

Energy-conscious turbo decoder design: A joint signal processing and transmit energy reduction approach

Liang Li, Robert G. Maunder, Bashir M. Al-Hashimi, Mark Zwolinski and Lajos Hanzo

School of ECS, University of Southampton, SO17 1BJ, UK

Email: {ll08r, rm, bmah, mz, lh}@ecs.soton.ac.uk

Abstract—Turbo codes have been proposed for reducing the required transmission energy in Wireless Sensor Networks (WSNs), although this gain must be offset by the turbo decoder’s processing energy consumption. Previously, it has not been possible to estimate this processing energy consumption until a relatively late stage in the turbo code design process. This has prevented the consideration of processing energy consumption at the early design stages, when there is the greatest opportunity to adjust the parameters of the design. To address this, we propose a generalized turbo decoder architecture that supports a wide variety of parameters, as well as a framework for estimating its energy consumption as a function of these parameters at an early design stage. We demonstrate that this facilitates a holistic optimization of the turbo code parameters, minimizing the sum of both the transmission and processing energy consumption.

Index Terms—turbo codes, wireless sensor networks, energy consumption.

I. INTRODUCTION

In recent years, Wireless Sensor Networks (WSNs) have attracted significant interest in mobile and vehicular applications, for monitoring and controlling various system components during transit. However in these applications, the WSN nodes typically do not have regular or guaranteed access to abundant sources of energy. Instead, the WSN nodes are required to operate for extended periods of time without replacement or recharging of their scarce energy resources. Owing to this, WSNs require energy-efficient wireless communication.

The employment of Error-Correcting Codes (ECCs) in WSNs has been proposed [1], [2] for improving their Bit Error Rate (BER) performance, at the cost of increasing their computational complexity. By correcting the transmission errors that occur at lower transmission powers, ECCs facilitate a reduction in the overall Energy Consumption (EC) of WSNs. However, previous studies [1], [3], [4] have shown that in relay-aided multi-hop networks relying on decoding-and-forwarding, the relatively high complexity and EC of the ECC decoders may become prohibitive. Explicitly, the overall EC of the ECC employed depends on both the transmission

EC E_b^{tx} and on the extra processing EC E_b^{pr} imposed by the embedded decoder. Here, E_b^{tx} is determined by the coding gain provided by the ECC employed, while E_b^{pr} depends on the decoding algorithm and its hardware implementation. The encoder’s EC may however be insignificant compared to E_b^{tx} and E_b^{pr} , according to [1], [2]. As a result, the decisions made during the code design stage, including the choices of the parameters, have a direct effect on both E_b^{tx} and E_b^{pr} .

In a conventional design of a ECC, the impact of the parameters on the transmission EC E_b^{tx} imposed can be readily investigated using the classic BER analysis relying on an appropriately chosen path-loss model [5]. However, the processing EC E_b^{pr} has not been considered during the conventional code design process, owing to the lack of accurate estimation methods that allow the designer to investigate E_b^{pr} of a particular ECC during the early design stage. Instead, the computational complexity has been the prevalent factor used by designers for considering the trade-off between the performance and the resource requirements imposed by a particular design [6]. However, following this approach, it is too late to make any changes in the code design for optimizing the overall energy efficiency during the implementation phase.

In order to address this, we propose a framework that can be employed at an early design stage for estimating the processing EC of the turbo decoder architecture of [7], which was shown to be particularly energy efficient. We focus on Turbo Codes (TCs) employing the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm, since they are popular codes that have been adopted in numerous wireless communication standards and because they are capacity-approaching codes, potentially facilitating the greatest possible reduction in the transmission energy E_b^{tx} . We begin in Section II, by generalizing the turbo decoder architecture of [7], so that it can adopt any set of TC parameters. In Section III, we propose our framework, which facilitates the accurate estimation of the generalized turbo decoder’s EC, as a function of the TC parameters. In Section IV, we continue by invoking our energy estimation framework for a holistic TC design, which considers both E_b^{tx} and E_b^{pr} during the code design stage for arriving at an energy-efficient design for a specific target scenario. Specifically, for demonstrating the benefits of our holistic design method, we apply it to the TC design of [6]. In [6], 36 different design candidates were investigated using both BER and computational complexity analysis. By using the proposed design method

The research leading to these results has received funding from the European Unions Seventh Framework Programme ([FP7/2007-2013]) under the Concerto project. The financial support of RC-UK under the auspices of grant EP/J015520/1 and the UK-India Advanced Technology Centre, as well as that of the European Research Council (ERC) under its Advanced Fellowship scheme is also gratefully acknowledged.

for investigating the same design candidates, we demonstrate that neither pure BER nor computational complexity results are sufficient for investigating the overall energy efficiency of a TC, which justifies the rationale of our proposed design method. Finally, Section V concludes the paper.

II. THE ENERGY EFFICIENT TURBO DECODER ARCHITECTURE

The design of a typical turbo encoder requires decisions concerning the parameters, including the number of input bits k for each component encoder, the number of memory elements m for each component encoder and the number of non-systematic output bits n for each component encoder, as illustrated in Figure 1 [8], [9]. The choice of the Generator Polynomial (GP) determines the convolutional code used by the components encoders. However, we will demonstrate that this choice does not affect the EC significantly. Additionally, the interleaver length ($N \times k$) has to be determined during the early design stage, regardless of which type of interleaver is chosen. The additional parameter that has to be determined is the number B , indicating how many times the BCJR algorithm is performed during the decoding process. In the typical Twin-Component Turbo Code (TCTC) decoder shown in Figure 1, B is twice the number of iterations I . However, in the less typical Multiple-Component Turbo Code (MCTC) decoders [6], the decoding process does not always perform an integer number of iterations. Therefore, B is a better choice for characterizing the decoding complexity. Furthermore, as discussed in [7], the sliding-window technique is employed by the proposed architecture for the sake of reducing the memory requirements. As introduced in [10], the sliding-window technique consists of three stages during the decoding process, namely the forward recursion, the pre-backward recursion and the backward recursion. The length w_s of the sliding windows and the length w_p of the pre-backward recursion are two essential parameters. Finally, to obtain quantitative EC estimates, some further assumptions are required, which are not directly related to the ECC performance, but are closely related to the decoding EC E_b^{pr} . These assumptions include the process technology used for implementing the decoder, the supply voltage v , the operating clock frequency f and the operand width z of the datapath in the decoder's architecture. Throughout this treatise, the Taiwan Semiconductor Manufacturing Company (TSMC)'s 90 nm technology is assumed for the EC estimation framework, while [11] investigates the impact of technology scaling to BCJR decoders.

Since the parameters v and f are rarely used by the code designers, recommended values will be given in this work. In summary, the parameters required by the EC estimation framework from the code design stage are given in Table I.

In practice, all operations of the turbo decoding scheme can be performed by a simple Look-Up-Table-based Logarithmic Bahl-Cocke-Jelinek-Raviv (LUT-Log-BCJR) decoder [7], which employs an LUT to approximate the Jacobian logarithm used in the Log-MAP BCJR algorithm [12]. Note that only one of the component decoders seen in Figure 1 is activated at a time. When the LUT-Log-BCJR decoder employed is

TABLE I
SUMMARY OF THE VARIABLES IN THE ENERGY ESTIMATION FRAMEWORK.

k	The number of inputs of each component encoder
m	The number of memory elements of each component encoder
n	The number of non-systematic outputs of each component encoder
w_s	The sliding-window length
w_p	The pre-backward recursion length
N	The interleaver length
B	The number of times that the BCJR algorithm is performed
v	The supply voltage
f	The clock frequency
z	The word length of the datapath

performing the task of the upper decoder in Figure 1, the memory blocks storing the Logarithmic Likelihood Ratios (LLR) represent the *a priori* and extrinsic LLR memories connected to the upper decoder. By contrast, when the LUT-Log-BCJR decoder is performing the task of the lower decoder of Figure 1, it will rely on a different set of memories storing the LLRs of the lower decoder in Figure 1. The LUT-Log-BCJR decoding algorithm of the decoder architecture employed is detailed in [7]. The top-level configuration of the generalized LUT-Log-BCJR decoder architecture of [7] is portrayed in Figure 2. The architecture was designed by ensuring that

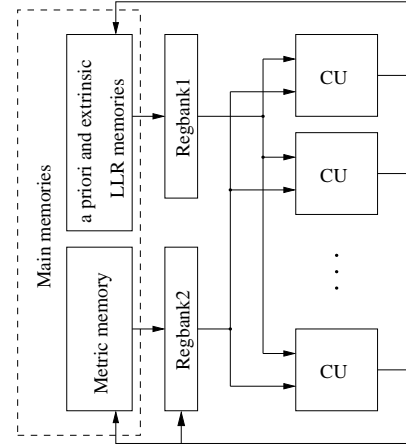


Fig. 2. The configuration of the proposed LUT-Log-BCJR decoder architecture.

