 Mbit/s	1.92 mm^2	540 mW	300 Mbit/s
T = 100 Mbit/s, 0 iterations	1 × Fixed Effort List Det.	110 Mbit/s	0.64 mm^2	180 mW	110 Mbit/s
T = 100 Mbit/s, 1 iteration	6 × Fixed Effort List Det. + list storage	660 Mbit/s	4.16 mm^2	918 mW	103 Mbit/s
T = 30 Mbit/s, 1 iteration	1 × Fixed Effort List Det. + list storage	110 Mbit/s	0.96 mm^2	153 mW	40 Mbit/s
Sphere Detector—Throughput Centric					
T = 300 Mbit/s, 0 iterations	8 × Sphere Detector	304 Mbit/s	2.08 mm^2	120 mW	300 Mbit/s
T = 100 Mbit/s, 0 iterations	3 × Sphere Detector	114 Mbit/s	0.78 mm^2	45 mW	114 Mbit/s
T = 30 Mbit/s, 0 iterations	1 × Sphere Detector	38 Mbit/s	0.26 mm^2	15 mW	38 Mbit/s
T = 30 Mbit/s, 1 iteration	2 × Sphere Detector	76 Mbit/s	0.52 mm^2	30 mW	30 Mbit/s

by a similar factor depending on the targeted throughput. Therefore, a list size of 4096 is not feasible.

The sphere detector is able to reach the target communications performance for all given signal-to-noise ratios with up to two iterations. However, the throughput is much lower than for the fixed effort detector. At 20 dB the radius can be lowered to increase the throughput as a frame-error rate of 10^{-3} is achieved easily. At 16 dB, 2 outer iterations are necessary heavily reducing the throughput to where it is not adequate anymore.

In the *throughput centric* scenarios, we analyze which parallelism is needed for the MIMO detector to reach a certain system throughput. For open-loop systems, the system throughput linearly increases with the number of detector instantiations. For an open-loop throughput of 300 Mbit/s, three fixed effort list detector instances or eight sphere detector instances are needed. Even though the MIMO detectors have a throughput higher than 300 Mbit/s, the system throughput is in this case limited by the channel decoder running at a constant throughput of 300 Mbit/s. For most throughput centric scenarios, the resulting area for both detectors are similar. The power consumption, however, for the sphere detector is much lower. This can be explained by the additional power needed for the list storage and the LLR units on one hand. Furthermore, the fixed effort detector architecture processes eight different nodes in parallel whereas the sphere detector is only working on two nodes in parallel which are siblings in the tree.

In summary, the fixed effort list detector is advantageous if a high throughput has to be guaranteed at a reasonable communications performance. However, best communications performance cannot be achieved because the required higher list sizes would imply infeasibly huge list storage memories. The depth-first sphere detector achieves best communications performance. With multiple instances, the sphere detector achieves a high throughput at a decent area and very good energy efficiency.

7.4. Open-Loop versus Closed-Loop Considerations. After comparing sphere detector and fixed effort detector for different application scenarios, we will now look at the effect on the whole system when moving from an open-loop implementation to a closed-loop implementation. For this analysis, we set the detector throughput to 300 Mbit/s balancing the throughput between MIMO detector and channel decoder.

The power consumption of the system memories does not depend on a specific detector architecture but only on the MIMO detector throughput. Based on the number of accesses (e.g., 4 read accesses on Y_HAT per detection),

Table 6: Difference in implementation cost between an open-loop and a closed-loop system. Area and energy efficiency drop by more than a factor of 2 for the iterative system.

	Open-loop system, 0 iterations		Closed-loop system, 1 iteration	
Employed MIMO detector	Sphere detector	Fixed effort detector	Sphere detector	Fixed effort detector
System throughput	300 Mbit/s	300 Mbit/s	75 Mbit/s	75 Mbit/s
Total system area	2.5 mm^2	2.3 mm^2	2.8 mm^2	3.6 mm^2
Total system power	180 mW	520 mW	195 mW	365 mW
System area efficiency	120 (Mbit/s)/mm^2	130 (Mbit/s)/mm^2	27 (Mbit/s)/mm^2	21 (Mbit/s)/mm^2
System energy efficiency	1.7 bit/nJ	0.6 bit/nJ	0.4 bit/nJ	0.2 bit/nJ

