An Energy-Efficient Error Correction Scheme for IEEE 802.15.4 Wireless Sensor Networks

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

Abstract—**In this paper, we validate a novel augmentation to the physical layer (PHY) of the IEEE 802.15.4 standard for wireless sensor networks. This augmentation implements interleaving and forward error correction (FEC) encoding within sensor node transmitters, facilitating a significant reduction in their transmission energy. We detail the design, parameterization, and implementation of this FEC encoder and show that it has insignificant energy consumption compared with the transmission energy reduction that it affords. Our analysis shows that net energy savings of 24.8%–31.4% can be achieved by the augmented PHY.**

Index Terms—**Application specific integrated circuits, channel coding, communication systems, energy conservation, networks, sensors.**

I. INTRODUCTION

THE sensor nodes of a wireless sensor network (WSN) are typically required to maintain sporadic but reliable data transmissions for extended periods of time. However, in applications such as body area networks [1], the sensor nodes have to be small, preventing the use of bulky batteries. Therefore, improvements in the energy efficiency of sensor nodes are desirable.

Starred WSN topologies have been shown to be beneficial in energy-constrained applications [2], [3]. Here, the sensor nodes transmit to a central node, which coordinates the reactions of a higher-level system to the sensed data. Owing to its integration into the higher-level system, the central node typically has less limited energy resources. It is therefore beneficial to redistribute the energy consumption from the sensor nodes to the central node whenever possible.

In [4], we proposed an augmentation to the physical layer (PHY) of the IEEE 802.15.4 standard for WSNs [5] to redistribute energy consumption as described above. This was achieved by employing a sophisticated forward error correction (FEC) decoder within the central node, which reduced the transmission energy required to achieve reliable communications. However, additional energy is consumed when performing FEC encoding in the sensor node transmitters. The novel contribution of this paper is the design, parameterization, and implementation of a dedicated FEC hardware module for sensor node transmitters. This allows the net energy saving to be
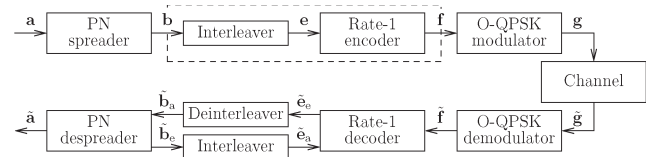
Fig. 1. Schematic of the augmented IEEE 802.15.4 PHY that was proposed in [4].

quantified by considering the signal-processing-related energy increase and the achievable transmission energy reduction.

The rest of this paper is organized as follows. Section II reviews the augmented IEEE 802.15.4 PHY that was proposed in [4]. In Section III, we detail a novel deterministic interleaver design that facilitates the practical implementation of the proposed PHY. In particular, this interleaver design is suitable for all possible PHY payload lengths without imposing an excessive memory requirement. A novel evolutionary algorithm (EA) [6] is employed for parameterizing the interleaver to maximize its performance, as detailed in Section IV. Section V discusses a novel hardware implementation of the FEC encoder, which employs parallel "just-in-time" processing to achieve a low processing latency and energy consumption. This energy consumption is analyzed in Section VI, and our analysis shows that it is insignificant compared with the transmission energy saving that it affords. As a result, the augmented PHY is shown to offer net energy consumption savings of 24.8%–31.4%. Finally, we offer our conclusions in Section VII.

II. AUGMENTED PHY

As detailed in [4], the proposed augmentation to the IEEE 802.15.4 PHY can be employed to convey the payloads of IEEE 802.15.4 data frames using a reduced transmission energy. However, the augmented PHY imposes additional interleaving and rate-1 encoding operations upon the sensor node transmitters. As shown in the schematic of Fig. 1, these operations are performed between the pseudonoise (PN) spreading and offset quadrature phase shift keying (O-QPSK) operations of the standard IEEE 802.15.4 PHY [5].