the LUT-Log-BCJR decoding algorithm involved only Add Compare Select (ACS) operations [13]. Each Calculation Unit (CU) of Figure 2 is capable of operating in three modes, namely the adder mode, the \max^* mode and the idle mode, which perform additions, \max^* operations, or remain idle, respectively. During \max^* operations, a Look-Up Table (LUT) is employed to approximate the second term in the expression

$$\max^*(\tilde{p}, \tilde{q}) = \max(\tilde{p}, \tilde{q}) + \ln(1 + \exp(-|\tilde{p} - \tilde{q}|)),$$

as described in [7, Section III-C]. These calculations are performed using a two's-complement fixed point number representation, having an operand width comprising z number of bits. When employing an operand width of $z = 9$ bits, the LUT-Log-BCJR decoding algorithm is tolerant to the overflow

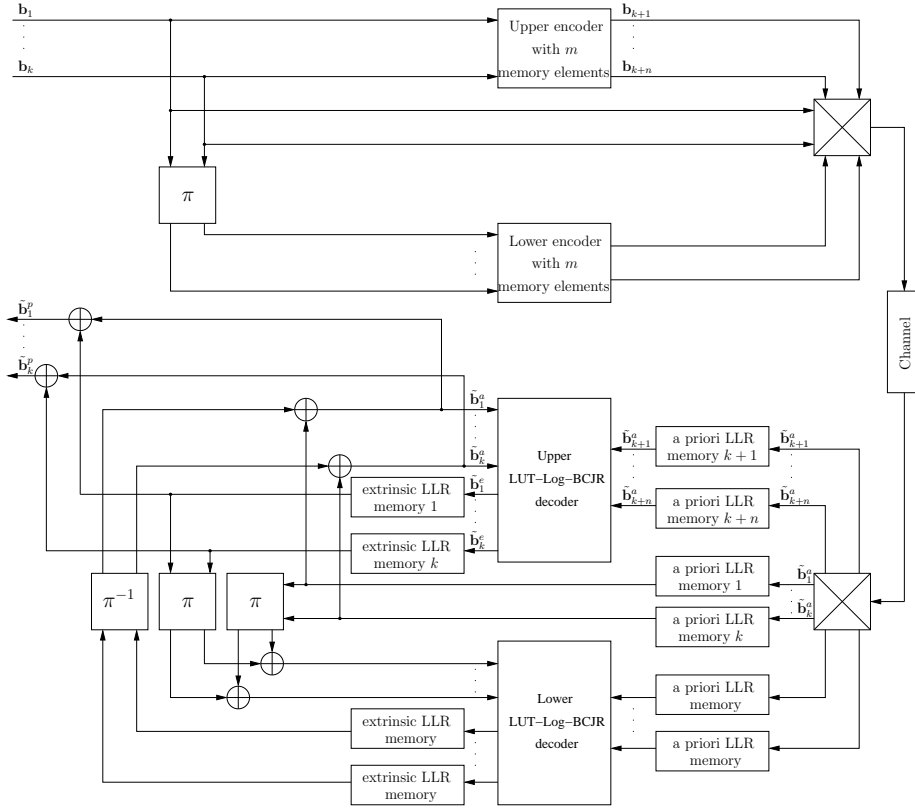


Fig. 1. The configuration of a typical TC scheme.

that is caused by adding two large numbers together [14]. For this reason, the architecture of [7] does not use saturation to avoid overflow. However, saturation and normalization techniques [15] may be introduced in order to facilitate lower operand widths, at the cost of a slightly increased hardware complexity. A total of 2^m CUs are operated in parallel, as described in [7]. A controller is used for scheduling the allocation of ACS operations to CUs. Since the interleavers of different TC designs are suited to implementation in many different ways, it is difficult to estimate the EC of the interleaver using a general method. For example, the Universal Mobile Telecommunications System (UMTS) [16], Long Term Evolution (LTE) [17] and WiMAX [18] TCs employ different deterministic interleaver designs, which employ different calculations to generate the interleaving patterns. In other TCs, pseudo-random interleaving patterns may be employed, which are not generated using calculations in an on-line manner, but are rather pseudo-randomly generated off-line and then stored for on-line use. However, as we will demonstrate later, the interleaver's EC in the WSN scenario may be insignificant compared to the remaining parts of the turbo decoder. For WSN applications, a fixed-length interleaver is assumed for estimating the EC.

III. ENERGY ESTIMATION FRAMEWORK

The EC is estimated in the unit of nJ/bit, which is defined as the energy consumed by the Sliding-Window LUT-Log-BCJR decoder when decoding a single bit of information. Note that there are $(N \times k)$ information bits per frame. In this

framework, the EC of the LUT-Log-BCJR decoder is divided into four parts, namely the datapath's, the controller's, the memories' and the interleaver's EC, which are estimated separately, yielding:

$$E_b^{\text{Turbo}} = E_b^{\text{Dp}} + E_b^{\text{Ctrl}} + E_b^{\text{Mem}} + E_b^{\text{Int}}. \quad (1)$$

In order to construct the EC models for E_b^{Dp} , E_b^{Ctrl} , E_b^{Mem} and E_b^{Int} , the time required by the different recursions of the decoding process, namely the forward recursion, the pre-backward recursion and the backward recursion, have to be calculated. Firstly, in Section III-A, the time required by the turbo decoder architecture employed is analyzed in terms of the units of clock cycles. Secondly, in Sections III-B to III-E, the energy models of E_b^{Dp} , E_b^{Ctrl} , E_b^{Mem} and E_b^{Int} are presented. Finally, the validation of the proposed framework is provided in Section III-F.

A. Timing analysis of the turbo decoder architecture employed

In this section, all the time durations allocated to the components during the decoding process are discussed, namely that of the forward recursion T_{fw} , the pre-backward recursion T_{pbw} and the backward recursion T_{bw} , as discussed in [7]. Additionally, each of these time durations is further divided into three components, which are the average time durations T^{add} of the addition, $T^{\text{max*}}$ of the \max^* operation and the idle time T^{idle} at each CU.

As discussed in [7], the scheduling of each CU in a LUT-Log-BCJR decoder can be designed with the aid of a

time schedule chart. More specifically, the number of clock cycles required to complete all operations associated with one trellis stage during the forward and pre-backward recursions can be quantified as

$$T_{fw} = T_{pbw} = T_{fw}^{add} + T_{fw}^{max^*} + T_{fw}^{idle}, \quad (2)$$

where $T_{fw}^{add} = 2^{k-1}(k+n)$, $T_{fw}^{max^*} = 4(2^k - 1)$ and $T_{fw}^{idle} = 1$ are the number of clock cycles in which addition, max* and idle operations are performed, respectively.

The corresponding number of clock cycles for the backward recursion can be quantified as

$$T_{bw} = T_{fw}^{add} + T_{fw}^{max^*} + T_{fw}^{idle}, \quad (3)$$

where

$$\begin{aligned} T_{bw}^{add} &= 2^{k-1}(k+n) + (2^{k+1} + \frac{1}{2^m})k, \\ T_{bw}^{max^*} &= 4(2^k - 1) + \left(4k + \left(\frac{\sum_{i=1}^{m-1} 2^i}{(m-1)2^m}\right)4(m-1)\right)k \text{ and} \\ T_{bw}^{idle} &= 2 + \left(1 - \frac{1}{2^m} + \left(1 - \frac{\sum_{i=1}^{m-1} 2^i}{(m-1)2^m}\right)4(m-1)\right)k. \end{aligned}$$

Finally, the number of clock cycles required per bit per BCJR operation is given by

$$T_e = \frac{w_s(T_{fw} + T_{bw}) + w_p T_{pbw}}{w_s \times k}, \quad (4)$$

where w_s is the length of the sliding-window employed in the forward and backward recursions, while w_p is the length of the window employed in the pre-backward recursion.

The overall throughput of the turbo decoder of Section II expressed in bit/s can be calculated as $f/(T_e B)$, where f is the clock frequency and B is the number of times that the BCJR algorithm is performed. Here, each decoding iteration comprises two operations of the BCJR algorithm.

B. Energy estimation of the datapath

For the datapath of the turbo decoder, the EC is estimated based on the separate analysis of the sub-modules, namely of the CU, the Regbank1 and the Regbank2 of Figure 2. Post-layout simulations of each of these sub-modules are performed for obtaining power-consumption-related information, which were based on $z = 9$ -bit operand-width implementations of the sub-modules. This operand-width was recommended in [14] for a $m = 3$ turbo decoder. For fixed-point datapath structures, the hardware complexity and EC scales linearly with the operand-width [19], while the corresponding turbo decoder's error correction performance was characterized in [14]. Based on our simulation results not included here due to the limited space available, the per-bit energy model is then derived for estimating the typical EC in terms of *nJ per clock cycle* for the different sub-modules, when performing different tasks. Finally, using the per-bit energy model of the sub-modules, the total EC of a datapath in a particular turbo decoder can be calculated based on the configuration of the datapath seen in Figure 2. Again, owing to space limitations, only some of the simulation results are presented as examples for supporting the mathematical models in this paper, because the simulation results would require excessive space.

