

An LDPC Decoder Architecture for Wireless Sensor Network Applications

*Original*

An LDPC Decoder Architecture for Wireless Sensor Network Applications / Biroli, A.D.G., Martina, M., Masera, G.. - In: SENSORS. - ISSN 1424-8220. - ELETTRONICO. - 12:2(2012), pp. 1529-1543. [10.3390/s120201529]

*Availability:*

This version is available at: 11583/2487047 since:

*Publisher:*

MDPI Publishing

*Published*

DOI:10.3390/s120201529

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

Article

## **An LDPC Decoder Architecture for Wireless Sensor Network Applications**

**Andrea Dario Giancarlo Birolì, Maurizio Martina \* and Guido Masera**

Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino 10129, Italy; E-Mails: andreadariogiancarlo.birolì@studenti.polito.it (A.D.G.B.); guido.masera@polito.it (G.M.)

\* Author to whom correspondence should be addressed; E-Mail: maurizio.martina@polito.it; Tel.: +39-011-090-4205; Fax: +39-011-090-4217.

*Received: 18 November 2011; in revised form: 25 January 2012 / Accepted: 31 January 2012 / Published: 6 February 2012*

---

**Abstract:** The pervasive use of wireless sensors in a growing spectrum of human activities reinforces the need for devices with low energy dissipation. In this work, coded communication between a couple of wireless sensor devices is considered as a method to reduce the dissipated energy per transmitted bit with respect to uncoded communication. Different Low Density Parity Check (LDPC) codes are considered to this purpose and post layout results are shown for a low-area low-energy decoder, which offers percentage energy savings with respect to the uncoded solution in the range of 40%–80%, depending on considered environment, distance and bit error rate.

**Keywords:** LDPC decoder architecture; wireless sensor networks; power consumption

---

### **1. Introduction**

Wireless Sensor Networks (WSN) have gained growing research interest in the last years. The possibility to monitor different physical quantities even in dangerous and hard-to-reach areas has found applications in several fields, including medical, industrial and surveillance environments [1]. WSNs are made of small nodes, where each node often relies on small size and light weight batteries. As a consequence, both energy consumption and area occupation are important aspects in the design of nodes. Although nodes feature a limited energy budget, they embody not only sensing but also computational

and transmit/receive circuits. Thus, energy consumption issues are critical and ought to be minimized at every design level. As an example in [2] several system level techniques, including modulation, Media Access Control (MAC) protocols and channel coding techniques are analyzed to achieve energy efficiency in WSNs.

In [3] it is shown that in WSNs the transmission energy can be lowered accepting to receive error-affected data. In this case the receiver should embed error correction strategies to recover the original data. In particular, the amount of energy spent to perform error correction should be significantly lower than the energy saved at the transmitter side. As an example, in [4,5] an energy efficient error correction scheme for WSNs is proposed. In particular, in [5] the physical layer of the IEEE 802.15.4 standard [6] is augmented introducing interleaving and forward error correction. In [2,3] several classes of codes are investigated, including Reed–Solomon codes, convolutional codes, turbo codes and Low-Density-Parity-Check (LDPC) codes [7,8]. Experimental results in [3] show that LDPC codes are good candidates for WSN applications as they feature a significant coding gain as compared with other codes. However, they consume about one order of magnitude more than simpler codes as the extended Hamming ones. Most of previous works proposing error correction codes for WSNs assume that networks contain at least two classes of nodes: sensing nodes and central nodes. Sensing nodes feature lower computational capabilities and available energy than central nodes. Thus, sensing nodes send coded information to a central node which performs the decoding operations. On the contrary, this work investigates homogeneous WSNs where each node can both transmit and receive coded information. A similar idea is proposed in [9] with focus on turbo codes. In particular, in [9] it is shown that the energy consumption of homogeneous WSN is reduced by about 70% resorting to turbo codes. In this work we show that even higher energy saving and smaller area can be achieved with LDPC codes. In particular, this work shows that small block length LDPC codes are adequate for typical throughput and data transmission requirements of WSNs.

The paper is structured as follows: Section 2 deals with LDPC coding and decoding algorithms whereas Section 3 concentrates on modeling the WSN environment. Section 4 details the proposed LDPC decoder architecture and Section 5 shows the experimental results. Finally, in Section 6 conclusions are drawn.

## 2. Coding and Decoding Algorithms for LDPC Codes

LDPC codes are a class of linear block codes, characterized by a very sparse  $M \times N$  parity-check matrix  $\mathbf{H}$  where valid codewords  $x$  satisfy  $\mathbf{H} \cdot (x)' = 0$  and  $(\cdot)'$  represents the transposition operator. Each LDPC code can be represented as a bipartite graph, known as Tanner Graph [10], containing two sets of nodes: Variable Nodes (VNs) and Check Nodes (CNs). VNs are associated to the  $N$  bits of the codeword, whereas CNs correspond to the  $M$  parity-check constraints. Edges in the graph correspond to ones in the  $\mathbf{H}$  and most of decoding algorithms imply the exchange of information along the edges of the Tanner graph. The most common algorithm to decode LDPC codes is the *Belief Propagation* (BP) algorithm. The VNs receive the intrinsic information  $\lambda$  (likelihood functions *i.e.*, probabilities) from the channel and update it depending on the results of the parity check equations computed at the CNs. This process is iterated several times until either the maximum number of iterations is reached, or a convergence criterion is met. This criterion may be that a codeword was successfully decoded.

There are two main scheduling schemes for the BP [11]: two-phase scheduling and layered scheduling [12]. The latter nearly doubles the convergence speed as compared to two-phase scheduling. In a layered decoder, parity-check constraints are grouped in layers, each of which is associated to a component code. Then, layers are decoded in sequence by propagating extrinsic information from one layer to the following one [12]. When all layers have been decoded, one iteration is complete and the overall process can be iteratively repeated up to the desired level of reliability.

Let  $S_j$  represent the Log-Likelihood-Ratio (LLR) of the bit in column  $j$  of  $\mathbf{H}$ . Bit LLR  $S_j$  is initialized to the corresponding received soft value. Then, for each parity constraints  $m$  in a given layer, the following operations are executed:

$$Q_{mj} = S_j^{(\text{old})} - R_{mj}^{(\text{old})} \quad (1)$$

$$A_{mj} = \sum_{n \in \mathcal{N}_m, n \neq j} \Psi(Q_{mn}) \quad (2)$$

$$s_{mj} = \prod_{n \in \mathcal{N}_m, n \neq j} \text{sgn}(Q_{mn}) \quad (3)$$

$$R_{mj}^{(\text{new})} = -s_{mj} \cdot \Psi(A_{mj}) \quad (4)$$

$$S_j^{(\text{new})} = Q_{mj} + R_{mj}^{(\text{new})} \quad (5)$$

$S_j^{(\text{old})}$  is the extrinsic information received from the previous layer and updated in Equation (5) to be propagated to the succeeding layer. Term  $R_{mj}^{(\text{old})}$ , pertaining to element  $(m, j)$  of  $\mathbf{H}$ , is used to compute Equation (1); the same amount is then updated in Equation (4),  $R_{mj}^{(\text{new})}$ , and stored to be used again in the following iteration. In Equations (2) and (3)  $\mathcal{N}_m$  is the set of all bit indices that are connected to parity constraint  $m$ .