The augmented PHY in Fig. 1 applies PN spreading to the M-byte PHY payload $\mathbf{a}$, where $M \in [10 \ldots 127]$ [5, Sec. 6.3]. This is achieved by decomposing the payload $\mathbf{a}$ into sets of $k = 4$ consecutive bits and mapping these to $n = 32$-chip PN sequences [5, Tab. 24], like in the standard PHY. These PN sequences are concatenated to obtain the N-chip sequence $\mathbf{b}$, where $N = 8 Mn/k$. The interleaver of the augmented PHY is employed to rearrange the order of the chips in $\mathbf{b}$ and may be implemented as described in Sections III and IV. As shown in

Fig. 1, the resultant N-chip sequence $\mathbf{e}$ is rate-1 encoded [7], as detailed in Section III. Finally, the encoded chip sequence $\mathbf{f}$ is O-QPSK modulated, like in the standard PHY. As detailed in Section III, the input of the augmented PHY's O-QPSK modulator $\mathbf{f}$ comprises the same number N of chips as the output of the PN spreader $\mathbf{b}$, like in the standard PHY. For this reason, the PN spreader and the O-QPSK modulator remain completely unchanged when the augmented PHY is employed in the sensor node transmitters.

In the receiving central node, additional rate-1 decoding and deinterleaving operations are employed by the augmented PHY. This employs iterative decoding [8], which repeatedly alternates the operation of the PN despreader and the rate-1 decoder, as shown in Fig. 1. This is in contrast with the receiver of the standard PHY, which employs only the "one-shot" operation of the PN despreader. Since the augmented PHY invests more decoding complexity in the central node than the standard PHY, it can achieve a target payload error ratio (PER) by using a reduced sensor node transmission energy. Indeed, our simulation results [4] demonstrated this for the case of line-of-sight transmissions in the presence of additive white Gaussian noise having a constant power spectral density N_0. More specifically, when transmitting $N = 640$-chip payloads, the augmented PHY can achieve a desirable PER of 10^{-3} at a transmission energy per chip E_c that is 3.22 dB lower than that required by the standard PHY. Furthermore, this gain increases to 6.75 dB when $N = 8128$-chip payloads are employed owing to the augmented PHY's interleaver gain [4], which is obtained when transmitting longer payloads.

To assess the practical sensor node energy saving, the augmentation of the Chipcon CC2430 PHY [9] was considered in [4]. The energy consumed during transmission is given by $E_{\mathrm{tx}} = I_{\mathrm{tx}} \cdot V \cdot t_{\mathrm{tx}}$, where $t_{\mathrm{tx}} = N/f_{\mathrm{tx}}$ is the transmit duration, and the IEEE 802.15.4 transmission rate is $f_{\mathrm{tx}} = 2 \cdot 10^6$ chips per second [5]. As may be expected, the current I_{tx} consumed during the transmission of a data payload depends on the particular transmit energy per chip E_c employed. In its maximum transmit power mode of 0.6 dBm, the Chipcon CC2430 consumes $I_{\mathrm{tx}}^{\mathrm{std}} = 32.4$ mA [9, Tab. 45]. At this transmit power, the amount of energy $E_{\mathrm{tx}}^{\mathrm{std}} = I_{\mathrm{tx}}^{\mathrm{std}} \cdot V \cdot t_{\mathrm{tx}}$ consumed by the standard PHY without augmentation is illustrated in Fig. 2 for payloads comprising various numbers N of chips. As described above, the augmented PHY reduces the transmission energy required to achieve a desirable PER of 10^{-3} by 3.22–6.75 dB, depending on the length of the payload N. Corresponding reductions from the Chipcon CC2430's maximum transmit power of 0.6 dBm allow its current consumption $I_{\mathrm{tx}}^{\mathrm{aug}}$ to be lowered from 32.4 mA to 21.7–23.7 mA [9, Tab. 45]. Fig. 2 shows the amount of transmission energy $E_{\mathrm{tx}}^{\mathrm{aug}} = I_{\mathrm{tx}}^{\mathrm{aug}} \cdot V \cdot t_{\mathrm{tx}}$ consumed by the augmented PHY for various values of N. These results show that the augmented PHY facilitates *gross* sensor node energy savings of $(E_{\mathrm{tx}}^{\mathrm{std}} - E_{\mathrm{tx}}^{\mathrm{aug}})$ that are 27.0%–33.0% of $E_{\mathrm{tx}}^{\mathrm{std}}$, depending on the payload length N.