1) *Calculation unit*: The parameters that have measurable impacts on the EC of CUs are k , m , n , v , w_s and w_p of Table I. The energy impact of the parameter z is averaged out, since the result considered here is the per-bit EC of the CU, which was derived from a 9-bit operand-width implementation. The parameters N and B are not considered here, since they are not related to this part of the model, which are for the average EC expressed in nJ/Clock Cycle. Furthermore, our simulation results in Figure 3 show that the range of the parameter f considered in this work, which is [10, 400] MHz, does not have a significant impact on the EC. Firstly, the per-bit EC of

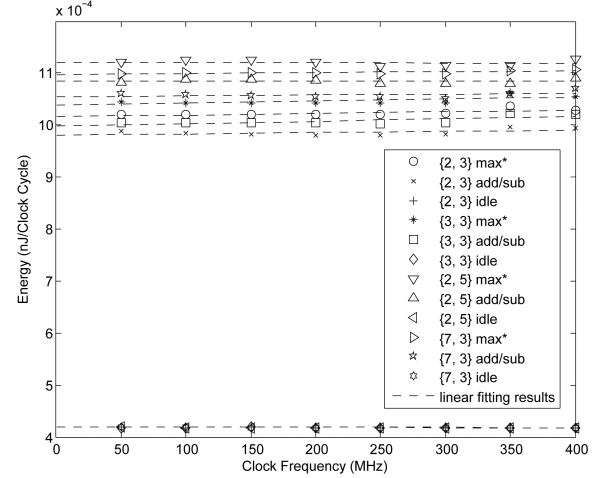


Fig. 3. E_{cyc} results of the CU with four different combination settings of $k+n$, m , where $v = 1.2$ V.

a CU per clock cycle evaluated for our three different modes, namely for the adder mode $E_{cyc}^{CU,add}$, max* mode E_{cyc}^{CU,max^*} and idle mode $E_{cyc}^{CU,idle}$ are modeled. According to the post-layout simulation results not included here, the parameters that have an observable impact on $E_{cyc}^{CU,add}$, E_{cyc}^{CU,max^*} and $E_{cyc}^{CU,idle}$ are n , m , k and v of Table I. The effect of the parameter v is independent of the effects of parameters n , m and k , since the former changes the current in the circuits while the latter changes the circuit structure of the CU. As for the circuit structure of the CU, each of the parameters $(k+n)$ and m affect the connection between the CUs and the register banks individually. Therefore, stipulating the assumption of $v = 1.2$ V for a particular operational mode, the CU's EC increases linearly with either $(k+n)$ or m , when the other one of the two is fixed, as shown in Figure 4. In a similar manner to [11], linear curve fitting may be applied to the simulation results for the sake of estimating the CU's EC as a function of both $(k+n)$ and m . These two functions are constrained to cross each other at the point where we have $k = 1$, $n = 1$ and $m = 1$, which are the smallest values for them. Furthermore, according to our simulation results not included here owing to space-economy, the impact of the variable v of Table I on the EC may be estimated after applying a scaling factor of $\frac{v^2}{1.2^2}$ [20].

As a result, all the three typical EC values can be modeled

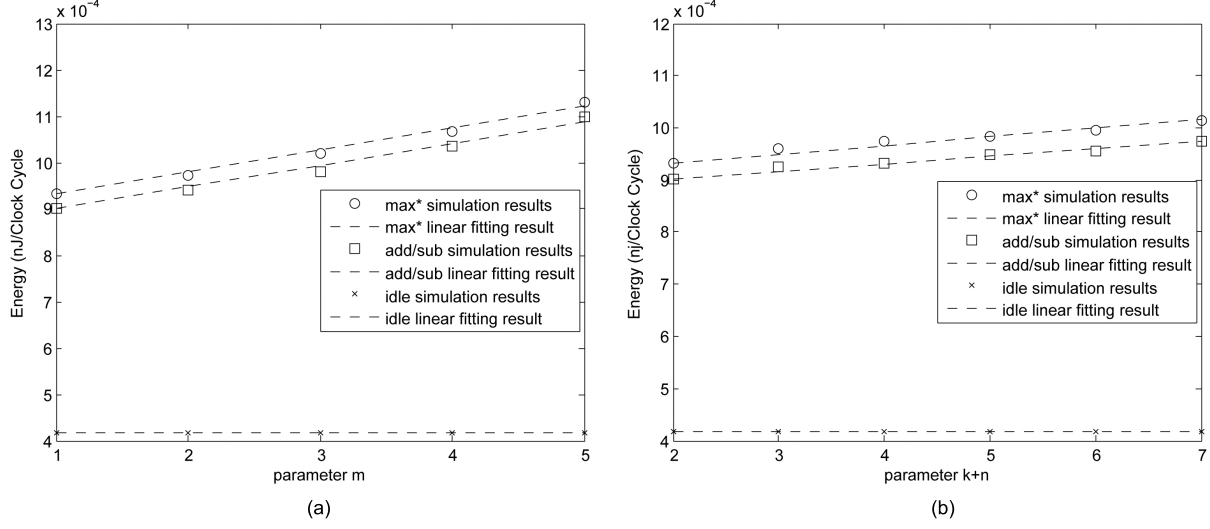


Fig. 4. E_{cyc} results of the CU for (a) different m , where $k = n = 1$ and (b) different $k + n$, where $m = 1$, both with $v = 1.2$ V and $f = 200$ MHz.

by the function

$$E_{cyc}^{CU, (mode)} = \frac{v^2}{1.2^2} (y_1 + y_2(k + n - 2) + y_3(m - 1)), \quad (5)$$

where $mode$ can be 'add', 'max*' or 'idle'. Naturally, for the different modes, the coefficients y_1 , y_2 and y_3 have different values, as seen in Table II. The action of the 1-bit CU during

TABLE II
SUMMARY OF THE COEFFICIENTS' VALUES OF EQUATION 5 WHEN THE 1-BIT CU IS IN DIFFERENT MODES.

mode	y_1	y_2	y_3
add	1.002×10^{-4}	0.163×10^{-5}	0.516×10^{-5}
max*	1.036×10^{-4}	0.188×10^{-5}	0.526×10^{-5}
idle	0.464×10^{-4}	0	0

the decoding process is based on a combination of the three operational modes. As a result, the typical per-bit EC of the CU during the forward recursion stage $E_{cyc}^{CU, fw}$, pre-backward recursion stage $E_{cyc}^{CU, pbw}$ and the backward recursion stage $E_{cyc}^{CU, bw}$ can be modeled on this basis, which is given by

$$E_{cyc}^{CU, fw} = E_{cyc}^{CU, pbw} = \frac{T_{fw}^{add} E_{cyc}^{CU, add} + T_{fw}^{max*} E_{cyc}^{CU, max*} + T_{fw}^{idle} E_{cyc}^{CU, idle}}{T_{fw}}, \quad (6)$$

$$E_{cyc}^{CU, bw} = \frac{T_{bw}^{add} E_{cyc}^{CU, add} + T_{bw}^{max*} E_{cyc}^{CU, max*} + T_{bw}^{idle} E_{cyc}^{CU, idle}}{T_{bw}}, \quad (7)$$

where T_{fw} , T_{bw} , T_{add} , T_{max*} and T_{idle} can be calculated based on Equation 2 to 4 in Section III-A. The average EC of the 1-bit CU for a turbo decoder can be modeled by

$$E_{cyc}^{CU} = \frac{w_s(E_{cyc}^{CU, fw} + E_{cyc}^{CU, bw}) + w_p E_{cyc}^{CU, pbw}}{2w_s + w_p}. \quad (8)$$

To validate the EC estimation results, we compared them to the post-layout simulation results of the CUs for four different

parametrizations over the operating clock frequency range of $f \in [10, 400]$ MHz. The results show that the maximum error of the estimation is 1.75%.