Unfortunately, the computation of Equations (2) and (4) is complex, as  $\Psi(\cdot)$  is a non-linear function. According to [13], Equation (2) can be simplified with a limited Bit-Error-Rate (BER) performance loss as

$$R_{mj}^{(\text{new})} \approx -s'_{mj} \cdot \min_{t \in \mathcal{N}_m \setminus j} \{|Q_{tj}|\} \quad (6)$$

usually referred to as *normalized-min-sum* approximation, where  $s'_{mj} = \sigma \cdot s_{mj}$  and  $\sigma \leq 1$ . For further details the reader can refer to [8,10].

A key concern in the design of high throughput LDPC code decoders comes from the communication structure that must be allocated to support message passing among VNs and CNs. Three approaches can be followed in the high level organization of the decoder:

1. Fully Parallel Architectures (FPA): separate processing units are allocated for each VN and CN and all messages are passed in parallel along dedicated routes.
2. Partially Parallel Architectures (PPA): more processing units work in parallel, serving all VNs and CNs within a number of cycles; suitable organization and hardware support is required to exchange messages.
3. Serial architectures (SA): a single processing instance is allocated for both VN and CN computations and nodes are served sequentially; messages are exchanged by means of a unique memory.

The first approach leads to very high throughput, large implementation cost and severe congestion problems in the routing of interconnects [14]. For these reasons it is not adopted in practical implementations. The partially parallel architecture requires a large bandwidth between processing units and memories where messages are stored. Moreover, special attention is necessary to avoid collisions in the memory access [15]. However, the partially parallel organization allows to precisely tune the wanted degree of parallelism with respect to the addressed throughput and it was proved to be the best solution for the implementation of efficient decoders [15–19]. The serial approach leads to low cost and low power implementations and it also offers a high level of flexibility with respect to the supported code. However serial architectures did not receive much attention, due to the fact that the sequential processing does not achieve large throughput. This solution is particularly suitable for software implementations on Digital Signal Processors [20]. As throughput requirements in WSN applications are usually much lower than in wireless communications, the serial approach appears as the best solution to implement low cost and low energy decoding in a sensor node.

### 3. Wireless Sensor Network Environment and Modeling

Required throughput and energy budget are important parameters to model the environment of a WSN. Although the throughput depends on the application, several recent works [21–24] as well as off-the-shelf products for the IEEE 802.15.4 standard target a throughput  $T$  of 250 kb/s. According to [3] the amount of energy per bit saved due to the use of a correcting code ( $\Delta E$ ) can be expressed as

$$\Delta E = E_{TX,U} - E_{TX,C} - E_{enc} - E_{dec} \quad (7)$$

where  $E_{TX,U}$  and  $E_{TX,C}$  are the amounts of energy per information bit spent to transmit one bit in an uncoded and coded system respectively.  $E_{enc}$  and  $E_{dec}$  are the amounts of energy per bit spent by the LDPC encoder and decoder. Assuming a Binary-Phase-Shift-Keying (BPSK) modulation, each  $E$  term in Equation (7) can be written as a function of the power consumption  $P$  and the throughput  $T$  of the corresponding task. For a fair comparison we assume that the throughput sustained by the transmitter is the same for both the uncoded and coded case. As a consequence, Equation (7) can be rewritten as

$$\Delta E = \frac{P_{TX,U} - P_{TX,C} - P_{enc} - P_{dec}}{T} \quad (8)$$

However, as shown in [25] and [26] the complexity and the power consumption of LDPC encoding is negligible with respect to decoding. As a consequence, in the following the  $P_{enc}$  term will be neglected. Moreover, as highlighted in [3], each  $P_{TX}$  term can be written as a function of the path loss  $A(d)$  at a given distance  $d$ , the thermal noise  $N_0 \cdot B$  (where  $B$  is the signal bandwidth and  $N_0$  is the noise power spectral density), the Signal-to-Noise-Ratio (SNR) at the receiver and the receiver noise figure  $F$ :

$$P_{TX} = A(d) \cdot N_0 \cdot B \cdot 10^{(SNR+F)/10} \quad (9)$$

According to [27],

$$A(d) = \left(\frac{4\pi}{\lambda}\right)^2 \cdot d^n \quad (10)$$

where  $\lambda$  is the wavelength of the corresponding carrier frequency  $f$  and  $n$  is the path loss exponent, where  $n = 2$  and  $n = 4$  are good approximations for free space and dense environment propagations respectively. Assuming the same  $A(d)$  and  $F$  values for both uncoded and coded systems, Equation (8) can be rewritten as

$$\Delta E = \frac{A(d) \cdot N_0 \cdot B \cdot 10^{F/10} \cdot (10^{\text{SNR}_U/10} - 10^{\text{SNR}_C/10}) - P_{dec}}{T} \quad (11)$$

where  $\text{SNR}_U$  and  $\text{SNR}_C$  are the SNR at the receiver in the uncoded and coded systems respectively. Thus, given the curves representing the BER of one system as a function of the SNR, we obtain for each BER value the amounts  $\text{SNR}_U$  and  $\text{SNR}_C$  with  $\text{SNR}_G = \text{SNR}_U - \text{SNR}_C$  representing the SNR gain achieved using error correction. So Equation (11) can be rewritten as

$$\Delta E = \frac{A(d) \cdot N_0 \cdot B \cdot 10^{(\text{SNR}_U+F)/10} \cdot (1 - 10^{-\text{SNR}_G/10}) - P_{dec}}{T} \quad (12)$$

The expression obtained in Equation (12) will be used in Section 5 to show the effectiveness of the proposed LDPC architecture.

#### 4. LDPC Decoder Architecture Design

LDPC codes are known to nearly achieve the Shannon limit when the block of data is very large ( $N \rightarrow \infty$ ) [10]. However, in WSN applications the amount of bits exchanged by nodes is limited, leading to small  $N$  values. Nevertheless, in [28,29] it is shown that LDPC codes can achieve excellent performance even when  $N$  is small. In this work, we analyze the minimum  $N$  LDPC code from the IEEE 802.16e standard [30], which corresponds to  $N = 576$  coded bits and  $K = R \cdot N = 288$  uncoded bits ( $R = 0.5$ ). Moreover, we considered the two best performing regular codes with  $N = 96$  and  $N = 204$  ( $K = 48$ ,  $K = 102$ ) respectively, taken from MacKay database [31] and referred to as 96.33.966 and 204.33.484 ( $R = 0.5$  for both).