However, to determine the *net* sensor node energy saving $[E_{\mathrm{tx}}^{\mathrm{std}} - (E_{\mathrm{tx}}^{\mathrm{aug}} + E_{\mathrm{pr}}^{\mathrm{aug}})]$ that is afforded by the augmented PHY, it is necessary to additionally consider the energy $E_{\mathrm{pr}}^{\mathrm{aug}}$ consumed during the operation of the interleaver and rate-1 encoder that are boxed in Fig. 1. In the following sections,
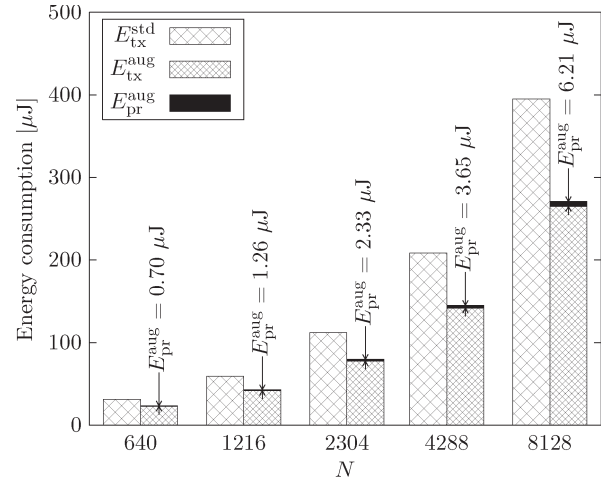

Fig. 2. Total energy consumed in the standard Chipcon CC2430 PHY $E_{\mathrm{tx}}^{\mathrm{std}}$ and in the proposed hardware implementation of the augmented PHY $(E_{\mathrm{tx}}^{\mathrm{aug}} + E_{\mathrm{pr}}^{\mathrm{aug}})$.

we detail the design, parameterization, implementation, and characterization of a hardware module for this purpose.

III. MODULE DESIGN

As described in Section II, the proposed module implements the interleaver and rate-1 encoder that are boxed in Fig. 1. Here, the standard IEEE 802.15.4 PN spreader [5, Sec. 6.5] provides the input chip sequence $\mathbf{b}$. As described in Section II, this comprises $8 M/k$ number of $n = 32$-chip PN sequences [5, Tab. 24], where M is the number of bytes in the data payload $\mathbf{a}$. Since this has 118 possible values $M \in [10 \ldots 127]$, there are 118 possible lengths $N = 8Mn/k \in \{640, 704, 768, \ldots, 8128\}$ for the chip sequence $\mathbf{b}$ [4].

When repositioning the chips in the sequence $\mathbf{b} = \{b_i\}_{i=0}^{N-1}$, the interleaver in Fig. 1 is required to desirably "randomize" the order of the chips in the resultant sequence $\mathbf{e} = \{e_i\}_{i=0}^{N-1}$. To achieve this, the interleaver must fully exploit the grade of freedom for repositioning the chips, which increases with the number of chips N. As a result, different interleaver designs are required for each of the 118 possible values of N, and the associated parameters of the interleaver design must be stored in ROM.

A naive interleaver design would be parameterized by 118 arrays $\{\pi_{640}, \pi_{704}, \pi_{768}, \ldots, \pi_{8128}\}$, each of which would comprise N unique integers in the range of $[0, N-1]$. The operation of the naive interleaver can formally be specified as $e_i = b_{\pi_N[i]}$, where $\pi_N[i]$ is the ith element in the array π_N, and $i \in [0, N-1]$. However, this approach would require approximately 800 kB of ROM, which we consider to be excessive, since memory accesses are typically associated with relatively high energy consumptions in systems on chip [10].