2) *Register bank*: For the register banks, the parameters that have measurable impacts on the EC are k , m , n , v , w_s and w_p of Table I. The rest of the parameters seen in Table I are not involved in this part of the mathematical model for reasons similar to those discussed in Section III-B1. Furthermore, two parameters are introduced for the energy model, namely the number of the registers r in a register bank and the updating rate u of a register bank quantified in terms of the average number of updated registers per clock cycle. According to the post-layout simulation results not included here, a register has a constant power consumption while its value remains unaltered, but it has an increased dynamic power consumption during the clock cycles, where its value is updated. As a result, the EC of a register bank is modeled by the variables r , u and v of Table I, where r and u of Regbank1 and Regbank2 seen in Figure 2 can be calculated using k , m , n , w_s , w_p , while the time duration results rely on Section III-A. Similarly to our model generated with the aid of the CU, based on the simulation results characterizing a register bank associated with different values of r , u and v , a function is generated with the aid of linear curve fitting [11] for the sake of modeling the EC of a 1-bit register bank, as follows:

$$E_{cyc}^{(Regbank)} = \frac{v^2}{1.2^2} r (0.168u + 0.1511) \times 10^{-3}, \quad (9)$$

where *Regbank* can be Regbank1 or Regbank2 of Figure 2. For Regbank1 and Regbank2, the parameters u and r can be calculated according to Table III. As shown in Equation 9, although there are six parameters for the register bank's energy model, essentially, the EC is determined by the parameter v and another two parameters, namely r and u . Except for v , the other five parameters of Table I are only used for calculating r and u . Therefore, to validate our energy estimation model, we compare the estimation results and the post-layout simulation results of the register bank associated with $r = 8$, $u \in [0, 0.5]$

TABLE III
SUMMARY OF THE INTERNAL VARIABLE VALUES OF EQUATION 9.

	r	u
Regbank1	$k + n$	$2w_s + w_p$
		$w_s(T_{fw} + T_{bw}) + w_p T_{pbw}$
Regbank2	$2^m(2^k - 1)$	$w_s \frac{k+n}{2T_{fw}} + w_s \frac{(k+n)(2^k-1)+2^{k+1}+2k(k+m-1)-4}{2(2^k-1)T_{bw}} + w_p \frac{k+n}{2T_{pbw}}$
		$2w_s + w_p$

for the operating clock frequency range of $f \in [10, 400]$ MHz. The results show that the maximum error of the estimation is as low as 1.24%.

3) *Datapath*: Finally, the EC of a datapath can be estimated by summing the EC of the CUs and register banks, which is expressed in nJ/bit as:

$$E_b^{Dp} = z \times B \times T_e \times (2^m E_{cyc}^{CU} + E_{cyc}^{Regbank1} + E_{cyc}^{Regbank2}). \quad (10)$$

To validate the final energy estimation of the total datapath EC, two LUT-Log-BCJR decoders of two different TCs were implemented using our generalized architecture. Post-layout simulations were then performed for obtaining the post-layout EC. Design-I has the specification of $k = 1$, $m = 3$ and $n = 1$. By contrast, Design-II relies on $k = 1$, $m = 2$ and $n = 1$. Inspired by the maximum block length of the LTE TC [17], we employ block lengths of $N = 6144$ bits for both designs. Additionally, $z = 9$, $w_s = 128$, $w_p = 24$, $f = 400$ MHz and $v = 1.2$ V were assumed in both cases, where $f = 400$ MHz is the maximum clock frequency that is supported by the architecture of [7]. Our results not included here demonstrated that the error in the estimated results is less than 2% of the post-layout simulation results.

C. Energy estimation of the controller

In typical ASIC design processes, no intricate knowledge of the controller's hardware implementation can be obtained before synthesis. This is because unlike the datapath and the memory blocks, the controller design is based on the behavior model. As a result, the EC of the controller is difficult to estimate at an early design stage [21], [22].

In this framework, an experience based model is proposed for estimating the controller's EC. The parameters that affect the controller include k , m , n , w_s , w_p and N . Firstly, a configurable Register-Transfer Level (RTL) model of the proposed architecture's controller is designed for investigating its EC in conjunction with different design parameters. This RTL module is not necessarily a complete controller for any particular LUT-Log-BCJR decoder, but it is designed to include the abstracted state machine and part of the combination logic circuits generating the control signals, which can be generalized for any decoder. The RTL module may be readily reconfigured by appropriately changing the parameters for the investigation. It represents up to 95% of the hardware complexity of the actual controllers. This inaccuracy in the controller's energy estimation is acceptable for the proposed architecture, since the simulation results show that the controller typically contributes only a small fraction (less than 5%) of the total EC of the turbo decoder.

Using the proposed RTL module, the EC of the proposed architecture's controller is investigated. Our post-layout simulation results not included here show that the EC variation caused by different clock frequencies f is insignificant. Therefore, $E_{cyc}^{Control}$ may be considered to be independent of f .

For $f = 400$ MHz, $v = 1.2$ V, $w_s = 128$, $w_p = 24$, $k = 1$, $m = 1$ and $n = 1$, $E_{cyc}^{Control}$ may be modeled as

$$E_{cyc,N}^{ctrl} = (0.01788[\log 2(N+1)] + 0.4293) \times 10^{-3}. \quad (11)$$

The parameter values of $w_s = 128$ and $w_p = 24$ are recommended for the proposed architecture, except for $N \leq 128$, in which case, the sliding window technique is not required and the situation is equivalent to $w_s = N$ and $w_p = 0$ for the design [7]. However, this exception does not affect the controller's EC, according to our simulation results using the WiMax TC as an example. Specifically, in this case, we have $N = 240$ and E_{cyc}^{ctrl} is 5.925×10^{-4} nJ/Clock Cycle when using the sliding window technique, while we have 5.9125×10^{-4} nJ/Clock Cycle, otherwise.

Let us now continue by proposing a technique of estimating E_{cyc}^{ctrl} as a function of the parameters k , m and n with the aid of four groups of simulation results. For $f = 400$ MHz, $v = 1.2$ V, $N = 1024$, $w_s = 128$, and $w_p = 24$ Table IV provides the four groups of results, which considered four different conditions of the variables k , m and n . To estimate E_{cyc}^{ctrl}

TABLE IV
 $E_{cyc}^{ctrl} (\times 10^{-4} \text{ nJ/CLOCK CYCLE})$ SIMULATION RESULTS OF VARIABLE k , m AND n .

group-1	k	1	2	3	4
$m = 1$	E_{cyc}^{ctrl}	6.255	6.4075	6.6	6.9725
$n = 1$	k	5	6	7	8
	E_{cyc}^{ctrl}	7.2475	7.3625	7.545	7.815
group-2	m	1	2	3	4
$k = 1$	E_{cyc}^{ctrl}	6.255	6.3275	6.3725	6.675
$n = 1$	m	5	6	7	8
	E_{cyc}^{ctrl}	6.21	6.39	6.3775	6.3225
group-3	n	1	2	3	4
$k = 1$	E_{cyc}^{ctrl}	6.255	6.205	6.145	6.39
$m = 1$	n	5	6	7	8
	E_{cyc}^{ctrl}	6.3325	6.2925	6.2825	6.4025
group-4	$k = m = n$	1	2	3	4
	E_{cyc}^{ctrl}	6.255	6.3925	6.8625	7.0175
$m = n$	$k = m = n$	5	6	7	8
	E_{cyc}^{ctrl}	7.465	7.5925	7.7225	7.7795

for a specific combination of k , m and n , firstly, $E_{cyc,k}^{ctrl}(k)$, $E_{cyc,m}^{ctrl}(m)$, $E_{cyc,n}^{ctrl}(n)$ and $E_{cyc,s}^{ctrl}(s)$ are used for generating the results of Table IV. For a certain specification of $\{k, m, n\}$, $s = \min(k, m, n)$ is defined and E_{cyc}^{ctrl} is estimated as follows:

$$E_{cyc,k,m,n}^{ctrl}(k, m, n) = E_{cyc,s}^{ctrl}(s) + [E_{cyc,k}^{ctrl}(k) - E_{cyc,k}^{ctrl}(s)] + [E_{cyc,m}^{ctrl}(m) - E_{cyc,m}^{ctrl}(s)] + [E_{cyc,n}^{ctrl}(n) - E_{cyc,n}^{ctrl}(s)]. \quad (12)$$

Combining the equations above for N , k , m , n and v allows E_{cyc}^{ctrl} to be estimated as

$$E_{cyc}^{ctrl} = \frac{v^2}{1.2} E_{cyc,k,m,n}^{ctrl}(k, m, n) + 0.01788(\lceil \log_2(N+1) \rceil - 11) \times 10^{-3}. \quad (13)$$

Finally, similar to the datapath, the energy efficiency of the controller can be calculated in nJ/bit as

$$E_b^{ctrl} = B \times T_e \times E_{cyc}^{ctrl}. \quad (14)$$

To verify the model, we compare the estimation results and the simulation results of E_b^{ctrl} for four prototype applications [23]–[26] with the operating clock frequency range of $f \in [10, 400]$ MHz. The estimation error is less than 1% of the post-layout simulation results not included here due to the space limit. However, as mentioned earlier in this section, neither the simulation results nor the estimation results used for validation are of the actual controllers, instead they were based on the abstracted RTL module of the controllers. As mentioned, the abstracted RTL module represents up to 95% of the actual controllers, which typically contribute less than 5% of the decoders' EC. Hence, the above-mentioned inaccuracy of using the abstracted RTL module is acceptable.