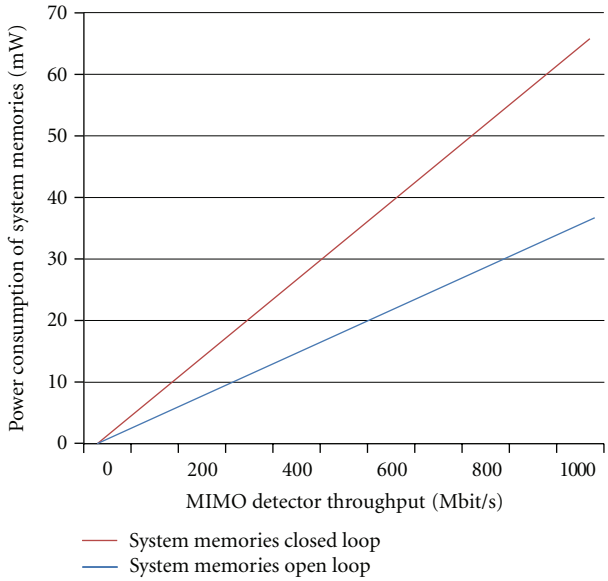


Figure 8: Power consumption of system memories depending on the MIMO detector throughput.

we determined the average power for each memory (see Figure 8). The power consumption of the memories for closed-loop decoding is approximately twice as high as in open-loop decoding. This stems from the fact that certain system memories are not needed in open-loop decoding (see Section 7.1). The implementation data of channel preprocessing and channel decoder have been summarized in Table 4.

Table 6 shows the main characteristics of the resulting open- and closed-loop *systems* employing sphere detector or fixed effort detector, respectively. We determine area and energy efficiency according to [31]. Higher numbers represent a higher efficiency. The throughput of the closed-loop system drops by a factor of 4 because only one codeword is processed at a time. This scheduling has a positive effect on the power consumption as each component is only active 50% of the time. The gain in communications performance by the outer iteration is between 3 and 4 dB. However, it can be observed that area and energy efficiency do not decrease by a factor of 2 as might be expected. In fact, the efficiency of the closed loop-system drops by factors between 3 and 6 compared to the open-loop system.

8. Conclusions

Multiple-antenna systems offer an increased bandwidth efficiency compared to single-antenna systems. Iterative receivers which exchange reliability information between MIMO detector and channel decoder will become mandatory in the near future. Choosing the MIMO detector algorithm and architecture from one of the various existing approaches has big effects on the complete system. In this paper, we have compared the depth-first variable throughput sphere detector to the breadth-first fixed effort detector in communication centric and throughput centric application scenarios. The fixed effort detector is advantageous if a high throughput has to be ensured at moderate communications performance. However, it has been observed that the sphere detector shows excellent behaviour for one outer iteration. Even with multiple instances, it obtains a decent area and very good energy efficiency.

Furthermore, we have presented an analysis of all components of the iterative receiver including channel preprocessing, MIMO detection, channel decoding, and system memories. We have shown that area and power efficiency decrease by more than a factor of 2 when changing from an open-loop decoder implementation to a closed-loop decoder employing 1 iteration independent of the choice of the MIMO detector.

Acknowledgment

The authors thank Christian Weis for his extensive help with the synthesis and place & route workflow.

References

[1] B. M. Hochwald and S. Ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, 2003.

[2] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 2299–2311, 2004.

[3] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI Implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, 2005.

[4] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: algorithms and VLSI implementation," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 290–300, 2008.

[5] E. M. Witte, F. Borlenghi, G. Ascheid, R. Leupers, and H. Meyr, "A scalable VLSI architecture for soft-input soft-output single tree-search sphere decoding," *IEEE Transactions on Circuits and Systems II: express Briefs*, vol. 57, no. 9, Article ID 5570931, pp. 706–710, 2010.

[6] C.-H. Liao, T.-P. Wang, and T.-D. Chiueh, "A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 2, Article ID 5405138, pp. 411–421, 2010.