This problem was solved in the implementations detailed in [11]–[13] by employing deterministic interleaver designs. These require the storage of only a limited number of parameters, which are employed to compute the elements of the interleaver pattern in an online manner, as and when they are required. However, the designs detailed in [11]–[13] are optimized for turbo codes and are not suitable for the augmented

$\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \ldots,$ $\mathbf{r}_{253}\}$	Each of the 254 arrays $\mathbf{r}_u$ comprises $n = 32$ unique integers from the range $[0, n-1]$.
$\{s_{640}, s_{704}, s_{768},$ $\ldots, s_{8128}\}$	Each of the 118 integers s_N has a value from the range $[0, n-1]$.
$\{p_{640}, p_{704}, p_{768},$ $\ldots, p_{8128}\}$	Each of the 118 integers p_N has a value from the range $[1, N-1]$.
$\{W_{640}, W_{704}, W_{768},$ $\ldots, W_{8128}\}$	Each of the 118 integers W_N has a value from the range $[1, N]$.
$\{\mathbf{w}_{640}, \mathbf{w}_{704}, \mathbf{w}_{768},$ $\ldots, \mathbf{w}_{8128}\}$	Each of the 118 arrays $\mathbf{w}_N$ comprises W_N unique integers from the range $[0, W_N - 1]$.

PHY. This is because the interleaver is required to mitigate a higher level of correlation within the soft information exchanged in the augmented PHY since the PN despreader in Fig. 1 operates on relatively long blocks, comprising $n = 32$ chips.

For this reason, we propose a deterministic design that resembles a dithered relative prime (DRP) interleaver [14], which has been shown to effectively "randomize" the order of the chips in the sequence $\mathbf{b}$ without requiring an excessive amount of ROM. Indeed, only 12 kB of ROM is required to store the parameters of our interleaver design, as listed in Table I. Like a DRP interleaver, our design is implemented in three stages, which are referred to as "Interleaver 1," "Interleaver 2," and "Interleaver 3" in our discussions below. As exemplified by the crisscrossing arrows in Fig. 3(a), these interleavers are employed to "randomize" the order of the intermediate chip sequences $\mathbf{c}$, $\mathbf{d}$, and $\mathbf{e}$, respectively. Note that the intermediate chip sequences comprise the same number of chips as the input sequence $\mathbf{b}$, namely N.

Interleaver 1: Similar to the first stage of a DRP interleaver, Interleaver 1 in Fig. 3(a) employs a block-based rearrangement of the chips in the input sequence $\mathbf{b} = \{b_i\}_{i=0}^{N-1}$ to generate the sequence $\mathbf{c} = \{c_i\}_{i=0}^{N-1}$. More specifically, each block of $n = 32$ chips in $\mathbf{c}$ is provided by rearranging the order of the corresponding $n = 32$-chips PN sequence in $\mathbf{b}$. However, in contrast to a conventional DRP interleaver, a *different* rearrangement is employed for each $n = 32$-chips PN sequence, as specified by the parameters $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{253}\}$ in Table I and as shown in Fig. 3(a). Note that 254 rearrangements are required because the sequence $\mathbf{b}$ comprises $N/n = 254$ PN sequences when it has a maximal length of $N = 8128$ chips. The operation of Interleaver 1 can be formally specified as $c_i = b_{j_i}$, where

$$j_i = n \cdot u + \mathbf{r}_u[v] \tag{1}$$

$\mathbf{r}_u[v]$ is the vth element in the array $\mathbf{r}_u$, $v = i \bmod n$, $u = i \operatorname{div} n$, and $i \in [0, N-1]$. Here, the "div" operator indicates integer division, whereas "mod" is employed to represent the modulo operator.

Interleaver 2: Similarly, the operation of Interleaver 2 from Fig. 3(a) can be specified as $d_i = c_{j_i}$, where

$$j_i = (s_N + p_N \cdot i) \bmod N \tag{2}$$

and $i \in [0, N-1]$. As in the second stage of a conventional DRP interleaver, s_N identifies the index of the chip in $\mathbf{c}$ that provides the first chip in $\mathbf{d} = \{d_i\}_{i=0}^{N-1}$, as shown in Fig. 3(a). The subsequent chips in $\mathbf{d}$ are provided by employing succes-

sive hops of p_N chips (modulo N) to select the corresponding chip in the sequence $\mathbf{c}$. Here, p_N is required to be a relative prime of N to ensure that each chip in $\mathbf{c}$ provides exactly one chip for $\mathbf{d}$. Note that the particular values that are employed for s_N and p_N depend upon the length N of the chip sequence, as shown in Table I.