D. Energy estimation of the memories

For the memories, the databook provided by the standard library developer [27] provides specifications, which allow the EC to be calculated. According to the TSMC 90 nm databook [27], the power consumption of a particular memory module size can be estimated by considering both the accessing rate a in units of accesses per clock cycle, as well as the clock frequency f and the supply voltage v . According to [27], memory writing and reading operations may be considered to have the same EC. In the standard cell library, the power consumption of the SRAM used in the architecture can be estimated using the reference table of [27]. In the reference table, the typical memory access power consumption p_a and leakage current I_l are given for memory blocks having various sizes and operand-widths. The power consumption P_a can be used for calculating the dynamic EC, when the memory is being accessed. The leakage current I_l can be used for calculating the static EC of the memory, when it is idle. However, the reference table only provides the reference data for typical supply voltages, hence, the voltage scaling factor $\frac{v^2}{1.2^2}$ used for the previous equations can still be applied. In this case, the typical specifications of the TSMC 90 nm SRAM operating at 1.2 V are used.

To estimate the memories' EC, the specific memories required by the proposed architecture are divided into two types, namely, the LLR memory blocks and the metric-storage memory block. Furthermore, the LLR memories in the turbo decoding scheme of Figure 1 are divided into three groups. The *a priori* LLR memories with indices 1 to k are defined as Group-1. The *a priori* LLR memories with indices $(k+1)$ to $(k+n)$ are defined as Group-2. Finally, the extrinsic LLR memories with indices 1 to k are defined as Group-3.

Based on the specifications provided by the databook [27], for a particular memory block 'M', the typical EC per clock cycle can be calculated as

$$E_{cyc}^M = \frac{(\frac{v^3}{1.2^2} f p_a a_{(M)} + v I_l) \times 10^{-3}}{f}, \quad (15)$$

where $a_{(M)}$ is the accessing rate of the particular memory block in the decoder. The variable (M) defines the four possible types of memories, namely the metric memory m , the memory in Group-1 ($g1$), the memory in Group-2 ($g2$) and the memory in Group-3 ($g3$). The calculation of $a_{(M)}$ is summarized in Table V. As a result, the EC for the particular

TABLE V
SUMMARY OF THE $a_{(M)}$ VALUES OF EQUATION 15.

M	$a_{(M)}$
m	$\frac{4w_s}{w_s(T_{fw}+T_{bw})+w_p T_{pbw}}$
$g1$	$\frac{2w_s+w_p}{w_s(T_{fw}+T_{bw})+w_p T_{pbw}}$
$g2$	$\frac{2w_s+w_p}{2(w_s(T_{fw}+T_{bw})+w_p T_{pbw})}$
$g3$	$\frac{3w_s+w_p}{2(w_s(T_{fw}+T_{bw})+w_p T_{pbw})}$

memory block 'M' can be calculated as

$$E_b^M = B \times T_e \times E_{cyc}^M. \quad (16)$$

There is one metric memory block, k memory blocks in Group-1, n memory blocks in Group-2 and k memory blocks in Group-3. Therefore, the total EC of the memories in the decoder is

$$E_b^{Mem} = E_b^m + kE_b^{g1} + nE_b^{g2} + kE_b^{g3}. \quad (17)$$

Since the energy model of the memories is provided by the manufacturer, our simulation results not included in here show that the estimation error becomes less than 0.5% compared to the post-layout simulation results, when the memory blocks are not embedded into any other circuit structure. Figure 5 gives both the simulation results and the estimation results of an 128×64 bits SRAM module, in order to verify this memory energy model.

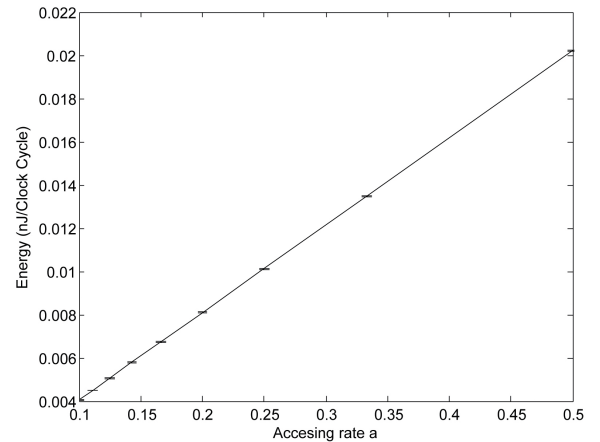


Fig. 5. The error bar result of 128×64 bits memory, $v = 1.2$ V.

E. Energy estimation of the interleaver

The interleaver is typically designed independently of the TC. As a result, it is not possible to devise a general model for estimating the EC of the interleaver in a turbo decoder, owing to the many different types of interleavers that can be used. However, the rate at which the interleaver is required to generate addresses is relatively low in the proposed architecture. As a result, it is straightforward to implement a low-complexity interleaver, having an insignificant EC compared to the turbo decoder. Therefore, a less accurate estimation of the interleaver's EC does not significantly impact the overall estimation accuracy of the proposed framework. To simplify the EC estimation of the interleaver, further assumptions may have to be made for the framework employed. Firstly, the interleaver may be limited to supporting only a single length. Secondly, the LTE interleaver design may be chosen for the estimation. These assumptions allow a relatively simple EC model to be obtained for the interleaver and are reasonable for WSN applications. The simulation and estimation results presented in this section will demonstrate that due to the low address generation speed requirement of the proposed architecture, the EC of the interleaver is insignificant.

The EC of the LTE interleaver is affected by the interleaver length N and the address generation rate g . Similarly to the modeling methods that were proposed for the register banks and the CU in Section III-B, the EC of the interleaver can be estimated in terms of nJ/Clock Cycle as

$$E_{\text{cyc}}^{\text{Int}} = \frac{v^2}{1.2^2} (0.9382g + 0.4359) \times 10^{-3}, \quad (18)$$

where g is calculated as

$$g = \frac{2w_s + w_p}{w_s(T_{\text{fw}} + T_{\text{bw}}) + w_p T_{\text{pbw}}}. \quad (19)$$

Finally, the EC of the interleaver normalized to represent the decoding of a single bit of information is

$$E_b^{\text{Int}} = B \times T_e \times E_{\text{cyc}}^{\text{Int}}. \quad (20)$$

To validate the model, we compared the estimation results and the post-layout simulation results not included here, for the interleaver considered for the four different interleaver lengths of $N = [512, 1024, 2048, 4096]$, for address generation rates of $g \in [0, 0.5]$ and for the operating clock frequency range of $f \in [10, 400]$ MHz. The results show that the maximum error of the estimation is 1.11%.

Note that the LTE interleaver employs a Quadratic Polynomial Permutation (QPP) design [28], having particular parameters f_1 and f_2 . More specifically, the LTE interleaver calculates the interleaved position of the LLR with index i according to

$$\pi(i) = (f_1 i + f_2 i^2) \mod N,$$

where N is the interleaver length. This operation is similar to that of the WiMAX interleaver, which employs an Almost Regular Permutation (ARP) design [28], according to

$$\pi(i) = [iP_0 + A + d(i \mod C)] \mod N,$$

where P_0 , A and $[d(j)]_{j=0}^{C-1}$ are parameters of the interleaver and C is a small number, such as 4 or 8. In the QPP and ARP

designs, the computational, storage and memory accessing demands are similar to each other. Furthermore, these demands are small compared to those of the LUT-Log-BCJR decoder, as we shall show in Section III-F. Owing to this, our analysis might be deemed to be sufficiently accurate for modeling all QPP and ARP interleaver designs. Note however, that non-deterministic interleaver designs, such as the S-random interleaver [29], have significantly higher storage demands than the deterministic QPP and ARP designs. For this reason, our model cannot be expected to provide an accurate energy estimation for non-deterministic interleavers. However, owing to their high storage demands, non-deterministic interleavers are rarely employed in practice.