In order to size the LDPC decoder architecture, finite precision analysis ought to be performed. Given that  $p_S$  and  $p_R$  are the number of bits to represent  $S_j$  and  $R_{mj}$  metrics respectively, as in Equations (1–6), simulations have been carried out for  $p_S \in \{5, 6\}$  and  $p_R \in \{3, 4\}$ ; normalized-min-sum approximation with  $\sigma = 0.875$  has been employed. The performance of the three considered codes are shown in Figures 1–3 both in the floating point and fixed point cases together with the performance of the corresponding uncoded system. Furthermore, it has been observed that targeting a BER of  $10^{-4}$  as in [3,9] and imposing a maximum of ten iterations ( $I = 10$ ), the performance loss is negligible.

Figure 1. BER performance of the  $N = 96$  LDPC code.

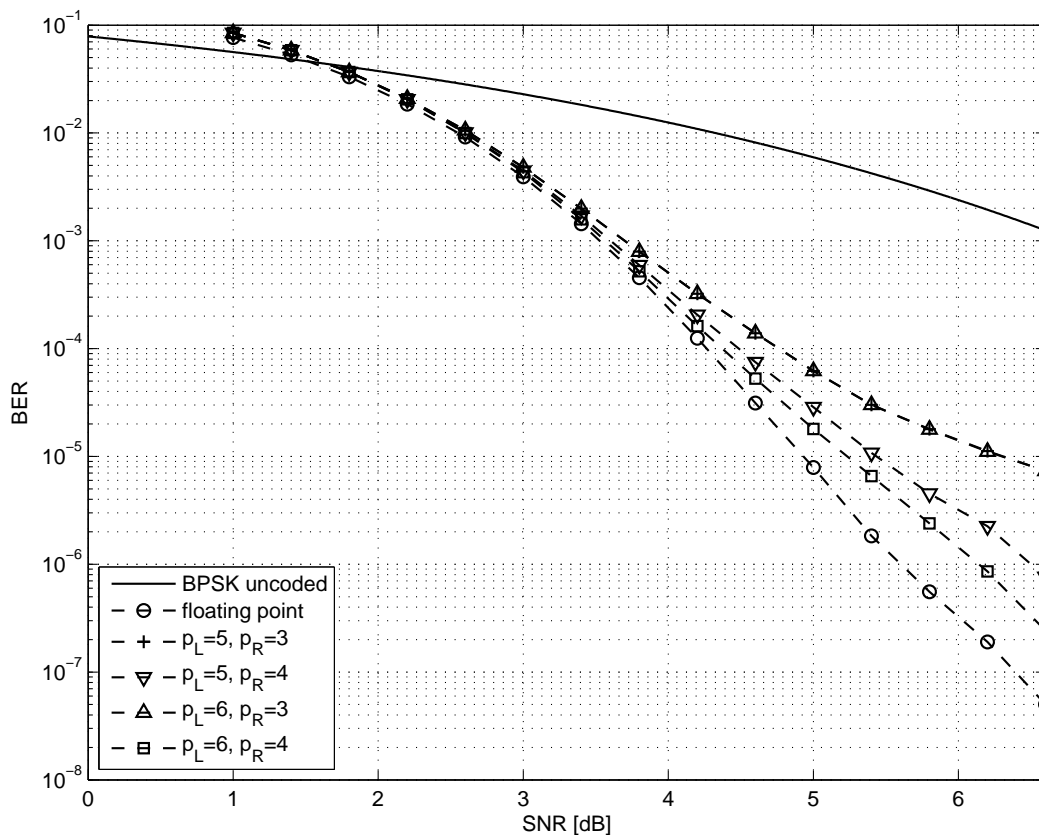
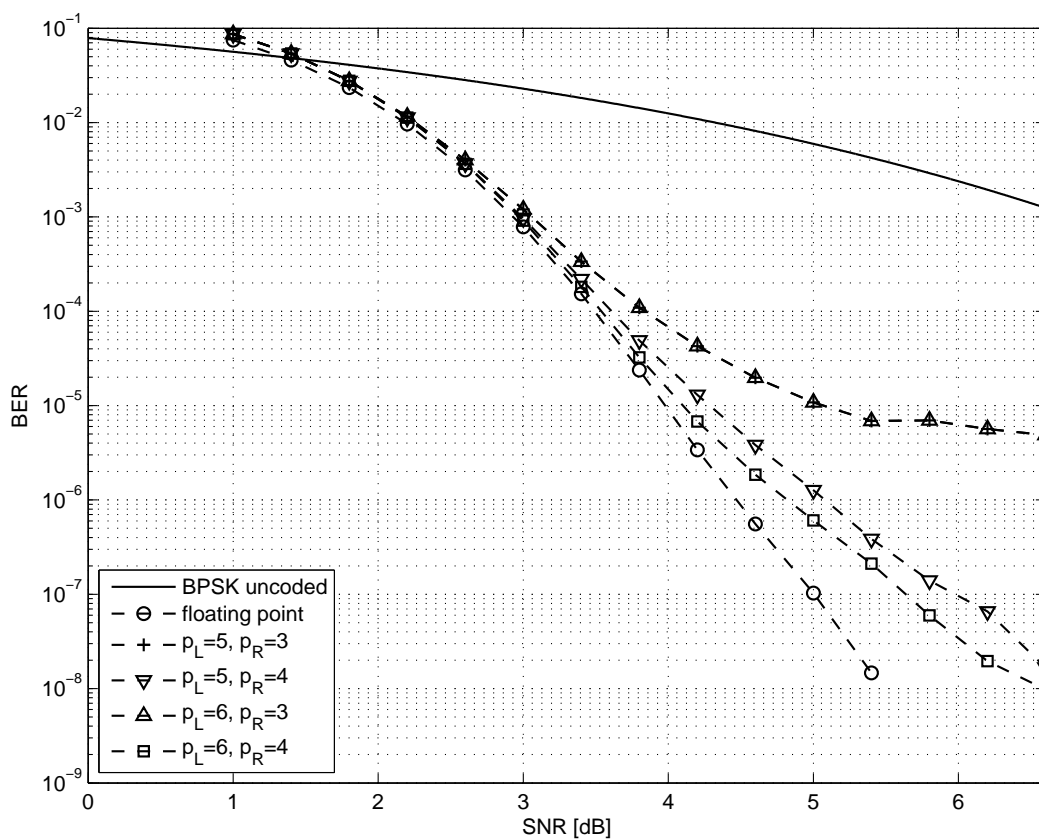
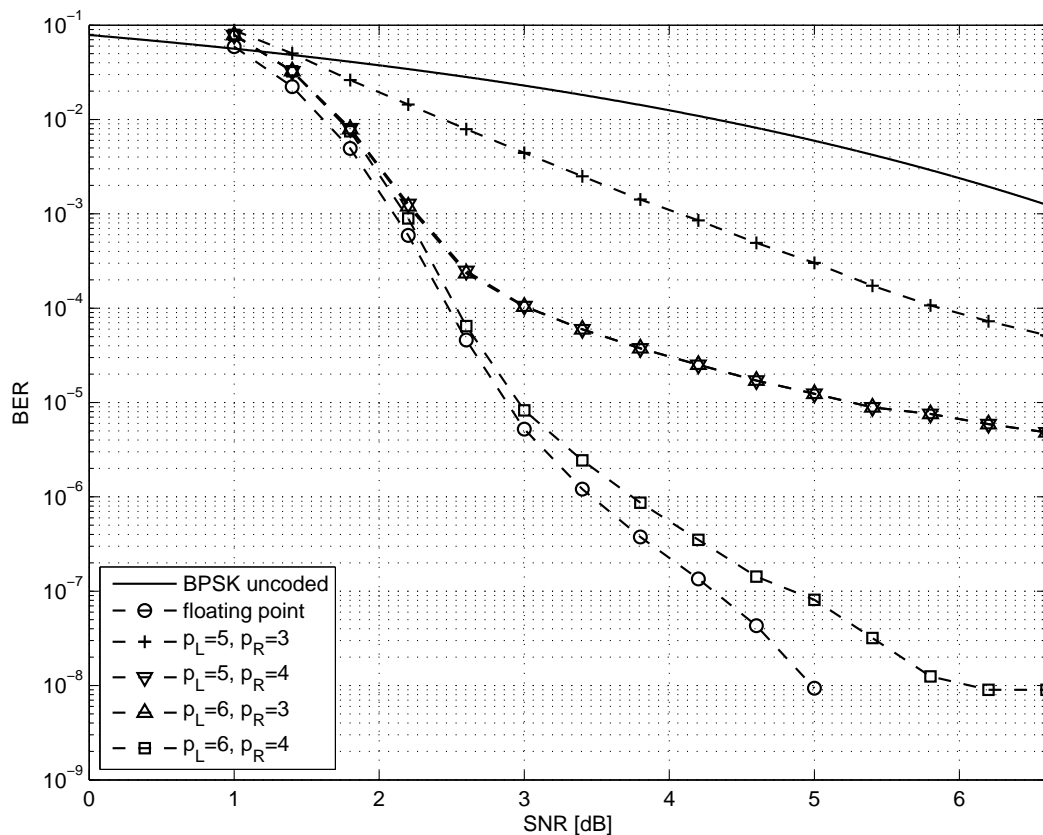