Interleaver 3: Similarly, the parameters employed for Interleaver 3 in Fig. 3(a) depend upon the length N of the chip sequence. This interleaver employs a block-based rearrangement of the chips in $\mathbf{d}$ to obtain the sequence $\mathbf{e}$, similar to Interleaver 1. However, in contrast to Interleaver 1, Interleaver 3 employs the *same* rearrangement for each block of W_N chips in the sequence $\mathbf{d}$, like in a conventional DRP interleaver. As shown in Table I, this rearrangement and the block length are described by the parameters $\mathbf{w}_N$ and W_N, respectively. Clearly, W_N is required to be a factor of the chip sequence length N. For example, $W_{640} = 16$, $W_{1216} = 32$, $W_{2304} = 64$, $W_{4288} = 64$, and $W_{8128} = 64$. The operation of Interleaver 3 can be formally specified as $e_i = d_{j_i}$, where

$$j_i = W_N \cdot u + \mathbf{w}_N[v] \tag{3}$$

$\mathbf{w}_N[v]$ is the vth element in the array $\mathbf{w}_N$, $v = i \bmod W_N$, $u = i \operatorname{div} W_N$, and $i \in [0, N-1]$.

Rate-1 Encoder: Finally, for each chip in the sequence $\mathbf{e} = \{e_i\}_{i=0}^{N-1}$, the rate-1 encoder in Fig. 1 generates one chip for its output sequence $\mathbf{f} = \{f_i\}_{i=0}^{N-1}$. More specifically, $f_0 = e_0$ and $f_i = e_i \oplus f_{i-1}$ for $i \in [1, N-1]$, as shown in Fig. 3(a), in which $\oplus$ indicates the modulo-2 addition of two binary chips. As a result, the output chip sequence $\mathbf{f}$ input to the standard IEEE 802.15.4 O-QPSK modulator [5, Sec. 6.5.2.4] also comprises N chips.

IV. MODULE PARAMETERIZATION

Let us now describe the offline algorithm employed to design values for the interleaver parameters in Table I to ensure that the order of the chips in the sequence $\mathbf{b}$ is effectively "randomized." Note that the N-chips input sequence $\mathbf{b}$ that is output by the $k/n = 1/8$-rate PN spreader in Fig. 1 has $2^{Nk/n}$ legitimate permutations [5, Tab. 24]. As described above, the module detailed in this section maps each of these permutations to a different permutation of the output chip sequence $\mathbf{f}$. The particular mapping that is employed depends upon the parameters of the interleavers. Our offline algorithm used for designing these parameters attempts to maximize the minimum Hamming distance $d_H^{\min}$ between the legitimate permutations of $\mathbf{f}$, as we shall detail below. This way, the number of chip errors that is required to transform the transmitted permutation of $\mathbf{f}$ into any other legitimate permutation is maximized. This maximizes the probability that transmission errors can be detected and corrected by the iterative decoder in Fig. 1, optimizing its performance.