F. Validation of the proposed framework

Using the above framework, the EC of a turbo decoder in nJ/bit can be estimated. The designer has the freedom to adjust all the parameters in Table I. For parameter v , the standard values of the TSMC 90 nm technology relying on $v = 1.2$ V can be used as the default value. Furthermore, we recommend the clock frequency's maximal value of $f = 400$ MHz, since this facilitates the highest decoding throughput and the lowest EC $E_{\text{cyc}}^{\text{M}}$ for the memories, as shown in (15). Although an iterative turbo decoder comprises a parallel concatenation of two BCJR decoders, these are operated alternately, rather than concurrently. Therefore, a single datapath can be employed to alternately support each of the two BCJR decoders. In addition to the datapath, the turbo decoder requires the controller, the memories and the interleaver of Sections III-B – III-E, respectively. When all the components are connected together to form a decoder, the chip layout will be adjusted for each individual implementation with the assistance of the Computer-Aided Design (CAD) tools of [30]. These adjustments cannot be predicted by the proposed framework. Therefore, to ascertain that these adjustments do not affect the accuracy of the estimation framework significantly, three different turbo code designs have been implemented for the sake of validation, as shown in Table VI. More specifically, we consider Design-I and Design-II from Section III-B, as well as an additional turbo code, which we refer to as Design-III. This employs component codes having the GP of the WiMAX turbo code [18], which corresponds to $k = 2$ inputs and $n = 2$ non-systematic outputs. All three considered designs employ block lengths of $N = 6144$ bits, in order to allow their comparison. Additionally, the parameter values of $z = 9$, $B = 10$, $f = 400$ MHz and $v = 1.2$ V were assumed in all cases. Table VI shows that in each case, our EC estimation is within 5% of the post-layout simulation result. We consider this accuracy to be sufficient for allowing the proposed framework to characterize a turbo decoder's EC in future studies, eliminating the need to carry out hardware design, synthesis, layout and simulation in order to estimate the EC. In each case, we found that the energy consumption of the interleaver represents less than 4% of the total turbo decoder energy consumption, as described in Section III-E.

TABLE VI
COMPARISON OF THE ESTIMATION RESULTS AND THE SIMULATION RESULTS OF THE ENERGY CONSUMPTIONS (NJ/BIT) OF THE EXAMPLE DESIGNS.

		Design-I	Design-II	Design-III
Specs	k	1	1	2
	m	2	3	3
	n	1	1	2
Simulation result		4.7686 nJ/bit	6.3955 nJ/bit	8.7326 nJ/bit
Estimation result		4.4244 nJ/bit	6.0826 nJ/bit	8.5146 nJ/bit

IV. HOLISTIC DESIGN METHOD

Based on the energy estimation framework of Section III, a holistic TC design method is proposed in this section for optimizing the overall EC. The particular design example of [6] is invoked for presenting our holistic design method. However, the approach adopted here is in contrast to that of [6], where a TC was designed by comparing different parametrizations relying on EXtrinsic Information Transfer (EXIT) charts and the BER performance alone. By contrast, in this contribution both E_b^{tx} and E_b^{pr} are considered during the design stage and a holistically energy-optimized design is created for the scheme considered.

A. Transmission energy estimation

In order to consider both E_b^{tx} and E_b^{pr} , an appropriate model is required for the estimation of E_b^{tx} . For example, the path-loss model of wireless communication relying on specifically chosen parameters based on the target scenario may be used. The path-loss model used in this paper has also been employed in [1], [3], [7], which is given by

$$P_1(d)[\text{dB}] = 20 \log_{10} \left(\frac{4\pi}{\lambda} \right) + 10p \log_{10}(d), \quad (21)$$

where $\lambda = c/f$ is the wave-length of the carrier, $c = 2.998 \times 10^8$ m/s is the speed of light, p is the path-loss exponent and d is the transmission distance. Furthermore, the environmental parameters and WSN system specifications of Table VII are assumed, where $N_0 = 10 \times \log_{10}(k \cdot T) = -203.8$ dBJ, with $k = 1.3806503 \times 10^{-23}$ being the Boltzmann constant and $T = 300\text{K}$ the room temperature. Finally, according to [3],

TABLE VII
ENVIRONMENT ASSUMPTIONS AND SYSTEM SPECIFICATION OF THE ESTIMATED WSN.

Transmission frequency (f)	5.8 GHz
Power amplifier efficiency loss (A)	4.81 dB
Receiver noise figure (RNF)	4 dB
Path loss exponent (p)	4
BER target	10^{-4}
Uncoded system minimum received SNR at the target BER (S_0)	34 dB
Temperature	300 K
Thermal noise (N_0)	-203.8 dBJ

the transmission energy expressed in J/bit is given by

$$E_b^{\text{tx}} = 10^{(N_0 + S_0 + \text{RNF} + P_1 + A - G)/10}, \quad (22)$$

where G is the coding gain provided by the TC employed, which may be quantified using conventional BER analysis.

Naturally, the coding gain G is a function of the TC parameters, such as its GP, interleaver design and the parameters of Table I.

For a real design, the parameters of Table VII have to be determined based on the specific target scenario considered. As shown in Table VII, we assume a power amplifier efficiency loss A of 4.81 dB, which corresponds to a power amplifier efficiency of 33%. This is typical of Class A/B amplifiers, as shown in [1, Table 3], which compares various different amplifier designs.

B. Overall energy estimation

Again, to demonstrate the estimation of E_b^{tx} and E_b^{pr} for the sake of determining the parametrization of a TC for a particular scenario, the design of [6] is chosen as an example. There were 36 candidate parametrizations of MCTCs and TCTCs in [6], as shown in Table VIII. The interleaver length of all the

TABLE VIII
THE CHOSEN TC DESIGNS.

candidate	k	m	n	R	B	polynomial	C
sysTCTC-1	1	3	1	1/3	3	$(17, 15)_o$	24
sysTCTC-1	1	3	1	1/3	6	$(17, 15)_o$	48
sysTCTC-1	1	3	1	1/3	12	$(17, 15)_o$	96
sysTCTC-2	1	3	1	1/4	3	$(17, 15)_o$	24
sysTCTC-2	1	3	1	1/4	6	$(17, 15)_o$	48
sysTCTC-2	1	3	1	1/4	12	$(17, 15)_o$	96
sysTCTC-3	1	3	1	1/5	3	$(17, 15)_o$	24
sysTCTC-3	1	3	1	1/5	6	$(17, 15)_o$	48
sysTCTC-3	1	3	1	1/5	12	$(17, 15)_o$	96
sysTCTC-4	1	3	1	1/6	3	$(17, 15)_o$	24
sysTCTC-4	1	3	1	1/6	6	$(17, 15)_o$	48
sysTCTC-4	1	3	1	1/6	12	$(17, 15)_o$	96
TCTC-1	1	3	1	1/3	3	$(10, 17)_o$	24
TCTC-1	1	3	1	1/3	6	$(10, 17)_o$	48
TCTC-1	1	3	1	1/3	12	$(10, 17)_o$	96
TCTC-2	1	3	1	1/4	3	$(10, 17)_o$	24
TCTC-2	1	3	1	1/4	6	$(10, 17)_o$	48
TCTC-2	1	3	1	1/4	12	$(10, 17)_o$	96
TCTC-3	1	3	1	1/5	3	$(10, 17)_o$	24
TCTC-3	1	3	1	1/5	6	$(10, 17)_o$	48
TCTC-3	1	3	1	1/5	12	$(10, 17)_o$	96
TCTC-4	1	3	1	1/6	3	$(10, 17)_o$	24
TCTC-4	1	3	1	1/6	6	$(10, 17)_o$	48
TCTC-4	1	3	1	1/6	12	$(10, 17)_o$	96
MCTC-1	1	2	1	1/3	6	$(4, 7)_o$	24
MCTC-1	1	2	1	1/3	12	$(4, 7)_o$	48
MCTC-1	1	2	1	1/3	24	$(4, 7)_o$	96
MCTC-2	1	2	1	1/4	6	$(2, 3)_o$	24
MCTC-2	1	2	1	1/4	12	$(2, 3)_o$	48
MCTC-2	1	2	1	1/4	24	$(2, 3)_o$	96
MCTC-3	1	2	1	1/5	6	$(2, 3)_o$	24
MCTC-3	1	2	1	1/5	12	$(2, 3)_o$	48
MCTC-3	1	2	1	1/5	24	$(2, 3)_o$	96
MCTC-4	1	2	1	1/6	6	$(2, 3)_o$	24
MCTC-4	1	2	1	1/6	12	$(2, 3)_o$	48
MCTC-4	1	2	1	1/6	24	$(2, 3)_o$	96

design candidates was $N = 2048$ and they were characterized using the BER performance. Their computational complexity was defined in terms of the number of trellis states 2^m and the number of iterations B as follows:

$$C = 2^m \cdot B. \quad (23)$$

Based on the comparison of the BER performance and the complexities, it was concluded that the MCTCs generally have a better performance than the corresponding TCTCs at all the complexities considered. The conclusions of [6] were inferred

from using the conventional TC design method and can be applied in conventional TC applications.