Figure 2. BER performance of the  $N = 204$  LDPC code.



**Figure 3.** BER performance of the  $N = 576$  LDPC code.

Due to the low throughput required, we assume that a fully serial processor architecture, which executes the decoding algorithm on one CN at the time, is a reasonable solution. In this case the throughput sustained by the architecture, defined as the number of decoded bits over the decoding time, is

$$T = \frac{K \cdot f_{clk}}{M \cdot I \cdot d_c^{\max} + D} = \frac{K \cdot f_{clk}}{\frac{1-R}{R} \cdot K \cdot I \cdot d_c^{\max} + D} \quad (13)$$

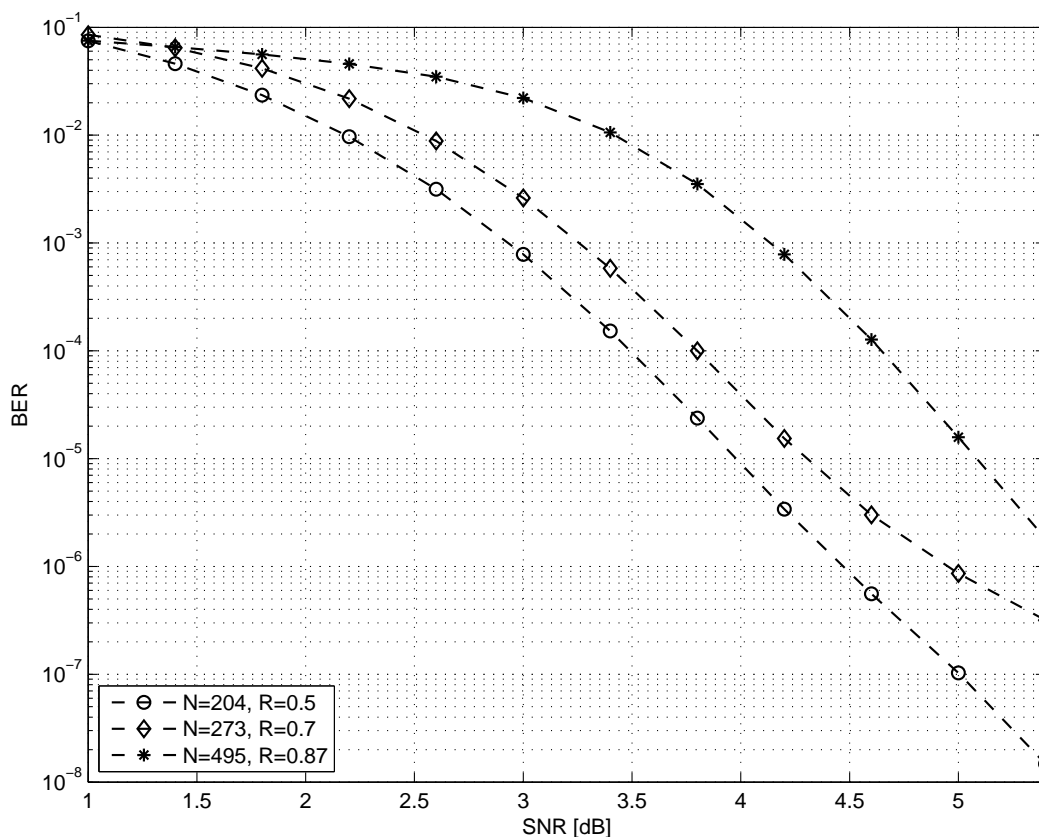
where  $f_{clk}$  is the decoder clock frequency,  $I$  is the maximum number of iterations,  $d_c^{\max}$  is the maximum degree of a CN, *i.e.*, the maximum number of edges on a CN and  $D$  is the latency of the architecture. It is worth noting that Equation (13) can be adapted to parallel and partially parallel architectures by substituting  $M$  with  $M/W$  where  $W$  is the number of rows (in  $\mathbf{H}$ ) processed in one clock cycle. The latency  $D$  in Equation (13) can be minimized avoiding idle cycles between iterations, so that  $D = d_c^{\max}$ . Thus, the throughput can be approximated as

$$T = \frac{K \cdot f_{clk}}{\left(\frac{1-R}{R} \cdot K \cdot I + 1\right) \cdot d_c^{\max}} \approx \frac{R \cdot f_{clk}}{(1-R) \cdot I \cdot d_c^{\max}} \quad (14)$$