Although it is beyond the scope of this paper, it can be shown that $d_H^{\min}$ can be as low as 6 if the interleaver does not effectively "randomize" the order of its chips. However, it can be shown that $d_H^{\min}$ will increase to at least 24 provided that the interleaver parameterization satisfies two conditions. First, the interleaver should maximize the minimum positional separation between any two chips in $\mathbf{e}$ that originate from the
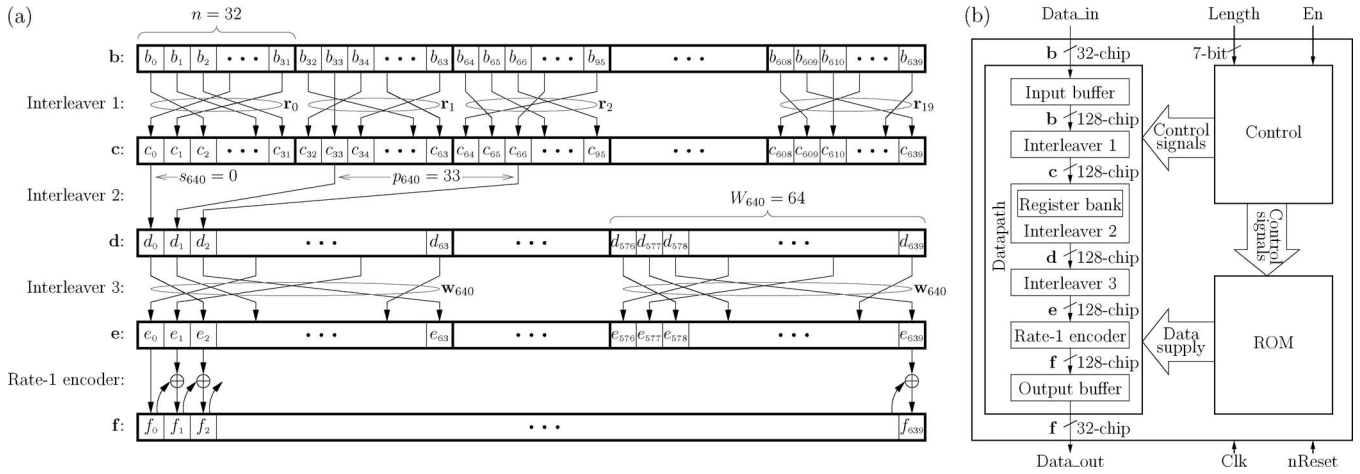
Fig. 3. (a) Example operation of the interleaver and rate-1 encoder of Fig. 1 for $N = 640$. (b) Schematic of the proposed hardware implementation.

same $n = 32$-chips PN sequence in $\mathbf{b}$. Second, the interleaver should minimize the maximum number of occurrences of any two $n = 32$-chips PN sequences in $\mathbf{b}$ having chips that are positioned next to each other in $\mathbf{e}$.

To generate interleavers that achieve the above-described goals, we designed a novel EA [6] to select beneficial values for the parameters s_N, p_N, W_N, and $\mathbf{w}_N$ associated with each chip sequence length $N \in [640, 704, 768, \dots, 8128]$, as detailed in Table I. Commencing from random choices, the parameter values were mutated in each generation of our EA. Any mutations that were found to improve the interleaver's ability to achieve the above-described goals were retained for the next generation of the EA. This process continued until no further improvements could be found within a reasonable amount of time. Note that random values were selected for the parameters $\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{253}\}$ since they employ the same values regardless of the chip sequence length N and cannot be optimized for any particular value of N.

V. MODULE IMPLEMENTATION

In this section, we describe a hardware module that implements the interleaver and the rate-1 encoder that are boxed in Fig. 1. The schematic in Fig. 3(b) is employed for the proposed module, which could be integrated between the PN spreader and the O-QPSK modulator of a standard IEEE 802.15.4 implementation, such as the Chipcon CC2430. A timing diagram for the proposed implementation is provided in Fig. 4. In the following discussions, we shall detail the proposed module's I/O interface, datapath, ROM, and controller.

The proposed module is specifically designed to avoid imposing any changes upon the I/O interfaces of the standard PN spreader and O-QPSK modulator. These exchange $n = 32$-chips PN sequences [5, Sec. 6.5.2.4] at a rate of $f_{\text{tx}}/n = 62.5 \cdot 10^3/\text{s}$, where $f_{\text{tx}} = 2 \cdot 10^6$ chips/s, as described in Section II. These features motivate our module's employment of a $f_{\text{pr}}^{\text{aug}} = 62.5$ kHz clock, which is supplied using the "Clk" port shown in Fig. 3(b), as well as the 32-chip I/O ports "Data_in" and "Data_out." As a result, $N/32$ clock cycles are required to clock the N-chip sequences $\mathbf{b}$ and $\mathbf{f}$ into and out of the proposed implementation, respectively, as shown in Fig. 4.
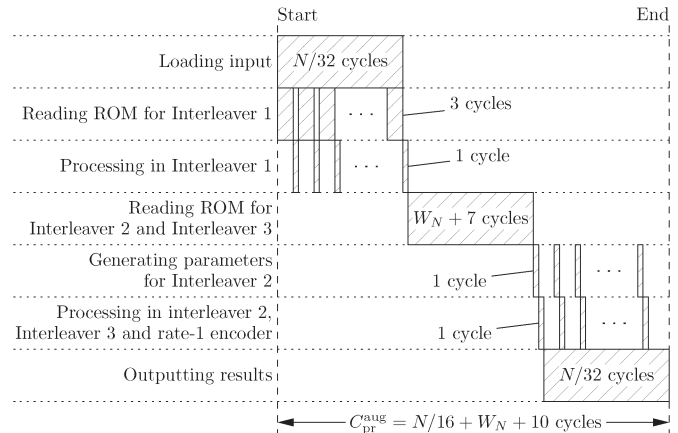