However, in this section we will demonstrate that when the EC is a major concern in a WSN target application, the conventional design method is sub-optimum, because we have to consider both E_b^{tx} and E_b^{pr} in the specific application scenario. By using the proposed framework, E_b^{pr} of each TC candidate listed in Table VIII can be estimated. Given a particular application scenario, the specifications of Table VII and the typical communication range d of the application can be taken into account. Therefore, using the BER results of [6] and the relevant path loss model, E_b^{tx} of each candidate listed in Table VIII can be estimated. Figure 6 shows the estimated results using the specifications given in Table VII for a WSN communication range of $d = 40$ m. The candidate designs characterized in Figure 6 are arranged in a descending order of the Signal-to-Noise Ratio (SNR) required for achieving $\text{BER} = 10^{-5}$ from left to right. In [6], the design MCTC-4 was recommended for situations where a complexity C of 96 or 48 can be afforded, since it facilitates a BER of 10^{-5} at the lowest SNR in these cases. When a complexity C of 24 can be afforded, [6] recommends MCTC-3, correspondingly. However, the results of Figure 6 show that neither MCTC-3 nor MCTC-4 offer the lowest overall EC $E_b = E_b^{\text{tx}} + E_b^{\text{pr}}$. Instead, the design sysTCTC-4 associated with $C = 48$ and sysTCTC-3 with $C = 48$ have the lowest overall EC amongst all the candidates. Indeed, these schemes offer a lower overall energy consumption than any of the schemes that were recommended in [6].

In Figure 7, the overall ECs are plotted versus the required SNRs, which are derived from the BER results and the computational complexities, respectively. It transpires from Figure 7 that neither of them has a direct relationship with the overall EC. Therefore, we conclude that neither the BER results nor the computational complexity facilitate an accurate EC $E_b = E_b^{\text{tx}} + E_b^{\text{pr}}$ prediction.

The case study of [6] offers a simple example for demonstrating the philosophy of the proposed holistic design method. Naturally, our assumptions concerning the propagation environment and the WSN system specifications were simplified for avoiding digression from the principles. Nonetheless, the proposed design method is capable of assisting the designer in optimizing a TC design in many different aspects. For example, apart from the basic TC parameters, the longest interleaver length N of a TC determines the memory requirement of the hardware implementation, which contributes a significant part of the total decoding EC. The number of decoding iterations performed has a significant effect on both the BER performance and on the decoder's EC. Additionally, the number of hops employed in a multi-hop network determines the average transmission range and the sensor densities. All of these aspects directly affect both the transmission EC and the decoding EC. As a result, the proposed design method can be used for optimizing a wide variety of related specifications for the sake of improving the system's energy efficiency.

Note that as in [3], our analysis assumes that the power amplifier and the turbo decoder are the only components of the transmitter and receiver that consume energy. In practice

however, energy will also be consumed by other baseband and Radio Frequency (RF) components, such as the turbo encoder, modulator, ADC/DAC, filters, oscillators, mixers, synchronizer, channel estimator, demodulator and low noise amplifier [31]. For the sake of simplicity and in order to adhere to the approach of [3], these components have been neglected in this analysis. However, they may be considered by employing $E_b = E_b^{\text{tx}} + E_b^{\text{pr}} + E_b^c$, where E_b^c is a constant that quantifies the total EC of the above listed components. An appropriate value may be selected for E_b^c using the discussions of [31]. Note however that adding the same constant value E_b^c to each of the overall EC results provided in Figure 7 would not change which particular scheme offers the lowest overall EC.

V. CONCLUSIONS

In this paper, we discussed the design of TCs in WSNs with the aim of reducing the overall EC. The importance of optimizing the TC at an early design stage was discussed, bearing in mind that both the transmission EC E_b^{tx} and the decoding EC E_b^{pr} have to be considered right from the commencement of the design. The conventional design method is capable of analyzing E_b^{tx} , the BER performance and the computational complexity during the design stage, but it is unable to consider the decoding EC. Therefore, a novel EC estimation framework based on the turbo decoder architecture of [7] was proposed for estimating the decoding EC during an early design stage. The EC estimation error was less than 5% compared to the post-layout simulation results. The proposed framework constitutes a novel holistic design method, which allows us to consider the overall EC $E_b^{\text{tx}} + E_b^{\text{pr}}$ for arbitrary TC designs during an early design stage. The wide-ranging TC design study of [6] was used for characterizing our design method. As a result, we showed that the holistic design method is capable of finding TC parametrizations optimized in terms of the overall EC for a particular application. Our future work will consider the generalization of the proposed framework to process technologies other than 90 nm.

REFERENCES

- [1] S. L. Howard, C. Schlegel, and K. Iniewski, "Error Control Coding in Low-Power Wireless Sensor Networks: When is ECC Energy-Efficient?" *EURASIP Journal of Wireless Communications and Networking, Special Issue: CMOS RF Circuits for Wireless Applications*, vol. 2006, Arti, pp. 1–14, 2006.
- [2] L. Li, R. Maunder, B. Al-Hashimi, and L. Hanzo, "An Energy-Efficient Error Correction Scheme for IEEE 802.15.4 Wireless Sensor Networks," *IEEE Transactions on Circuits and Systems II*, vol. 57, no. 3, pp. 233–237, Mar. 2010.
- [3] N. Sadeghi, S. Howard, S. Kasnavi, K. I. V. C. Gaudet, and C. Schlegel, "Analysis of Error Control Code Use in Ultra-Low-Power Wireless Sensor Networks," in *Proceedings of International Symposium on Circuits and Systems*, Island of Kos, 2006, pp. 3558–3561.
- [4] M. E. Pellenz, R. D. Souza, and M. Fonseca, "Error Control Coding in Wireless Sensor Networks," *Telecommunication Systems*, vol. 44, no. 1–2, pp. 61–68, 2009.
- [5] K. Doddapaneni, E. Ever, O. Gemikonakli, I. Malavolta, L. Mostarda, and H. Muccini, "Path Loss Effect on Energy Consumption in a WSN," in *2012 UKSim 14th International Conference on Computer Modelling and Simulation*. IEEE, Mar. 2012, pp. 569–574.
- [6] H. Chen, R. G. Maunder, and L. Hanzo, "An Exit-Chart Aided Design Procedure for Near-Capacity N-Component Parallel Concatenated Codes," in *Proceedings of the IEEE Global Telecommunications Conference GLOBECOM*. Miami, Florida, US: IEEE, Dec. 2010, pp. 1–5.

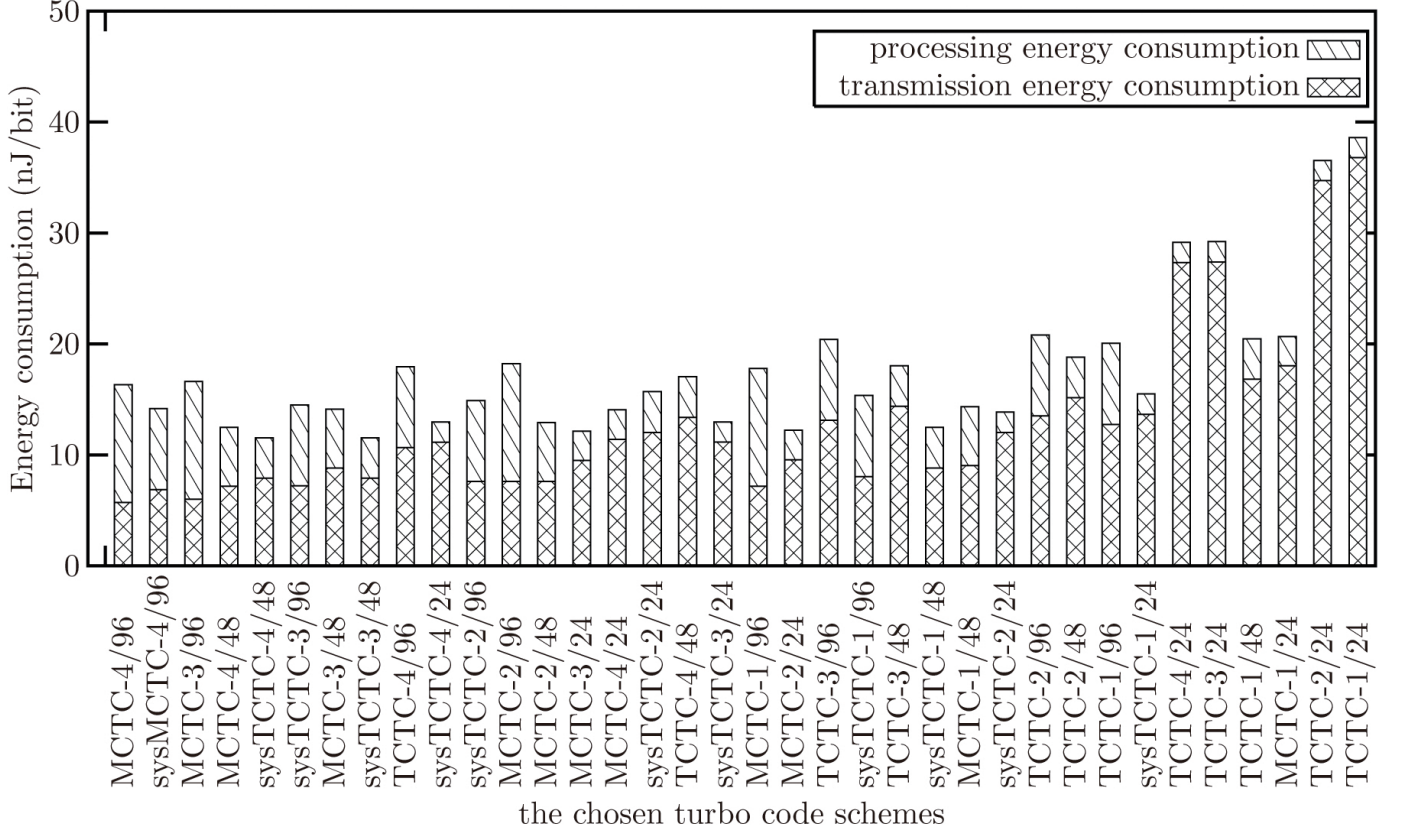


Fig. 6. Overall EC $E_b = E_b^{\text{tx}} + E_b^{\text{pr}}$ of the chosen schemes, when $d = 40$ m.