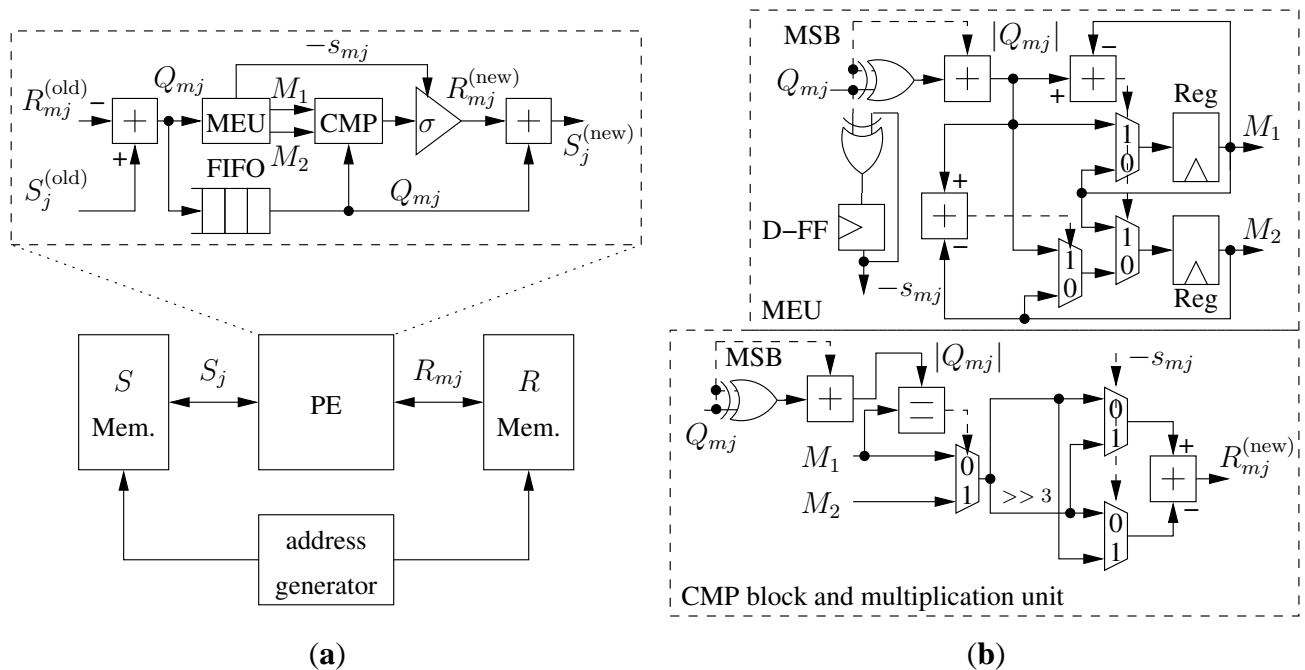
As it can be observed, the throughput increases with  $R$  so low-rate codes are a conservative choice to achieve the target throughput. Moreover, if we fix  $N$  we observe that increasing the rate has the effect of reducing the BER performance of the code. Thus, we considered the  $N = 204$ ,  $R = 0.5$  code and tried to increase both  $N$  and  $R$ . From MacKay database [31] we considered the following two high-rate codes where  $N > 204$ :  $N = 273$ ,  $R = 0.7$  and  $N = 495$ ,  $R = 0.87$  referred to as 273.82.3.353 and 495.62.3.2915 respectively. As shown in Figure 4 the BER performance of both codes

is lower than the one obtained for  $N = 204, R = 0.5$ . Furthermore, codes with  $N > 204$  require a larger amount of memory than the  $N = 204, R = 0.5$  code. From this analysis we infer that for the most complex code among the ones considered in this work, *i.e.*,  $d_c^{max} = 7$  for the IEEE 802.16e  $N = 576, R = 0.5$  code, and given the target throughput  $T = 250$  kb/s and  $I = 10$ , Equation (14) leads to  $f_{clk} \geq 17.5$  MHz. In this work we fix  $f_{clk} = 20$  MHz as a conservative value. Thus, the proposed architecture, inspired by the data-path of the solution proposed in [32], is made of four blocks as shown in the bottom part of Figure 5(a): a processing element (PE) devoted to implement the computation described in Equations (1–6) with the normalized-min-sum approximation;  $S$  and  $R$  memories, where  $S_j$  and  $R_{mj}$  metrics are stored; and an address generator. As depicted in the upper part of Figure 5(a) the PE contains: (i) a subtractor to compute  $Q_{mj}$  Equation (1), (ii) a Minimum-Extractor-Unit (MEU), a compare block (CMP) and a multiplication by  $\pm\sigma$  required to compute  $s_{mj}$  Equation (3) and  $R_{mj}^{(new)}$  with the normalized-min-sum approximation Equation (6), (iii) a synchronization FIFO with  $d_c^{max}$  locations, (iv) an adder to compute  $S_j^{(new)}$  Equation (5).

**Figure 4.** BER performance of the ( $N = 204, R = 0.5$ ), ( $N = 273, R = 0.7$ ) and ( $N = 495, R = 0.87$ ) LDPC codes.



**Figure 5.** Proposed decoder architecture: (a) general structure and PE detail; (b) MEU, CMP block and multiplication unit block schemes.



The MEU, detailed in the upper part of Figure 5(b) is made of two parts. The first one computes  $-s_{mj}$  xoring the sign of  $Q_{mj}$  values, *i.e.*, the most significant bit (MSB) of  $Q_{mj}$ , and saving the result in a D-Flip-Flop (D-FF). The second part computes the absolute value of  $Q_{mj}$ . Then, since the min function in Equation (6) is on  $\mathcal{N}_m \setminus j$ , the MEU finds the first two minimum values among the possible  $\mathcal{N}_m$  leaving to the CMP block to exclude the  $j$ -th one. The first two minimum values ( $M_1$  and  $M_2$ ) are obtained by the means of two subtractors, three multiplexer and two registers that implement Algorithm 1, where MPV is the Maximum Positive Value.

---

**Algorithm 1** Algorithm to find the first two minimum values

---

**Require:**  $M_1 \leftarrow \text{MPV}$  and  $M_2 \leftarrow \text{MPV}$

- 1: **for**  $t \in \mathcal{N}_m$  **do**
  - 2:   **if**  $|Q_{tj}| < M_1$  **then**
  - 3:      $M_2 \leftarrow M_1$
  - 4:      $M_1 \leftarrow |Q_{tj}|$
  - 5:   **else if**  $|Q_{tj}| < M_2$  **then**
  - 6:      $M_2 \leftarrow |Q_{tj}|$
  - 7:   **end if**
  - 8: **end for**
- 

The CMP block and the multiplication unit are shown in the bottom part of Figure 5(b). The CMP block compares  $|Q_{mj}|$  with  $M_1$ . If they are equal,  $M_2$  is passed to the multiplication unit. The multiplication unit does not contain a real multiplier as  $\sigma = 0.875 = 1 - 1/8$  requires only a subtractor and a hard-wired three-bit right shift ( $\gg 3$ ). In order to take into account the  $-s_{mj}$  term, two multiplexers, driven by  $-s_{mj}$  are added to obtain  $R_{mj}^{(\text{new})}$  as in Equation (6).