Fig. 4. Timing diagram of the proposed hardware implementation.

Note that the proposed module has three other ports, as shown in Fig. 3(b). The module begins operating when a logic 1 is placed upon the "En" port, allowing its input port to be synchronized with the PN spreader's output port. As described in Section III, the module operates in 1 of 118 different modes, depending on the length N of the chip sequence $\mathbf{b}$. The value of N can be extracted from the 7-bit "frame-length field" employed in the IEEE 802.15.4 PHY header [5, Sec. 6.5.3], which conveys the number of bytes in the PHY payload $\mathbf{a}$, namely, M. This "frame-length field" is provided to the module using its 7-bit "Length" port. Finally, the "nReset" port may be used to reset the registers employed within the proposed module.

A serial structure is employed within the datapath block of the proposed module, as shown in Fig. 3(b). Here, the interleaver in Fig. 1 is implemented in three stages, namely, Interleaver 1, Interleaver 2, and Interleaver 3, as described in Section III. These stages interleave multiple chips in parallel [15] to process the chips at the same rate that they are supplied by the PN spreader, as shown in Fig. 4. This approach facilitates a low processing latency and "just-in-time" processing, which reduces the number of registers that are required to store intermediate results.

A uniform 128-chip dataflow width is chosen within the datapath block in Fig. 3(b). This uniform width allows Interleaver 2, Interleaver 3, and the rate-1 encoder to be operated within a

single clock cycle without the need for intermediate registers, as shown in Fig. 4. However, owing to its 8128-chip register bank, Interleaver 2 cannot be tightly connected to Interleaver 1. More specifically, this register bank must collect all of the chips from the sequence **c** before Interleaver 2 can commence generating the sequence **d** owing to the relative prime-number-based hops that are required, as discussed in Section III. The 128-chip input and output buffers shown in Fig. 3(b) are required owing to the different port widths employed inside and outside of the datapath block.

As described in Section III, the three interleaving stages are parameterized as described in Table I. In the proposed module, these parameters are stored in the ROM block shown in Fig. 3(b). Combinational logic is employed to convert these parameters into chip indices j_i according to (1)–(3) for Interleaver 1, Interleaver 2, and Interleaver 3, respectively. The number of clock cycles required to read these parameters from ROM is shown in Fig. 4. Note that separate clock cycles are required to perform the calculations in (2), which are performed recursively according to

$$j_i = (j_{i-1} + p_N) \bmod N \tag{4}$$

where $i \in [1, N-1]$, and $j(0) = s_N$.

VI. ENERGY CONSUMPTION ANALYSIS

In this section, we shall consider the energy consumption of the module detailed in Section V. We shall estimate the effect that integrating this module into the Chipcon CC2430 hardware [9] would have upon its total energy consumption.

The Synopsys Design Complier was employed to synthesize a gate-level implementation of the module detailed in Section V. Our synthesis additionally employed a STMicroelectronics 0.12-μm technology standard cell library, resulting in a 1.6-mm^2 chip area, including the ROM. Synopsys PrimeTime was employed to determine the resultant implementation's average current consumption, which was found to be $I_{pr}^{aug} = 222.3$ μA. We assume that our proposed module consumes no current when it is deactivated by placing a logic zero upon its En port, as shown in Fig. 3(b). Hence, the duration t_{pr}^{aug} for which the proposed module consumes current is given by $t_{pr}^{aug} = C_{pr}^{aug}/f_{pr}^{aug}$, where the clock frequency is $f_{pr}^{aug} = 62.5$ kHz and the total number of clock cycles employed is $C_{pr}^{aug} = N/16 + W_N + 10$, as shown in Fig. 4. Finally, the proposed module's energy consumption is given by $E_{pr}^{aug} = I_{pr}^{aug} \cdot V \cdot t_{pr}^{aug}$, where a supply voltage of $V = 3$ V is assumed, like in Section II.