[7] L. Liu, F. Ye, X. Ma, T. Zhang, and J. Ren, "A 1.1-Gb/s 115-pJ/bit configurable MIMO detector using 0.13-muhboxm CMOS technology," *IEEE Transactions on Circuits and Systems II*, vol. 57, no. 9, pp. 701–705, 2010.

[8] C. Studer, S. Fateh, and D. Seethaler, "A 757 Mb/s 1.5 mm^2 90 nm CMOS soft-input soft-output MIMO detector for IEEE 802.11n," in *Proceedings of the 36th European Solid State Circuits Conference (ESSCIRC'10)*, pp. 530–533, Seville, Spain, September 2010.

[9] D. Garrett, L. Davis, S. Ten Brink, B. Hochwald, and G. Knagge, "Silicon complexity for maximum likelihood MIMO detection using spherical decoding," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1544–1552, 2004.

[10] D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition,," in *Proceedings of the IEEE 58th Vehicular Technology Conference (VTC'03)*, vol. 1, pp. 508–512, October 2003.

[11] G. J. Foschini, G. D. Golden, R. A. Valenzuela, and P. W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 11, pp. 1841–1852, 1999.

[12] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best Sphere decoding for MIMO detection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 491–503, 2006.

[13] K. W. Wong, C. Y. Tsui, R. S. K. Cheng, and W. H. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'02)*, vol. 3, pp. III-273–III-276, May 2002.

[14] L. G. Barbero and J. S. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, Article ID 4543065, pp. 2131–2142, 2008.

[15] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.

[16] B. Mennenga and G. Fettweis, "Search sequence determination for tree search based detection algorithms," in *Proceedings of the IEEE Sarnoff Symposium (SARNOFF'09)*, pp. 1–6, April 2009.

[17] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI Implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1576, 2005.

[18] M. Wenk, A. Burg, M. Zellweger, C. Studer, and W. Fichtner, "VLSI implementation of the list sphere algorithm," in *Proceedings of the 24th Norchip Conference*, pp. 107–110, Linkoping, Sweden, November 2006.

[19] M. T. Gamba and G. Masera, "Look-ahead sphere decoding: algorithm and VLSI architecture," *IET Communications*, vol. 5, no. 9, pp. 1275–1285, 2011.

[20] C. Studer and H. Bölcskei, "Soft-input soft-output sphere decoding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT'08)*, pp. 2007–2011, Toronto, Canada, July 2008.

[21] M. Wenk, L. Bruderer, A. Burg, and C. Studer, "Area- and throughput-optimized VLSI architecture of sphere decoding," in *Proceedings of the 18th IEEE/IFIP International Conference on VLSI and System-on-Chip (VLSI-SoC'10)*, pp. 189–194, Madrid, Spain, September 2010.

[22] S. Chen and T. Zhang, "Low power soft-output signal detector design for wireless MIMO communication systems," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED'07)*, pp. 232–237, 2007.

[23] S. A. Laraway and B. Farhang-Boroujeny, "Implementation of a Markov chain Monte Carlo based multiuser/MIMO detector," *IEEE Transactions on Circuits and Systems I*, vol. 56, no. 1, pp. 246–255, 2009.

[24] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. D. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electronics Letters*, vol. 37, no. 22, pp. 1348–1350, 2001.

[25] L. G. Barbero and J. S. Thompson, "Extending a fixed-complexity sphere decoder to obtain likelihood information for turbo-MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 2804–2814, 2008.

[26] I. 802.16, Local and metropolitan area networks; Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems; Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands.

[27] G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, London, UK, 3rd edition, 1996.

[28] G. L. Nazar, C. Gimmler, and N. Wehn, "Implementation comparisons of the QR decomposition for MIMO detection," in *Proceedings of the 23rd Symposium on Integrated Circuits and Systems Design (SBCCI'10)*, pp. 210–214, September 2010.

[29] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[30] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. IT-20, no. 2, pp. 284–287, 1974.

[31] F. Kienle, N. Wehn, and H. Meyr, "On complexity, energy and implementation-efficiency of channel decoders," *IEEE Transactions on Communications*, vol. 59, no. 12, pp. 3301–3310, 2011.

[32] C. Studer, *Iterative MIMO decoding: algorithms and VLSI implementation aspects*, Ph.D. dissertation, ETH Zürich, Zurich, Switzerland, 2009.