## 5. Experimental Results

The proposed architecture has been described using VHDL language. The complete design flow, including synthesis, place and route has been performed with Synopsys Design Compiler and Cadence Encounter on a 90 nm CMOS standard cell technology with 9 levels of metal and supply voltage equal to 1 V. Post place and route simulations was run to obtain accurate capacitances and switching activities [33], which are necessary for estimating the power consumption. Area and power consumption results for the three codes analyzed in Section 4 with  $p_S \in \{5, 6\}$ ,  $p_R \in \{3, 4\}$  and  $f_{clk} = 20$  MHz are shown in Table 1.

**Table 1.** Post place and route area and power consumption of the proposed architecture.

$N$	$p_S$	$p_R$	Area [ $\mu\text{m}^2$ ]	$P_{dec}$ [ $\mu\text{W}$ ]
96	5	3	66,046	359
96	5	4	67,994	373
96	6	3	67,752	363
96	6	4	69,720	379
204	5	3	86,165	445
204	5	4	88,670	458
204	6	3	88,283	448
204	6	4	90,613	459
576	5	3	125,257	648
576	5	4	131,681	670
576	6	3	128,146	663
576	6	4	133,934	674

It is worth noting that it is difficult to make a fair comparison of the proposed architectures with other solutions proposed in the literature because the target applications are different. However, for the sake of completeness in Table 2 several LDPC decoder architectures are compared with the most area demanding and power consuming solution among the proposed ones ( $N = 576$ ,  $p_S = 6$ ,  $p_R = 4$ , last row of Table 1).

As it can be observed, most solutions proposed in the literature address partially parallel architectures designed for wireless communications and broadcasting applications. As a consequence, they are sized to obtain throughput of hundreds of Mb/s or even Gb/s with large blocks of data. On the contrary, the proposed serial architecture is specifically tailored for WSN applications where throughput and block length are much smaller, we assume here  $T \leq 250$  kb/s and  $N \leq 576$ . Since the considered architectures have been designed on different technologies, we scale them all to the 90 nm technology node ( $A_{90}$ ) for the sake of fairness. The scaling is obtained multiplying the area (fifth column in Table 2) by  $(\mathcal{F}/90)^2$ , where  $\mathcal{F}$  is the feature size shown in the fourth column of Table 2. As expected, the proposed architecture is about one order of magnitude smaller than the other ones (fifth and sixth columns in Table 2). On the contrary, partially parallel architectures consume less energy per bit and energy per bit per iteration than

serial solutions (eleventh and twelfth columns in Table 2). Assuming that area and energy consumption are the most important metrics to choose a decoder architecture for WSN applications, we introduce two figures of merit. The first one is the normalized area  $\Phi_A(k) = A_{90}(k) / \min_k \{A_{90}(k)\}$  where  $A_{90}(k)$  is the area of the  $k$ -th architecture scaled to the 90 nm technology node. The second one is the normalized energy per bit per iteration  $\Phi_E(k) = E_I(k) / \min_k \{E_I(K)\}$ . These two figures of merit represent how far an architecture is from the minimum area and minimum energy per bit per iterations ones respectively. Assuming that  $\Phi_A$  and  $\Phi_E$  are equally important, their product shows which architecture is more suited for WSN applications among the compared ones. As shown in the last column of Table 2 the proposed architecture is the one with minimum  $\Phi = \Phi_A \cdot \Phi_E$ . It is worth noting that as shown in the last two rows of Table 2 the proposed architecture shows better area and energy figures than the recently proposed turbo decoder architecture for WSN applications described in [9].

**Table 2.** Comparison of different architectures.

Reference	Arch.	$N$	Tech. [nm]	Area [mm <sup>2</sup> ]	$A_{90}$ [mm <sup>2</sup> ]	$f_{clk}$ [MHz]	$T$ [Mb/s]	$P_{dec}$ [mW]	$I$	$E$ [pJ/b]	$E_I$ [pJ/b/it]	$\Phi$
[14]	FPA	1,024	160	52.5	16.6	64	1,000	690	64	690	11	404
[18]	PPA	64,800	90	13.1	13.1	270	180	853	-	4,740	-	-
[19]	PPA	2,304	130	4.8	2.3	214	955	397	10	416	42	141
[32]	PPA	64,800	90	4.1	4.1	300	90	-	30	-	-	-
[34]	PPA	1,944	130	7.4	3.5	111	250	76	8	304	38	197
[35]	PPA	2,048	65	7.15	13.7	300	6,680	1,030	8	154	19	95
[9]	SA <sup>(a)</sup>	6,144	90	0.35	0.35	333	1.03	4.17	5	4,049	810	198
This	SA	576	90	0.13	0.13	20	0.25	0.67	10	2,696	270	25

<sup>(a)</sup> Serial turbo decoder architecture for WSN applications.

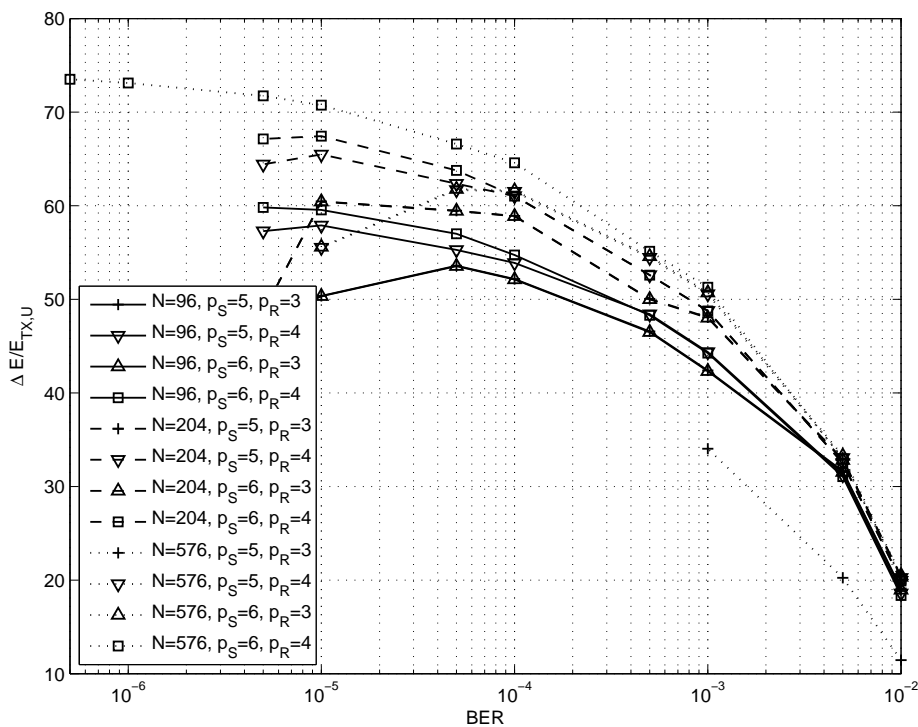
As highlighted in [36], several standards have been proposed for WSNs. It can be interestingly noted that most of them rely on the physical layer of the IEEE 802.15.4 standard. Thus, to evaluate the gain of the proposed architecture in a WSN environment we assume typical parameters taken from the IEEE 802.15.4 standard, namely  $f = 2.4$  GHz and  $B = 80$  MHz and we fix  $d = 50$  m. Moreover, employing an ultra-low-power low-noise-amplifier, as the one proposed in [37], we can fix  $F = 3.8$  dB.