Fig. 2 provides the energy consumed by the proposed module E_{pr}^{aug} for payloads comprising various numbers N of chips. The resultant net energy savings $[E_{tx}^{std} - (E_{tx}^{aug} + E_{pr}^{aug})]$ afforded by employing the augmented PHY are 24.8%–31.4% of E_{tx}^{std}.

VII. CONCLUSION

In this paper, we have considered the augmentation of the IEEE 802.15.4 PHY that was proposed in [4]. This significantly reduces the transmission energy required to achieve a target PER at the cost of requiring some additional processing within the sensor nodes. We have proposed a dedicated hardware module for performing this processing and detailed its design, parameterization, implementation, and energy consumption analysis. This analysis revealed that the energy consumed by the module is modest compared with the transmission energy saving that it facilitates. For this reason, we can conclude that sophisticated FEC techniques are desirable for future WSN PHY standards.

REFERENCES

[1] S. Drude, "Requirements and application scenarios for body area networks," in *Proc. IST Mobile Wireless Commun. Summit*, Budapest, Hungary, Jul. 2007, pp. 1–5.

[2] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks," in *Proc. IEEE Int. Conf. Perform. Comput. Commun.*, Phoenix, AZ, Apr. 2004, pp. 701–706.

[3] O. Omeni, A. Wong, A. J. Burdett, and C. Toumazou, "Energy efficient medium access protocol for wireless medical body area sensor networks," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 4, pp. 251–259, Dec. 2008.

[4] R. G. Maunder, A. S. Weddell, G. V. Merrett, B. M. Al-Hashimi, and L. Hanzo, "Iterative decoding for redistributing energy consumption in wireless sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun. Netw.*, St. Thomas, U.S. Virgin Islands, Aug. 2008, pp. 1–6.

[5] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std. 802.15.4, Sep. 2006.

[6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[7] D. Divsalar, S. Dolinar, and F. Pollara, "Serial concatenated trellis coded modulation with rate-1 inner code," in *Proc. IEEE Global Telecommun. Conf.*, San Francisco, CA, Nov. 2000, vol. 2, pp. 777–782.

[8] S. Benedetto and G. Montorsi, "Iterative decoding of serially concatenated convolutional codes," *Electron. Lett.*, vol. 32, no. 13, pp. 1186–1188, Jun. 1996.

[9] Chipcon, A True System-on-Chip Solution for 2.4 GHz IEEE 802.15.4/ ZigBee(TM) Datasheet, Oslo, Norway, Jun. 2007.

[10] T. Okuma, M. Muroyama, and H. Yasuura, "Reducing access energy of on-chip data memory considering active data bitwidth," in *Proc. Int. Symp. Low Power Electron. Des.*, Monterey, CA, Aug. 2002, pp. 88–91.

[11] Z. Wang and Q. Li, "Very low-complexity hardware interleaver for turbo decoding," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 7, pp. 636–640, Jul. 2007.

[12] J.-H. Kim and I.-C. Park, "Double-binary circular turbo decoding based on border metric encoding," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 1, pp. 79–83, Jan. 2008.

[13] M. Martina, M. Nicola, and G. Masera, "A flexible UMTS-WiMax turbo decoder architecture," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 4, pp. 369–373, Apr. 2008.

[14] S. Crozier and P. Guinand, "High-performance low-memory interleaver banks for turbo-codes," in *Proc. IEEE Veh. Technol. Conf.*, Atlantic City, NJ, Oct. 2001, vol. 4, pp. 2394–2398.

[15] R. Dobkin, M. Peleg, and R. Ginosar, "Parallel interleaver design and VLSI architecture for low-latency MAP turbo decoders," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 4, pp. 427–438, Apr. 2005.