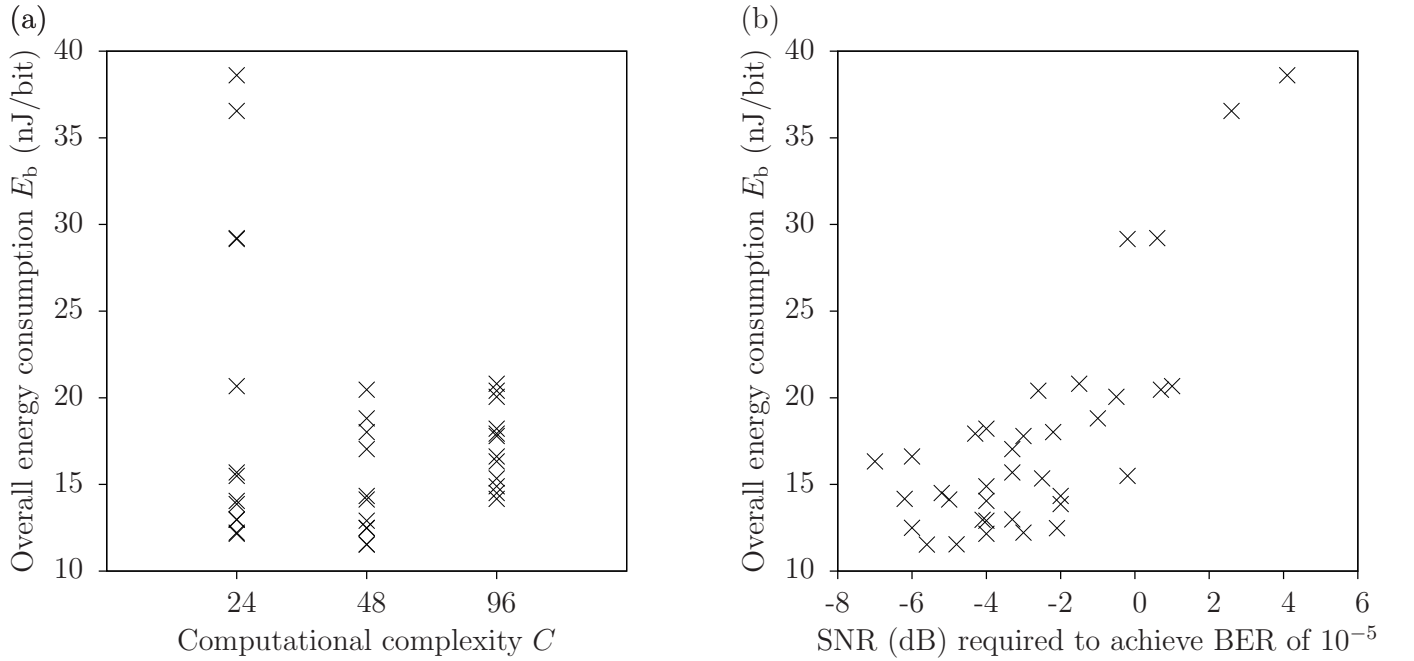


Fig. 7. Overall EC $E_b = E_b^{\text{tx}} + E_b^{\text{pr}}$ versus (a) the computational complexity C and (b) the SNR requirements of the chosen schemes.

- [7] L. Li, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A Low-Complexity Turbo Decoder Architecture for Energy-Efficient Wireless Sensor Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–9, 2011.
- [8] L. Hanzo, T. H. Liew, B. L. Yeap, R. Tee, and S. X. Ng, *Turbo Coding, Turbo Equalisation and Space-Time Coding*. John Wiley & Sons Inc, 2011.
- [9] L. Hanzo, J. P. Woodard, and P. Robertson, "Turbo Decoding and Detection for Wireless Applications," in *Proceedings of the IEEE*, vol. 95, no. 6, 2007, pp. 1178–1200.
- [10] M. Marandian, J. Fridman, Z. Zvonar, and M. Salehi, "Performance analysis of turbo decoder for 3GPP standard using the sliding window algorithm," in *12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2001, pp. 127–131.
- [11] C. Studer, S. Fateh, C. Benkeser, and Q. Huang, "Implementation trade-offs of soft-input soft-output MAP decoders for convolutional codes," *IEEE Trans. Circuits Syst. I*, vol. 59, no. 11, pp. 2774–2783, Nov. 2012.
- [12] P. Robertson, E. Vilebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. on Communications*, vol. 2, Seattle, WA, USA, June 1995, pp. 1009–1013.
- [13] 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. 3, pp. 284–287, 1974.
- [14] L. Li, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "Design of Fixed-Point Processing Based Turbo Codes Using Extrinsic Information Transfer Charts," in *Proceeding of IEEE Vehicular Technology Conference*, Ottawa, Canada, 2010, pp. 1–5.
- [15] C. Benkeser, A. Burg, T. Cupaiuolo, and Q. Huang, "Design and optimization of an HSDPA turbo decoder ASIC," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 98–106, Jan. 2009.
- [16] "Universal Mobile Telecommunications System (UMTS); Multiplexing and Channel Coding (FDD)," 2012.
- [17] *ETSI TS 136 212 LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding*, V10.2.0 ed., 2011.
- [18] *IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE 802.16-2004*, IEEE Std., 2004.
- [19] W.-K. Chen, *The VLSI Handbook*, 2nd ed. CRC Press, 2007.
- [20] B. Razavi, *Design of Analog CMOS Integrated Circuits*. Boston, MA, USA: McGraw-Hill, 2001.
- [21] A. Raghunathan, S. Dey, and N. K. Jha, "Register-Transfer Level Estimation Techniques for Switching Activity and Power Consumption," in *Proceedings of International Conference on Computer Aided Design*. IEEE Comput. Soc. Press, 1996, pp. 158–165.
- [22] P. Surti and L.-F. Chao, "Controller Power Estimation Using Information from Behavioral Description," in *1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96*, vol. 4. IEEE, pp. 679–682.
- [23] D.-F. Zhao, Y.-P. Wu, and N.-N. Tong, "The Applied Research of Convolutional Turbo Code Based on WiMAX Protocol," in *4th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, Oct. 2008, pp. 1–3.
- [24] Q. Li and N. S. Ramesh, "Channel Coding Performance in CDMA2000 Systems," in *IEEE Emerging Technologies Symposium on Broadband, Wireless Internet Access. Digest of Papers (Cat. No.00EX414)*. IEEE, 2000, p. 5.
- [25] X.-M. Yu, Y.-M. Kang, and D.-F. Yuan, "Performance Analysis of Turbo Codes in Wireless Rician Fading Channel with Low Rician Factor," in *IEEE 12th International Conference on Communication Technology*. IEEE, Nov. 2010, pp. 48–51.
- [26] D. Divsalar and F. Pollara, "Turbo Codes for Deep-Space Communications," Tech. Rep., 1995.
- [27] "TSMC 90nm Low Power High Density Synchronous Single Port with Redundancy SRAM Compiler Databook," 2007.
- [28] A. Nimbalkar, Y. Blankenship, B. Classon, and T. K. Blankenship, "ARP and QPP interleavers for LTE turbo coding," in *Proc. IEEE Wireless Commun. Networking Conf.*, Las Vegas, NV, USA, Mar. 2008, pp. 1032–1037.
- [29] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations," *Telecommunications and Data Acquisition Progress Report*, vol. 122, pp. 56–65, Apr. 1995.
- [30] "Encounter User Guide," 2005. [Online]. Available: http://www.cadence.com/tl/resources/datasheets/edi_system_ds.pdf
- [31] S. Cui, A. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2349–2360, Sept. 2005.