In the following we investigate the energy saving obtained for a path loss exponent equal to three and four respectively, to model either typical indoor environments and outdoor urban/suburban foliated areas [38] or dense outdoor urban environments [39]. From Equation (12) the energy per bit required by an uncoded system ranges from tens of nJ/bit to few  $\mu$ J/bit depending on the considered path loss exponent value. As a consequence, to obtain a more significant information we compute the percentage of saved energy per bit with respect to the energy per bit of an uncoded system ( $\Delta E / E_{TX,U}$ ) as a function of the BER. The percentage of saved energy as function of the BER for all the results shown in Table 1 is depicted in Figures 6 and 7 for  $n = 3$  and  $n = 4$  respectively.

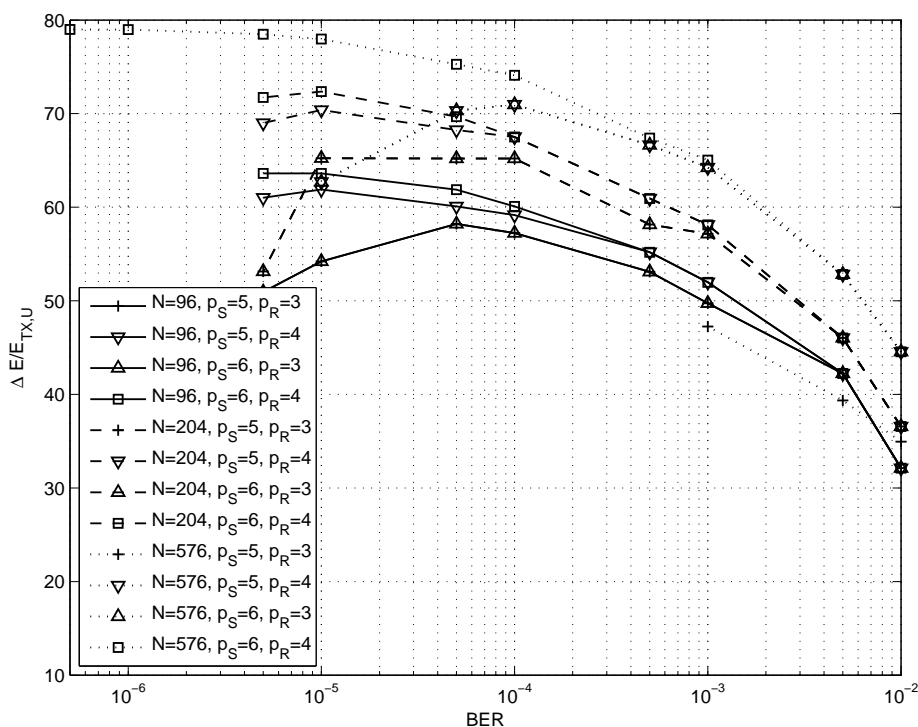
As it can be observed, both for  $n = 3$  and  $n = 4$  at a BER of  $10^{-4}$  the percentage of saved energy is more than the 50% and, in the best case, it achieves the 80%. It is worth pointing out that when a code

reaches the error floor region, the percentage of saved energy is maximum and then it decreases. Thus, the best energy saving performance is achieved in the waterfall region of the code.

**Figure 6.** Percentage of energy per bit saved as a function of the BER for  $n = 3$ .



**Figure 7.** Percentage of energy per bit saved as a function of the BER for  $n = 4$ .



## 6. Conclusions

Notwithstanding continuous progresses in the capacity of batteries, minimizing the energy dissipation still is one of the key objectives in the design of most sensor devices. In particular, transmission energy is a relevant component of the overall energy budget of a wireless sensor. This paper explores the use of LDPC codes to protect sent information against channel errors, thus allowing for a lower transmission energy. The energy that is saved at the transmission side depends on the coding gain of the selected code: more powerful the code, larger the saved energy. However a decoder is required at the receiver side to reconstruct the original information. The node to node communication throughput is low in wireless sensor applications and this enables the design of a fully serial decoding architecture, with limited implementation complexity and extremely low dissipated power. The additional energy consumed by the decoder has been evaluated by means of logical synthesis and layout generation. Final results prove that percentage saving as high as 80% can be achieved with the coded approach with respect to the usual uncoded transmission.

## References

1. Alkyldiz, I.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Int. J. Comput. Telecommun. Netw.* **2008**, *52*, 292–422.
2. Shih, E.; Cho, S.; Lee, F.S.; Calhoun, B.H.; Chandrakasan, A. Design considerations for energy-efficient radios in wireless microsensor networks. *J. VLSI Signal Process.* **2004**, *37*, 77–94.
3. Howard, S.L.; Schlegel, C.; Iniewski, K. Error control coding in low-power wireless sensor networks: When is ECC energy-efficient? *EURASIP J. Wirel. Commun. Netw.* **2006**, *2006*, 1–14.
4. Maunder, R.G.; Weddell, A.S.; Merrett, G.V.; Al-Hashimi, B.M.; Hanzo, L. Iterative decoding for redistributing energy consumption in wireless sensor networks. In *Proceedings of the 17th IEEE International Conference on Computation, Communications and Networking*, Saint Paul, MN, USA, 3–7 August 2008; pp. 1–6.
5. Li, L.; Maunder, R.G.; Al-Hashimi, B.M.; Hanzo, L. An energy-efficient error correction scheme for IEEE 802.15.4 wireless sensor networks. *IEEE Trans. Circuits Syst. II* **2010**, *57*, 233–237.
6. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Standard for Information technology, Telecommunications and information exchange between systems, Local and metropolitan area networks Specific requirements 2006; Part 15.4.
7. Benedetto, S.; Biglieri, E. *Principles of Digital Transmission: With Wireless Applications*; Kluwer Academic: Dordrecht, The Netherlands, 1999.
8. Richardson, T.; Urbanke, R. *Modern Coding Theory*; Cambridge University Press: Cambridge, UK, 2008.
9. Li, L.; Maunder, R.G.; Al-Hashimi, B.M.; Hanzo, L. A low-complexity turbo decoder architecture for energy-efficient wireless sensor networks. *IEEE Trans. VLSI Syst.* **2010**, in press.
10. MacKay, D.J.C. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inf. Theory* **1999**, *45*, 399–341.

11. Guilloud, F.; Boutillon, E.; Tusch, J.; Danger, J.L. Generic description and synthesis of LDPC decoders. *IEEE Trans. Commun.* **2007**, *55*, 2084–2091.
12. Hocevar, D.E. A reduced complexity decoder architecture via layered decoding of LDPC codes. In *Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS' 04)*, Austin, TX, USA, 13–15 October 2004; pp. 107–112.
13. Chen, J.; Dholakia, A.; Eleftheriou, E.; Fossorier, M.; Hu, X.Y. Reduced-complexity decoding of LDPC codes. *IEEE Trans. Commun.* **2005**, *53*, 1288–1299.
14. Banksby, A.J.; Howland, C.J. A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder. *IEEE J. Solid-State Circuits* **2002**, *37*, 404–412.
15. Quaglio, F.; Vacca, F.; Castellano, C.; Tarable, A.; Masera, G. Interconnection framework for high-throughput, flexible LDPC decoders. In *Proceedings of the Design, Automation and Test in Europe (DATE '06)*, Munich, Germany, 6–10 March 2006; pp. 1–6.
16. Moussa, H.; Baghdadi, A.; Jezequel, M. Binary de bruijn on-chip network for a flexible multiprocessor LDPC decoder. In *Proceedings of the 45th Annual Design Automation Conference*, Anaheim, CA, USA, 9–13 June 2008; pp. 429–434.
17. Shih, X.Y.; Zhan, C.Z.; Wu, A.Y. A 7.39 mm<sup>2</sup> 76 mW (1944, 972) LDPC Decoder Chip for IEEE 802.11n Applications. In *Proceedings of the IEEE Asian Solid-State Circuits Conference (A-SSCC '08)*, Fukuoka, Japan, 3–5 November 2008; pp. 301–304.
18. Muller, S.; Schreger, M.; Kabutz, M.; Alles, M.; Kienle, F.; Wehn, N. A novel LDPC decoder for DVB-S2 IP. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE '09)*, Nice, France, 20–24 April 2009; pp. 1308–1313.
19. Xiang, B.; Bao, D.; Huang, S.; Zeng, X. An 847-955 Mb/s 342-397 mW dual-path fully-overlapped QC-LDPC decoder for WiMAX system in 0.13  $\mu\text{m}$  CMOS. *IEEE J. Solid-State Circuits* **2011**, *46*, 1416–1432.
20. Lechner, G.; Sayir, J.; Rupp, M. Efficient DSP implementation of an LDPC decoder. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, Montreal, QC, Canada, 17–21 May 2004; pp. 665–668.
21. Hanzalek, Z.; Jurcik, P. Energy efficient scheduling for cluster-tree wireless sensor networks with time-bounded data flows: Application to IEEE 802.15.4/ZigBee. *IEEE Trans. Ind. Inf.* **2010**, *6*, 438–450.
22. Chen, J.; Yu, Q.; Zhang, Y.; Chen, H.H.; Sun, Y. Feedback-based clock synchronization in wireless sensor networks: A control theoretic approach. *IEEE Trans. Veh. Technol.* **2010**, *59*, 2963–2973.
23. Yoo, S.; Chong, P.K.; Kim, D.; Doh, Y.; Pham, M.L.; Choi, E.; Huh, J. Guaranteeing real-time services for industrial wireless sensor networks with IEEE 802.15.4. *IEEE Trans. Ind. Electron.* **2010**, *57*, 3868–3876.
24. Chen, L.; Carpenter, G.; Greenberg, S.; Frolik, J.; Wang, X.S. An implementation of decentralized consensus building in sensor networks. *IEEE Sens. J.* **2011**, *11*, 667–675.
25. Lee, C. Design of encoder and decoder for LDPC codes using hybrid H-matrix. *ETRI J.* **2005**, *27*, 557–562.
26. Xie, N.; Dong, G.; Zhang, T. Using lossless data compression in data storage systems: Not for saving space. *IEEE Trans. Comput.* **2011**, *60*, 335–345.

27. Rappaport, T.S. *Wireless Communications: Principles and Practice*; Prentice Hall: Upper Saddle River, NJ, USA, 1996.
28. Sorokine, V.; Kschischang, F.R.; Pasupathy, S. Gallager codes for CDMA applications—Part I: Generalizations, constructions and performance bounds. *IEEE Trans. Commun.* **2000**, *48*, 1660–1668.
29. Sorokine, V.; Kschischang, F.R.; Pasupathy, S. Gallager codes for CDMA applications—Part II: Implementations, complexity and system capacity. *IEEE Trans. Commun.* **2000**, *48*, 1818–1828.
30. *Air Interface for Fixed and Mobile Broadband Wireless Access Systems*; The Institute of Electrical and Electronics Engineers, Inc.: New York, NY, USA; September 2000; IEEE P802.16e/D12.
31. MacKay LDPC codes database. 2003. Available online: <http://www.inference.phy.cam.ac.uk/mackay/codes/data.html> (accessed on 29 January 2012).
32. Dielissen, J.; Hekstra, A.; Berg, V. Low cost LDPC decoder for DVB-S2. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE '06)*, Munich, Germany, 6–10 March 2006; pp. 1–6.
33. Pulimeno, A.; Graziano, M.; Piccinini, G. UDSM trends comparison: From technology roadmap to UltraSparc Niagara2. *IEEE Trans. VLSI syst.* **2011**, in press.
34. Shih, X.Y.; Zhan, C.Z.; Wu, A.Y. A 7.39 mm<sup>2</sup> 76 mW LDPC decoder chip for IEEE 802.11n applications. In *Proceedings of the IEEE Asian Solid-State Circuits Conference*, Fukuoka, Japan, 3–5 November 2008; pp. 301–304.
35. Zhang, Z.; Anantharam, V.; Wainwright, M.J.; Nikolic, B. An efficient 10G BASE-T ethernet LDPC decoder design with low error floors. *IEEE J. Solid-State Circuits* **2010**, *45*, 843–855.
36. Wagner, R.S. Standards-based wireless sensor networking protocols for spaceflight applications. In *Proceedings of the IEEE Aerospace Conference*, Dallas, MT, USA, 6–13 March 2010; pp. 1–7.
37. Taris, T.; Mabrouki, A.; Kraimia, H.; Deval, Y.; Begueret, J. Reconfigurable ultra low power LNA for 2.4GHz wireless sensor networks. In *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems*, Athens, Greece, 12–15 December 2010; pp. 74–77.
38. Sydor, J. True broadband for the countryside. *IEEE Commun. Eng.* **2004**, *2*, 32–36.
39. Aguiar, A.; Gross, J. *Wireless Channel Models*, Technical Report TKN-03-007, Telecommunications Networks Group, Technische Universitat Berlin: Berlin, Germany, 2003.