A 5.4 μW Soft-Decision BCH Decoder for Wireless Body Area Networks

Chia-Hsiang Yang, *Member, IEEE*, Ting-Ying Huang, Mao-Ruei Li, and Yeong-Luh Ueng, *Member, IEEE*

***Abstract*—This paper presents an IEEE 802.15.6 compliant soft-decision BCH decoder for energy-constrained wireless body area networks. The proposed soft-decision decoder (SDD) provides a 1 dB coding gain compared to the hard-decision decoder (HDD). The improvement in BER performance can translate into power savings at the transmitter. The energy dissipation and area of the soft-decision BCH decoder is minimized by jointly considering the algorithm, architecture, and circuit parameters. An early termination strategy is proposed to reduce the number of redundant test patterns. Probabilistic sorting is proposed to determine the test patterns, and its hardware complexity is only 54.7% of the conventional sorting method. The HDD kernel is implemented by adopting the Peterson rule, reducing the area by 44.2%. A pass-transistor logic based Chien search circuit consumes 33.3% less energy compared to the standard-cell based implementation. The chip is designed to operate at the minimum energy point of 0.29 V, yielding an energy reduction of 94% compared to a direct-mapped SDD at $\mathrm{SNR} = 5$ dB. Fabricated in 90 nm CMOS, the chip dissipates 5.4 μW at 500 kHz, achieving a throughput of 6.38 Mbps.**

***Index Terms*—BCH code, IEEE 802.15.6, integrated circuits, power-area minimization, soft-decision decoding, WBAN.**

I. Introduction

RECENTLY, wireless body area network (WBAN) technology has drawn wide attention owing to the demand for short-range wireless communications and the advent of energy-efficient VLSI technology. The WBAN supports connections between a variety of devices, such as consumer and biomedical devices. For healthcare applications, physiological signals can be sensed and monitored through a WBAN to diagnose chronic diseases and to trigger emergency alarms.

Error-correction codes are used to reduce the error probability and provide a reliable communication link [1]. The Bose-Chaudhuri-Hocquenghem (BCH) code [1] is an error correction code that is commonly used in storage and communication systems [2]–[5]. In the IEEE 802.15.6 standard [6], the (63, 51) BCH code and its shortened (31, 19) code are adopted to enhance transmission reliability under different channel conditions. Either hard-decision decoding/decoders (HDD) or soft-decision decoding/decoders (SDD) can be adopted at the receiver. In general, SDD can achieve a better error performance compared to HDD, and the improvement in BER performance of the SDD can translate to power savings at the transmitter, given the same data link requirements. Consequently, the use of SDD can increase the energy efficiency of the network [7].

In [8], Chase proposed SDD by evaluating several test patterns using the HDD kernel in order to improve the error-rate performance. For the type-II Chase-based SDD, there are a total of 2^t test patterns that should be considered, where t denotes the error-correction capability of the code. Although the Chase-based SDD can achieve a better error performance compared to the HDD, the SDD also introduces additional energy dissipation and hardware complexity. In this paper, several techniques are proposed in order to mitigate the incremental complexity and minimize the energy consumption.

In the proposed SDD, an early termination (ET) scheme is used to reduce the number of unnecessary test patterns and, thereby reducing unnecessary energy dissipation. In contrast to the parallel syndrome-checking architecture used in [9], a serial syndrome-checking architecture with two termination criteria is used in the proposed ET scheme. A probabilistic sorting scheme is utilized to reduce the area of the sorting circuit. Compared to the method presented in [10], the proposed probabilistic sorting scheme can significantly reduce the architectural complexity with negligible degradation in error-rate performance. To reduce the energy dissipation in the core Chien search module, pass-transistor logic is used to extend the operational range of the MOS logic [11]. The proposed area-efficient XOR gate is more robust against process variations than the circuit presented in [12] for subthreshold operation. In order to minimize the energy dissipation, the circuit is designed to operate at the minimum energy point [13], [14].

The proposed SDD provides coding gains of 0.75 and 1 dB for (63, 51) and (31, 19) codes, respectively, compared to the HDD. Compared to a lookup table (LUT)-based HDD, the proposed SDD has a comparable area and dissipates less than 25% of the energy. A 94% energy saving is achieved compared to the direct-mapped SDD with a LUT-based HDD kernel. Fabricated in a 90 nm CMOS process, the BCH decoder chip meets the throughput specifications of the IEEE 802.15.6 standard and dissipates 5.4 μW.

The remainder of this paper is organized as follows. In Section II, a brief review of finite field arithmetic, BCH codes, and the associated decoding algorithms is given. In Section III, the techniques adopted for the area and energy-efficient architecture are presented. Circuit design for minimum energy operation and its realization are discussed in Section IV. The chip implementation and measurement results are shown in Section V. Finally, Section VI concludes the paper.

Manuscript received August 29, 2013; revised December 28, 2013, February 05, 2014, and March 06, 2014; accepted March 07, 2014. Date of publication April 25, 2014; date of current version August 26, 2014. This research was supported by the National Chip Implementation Center, Taiwan. This paper was recommended by Associate Editor Z. Zhang.

C.-H. Yang is with the Department of Electronics Engineering, National Chiao Tung University, 30013 Hsinchu, Taiwan (e-mail: chy@nctu.edu.tw).

T.-Y. Huang was with the Department of Electrical Engineering, National Tsing Hua University, 30013 Hsinchu, Taiwan, and is with Phison Electronics Corporation, Hsinchu, Taiwan.

M.-R. Li is with the Department of Electrical Engineering, National Tsing Hua University, 30013 Hsinchu, Taiwan.

Y.-L. Ueng is with the Department of Electrical Engineering and the Institute of Communications Engineering, National Tsing Hua University, 30013 Hsinchu, Taiwan (e-mail: ylueng@ee.nthu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2014.2312478

II. BCH Codes and Decoding Algorithms

In this section, we introduce the basics of finite field arithmetic and the (n, k, t) BCH codes adopted in the IEEE 802.15.6 standard, where n and k denote the codeword length and the information length, respectively, and t denotes the error-correcting capability. The hard- and soft-decision decoding algorithms considered in this paper are also described.

A. Primitive Binary BCH Codes

A Galois field contains a finite number of elements. The Galois field with 2^m elements is denoted by $\mathrm{GF}(2^m)$ and can be used to construct primitive binary BCH codes. The 2^m elements in $\mathrm{GF}(2^m)$ are written as $\{0, \alpha^0, \alpha^1, \cdots, \alpha^{2^m-2}\}$, where α is the primitive element in $\mathrm{GF}(2^m)$. We can construct $\mathrm{GF}(2^m)$ based on the ground field GF(2) through the primitive polynomial $p(x) = 1 + p_1 x + \cdots + p_{m-1}x^{m-1} + x^m$, which includes the property $p(\alpha) = 0$, where $p_i \in \mathrm{GF}(2)$, $i = 1, 2, \cdots, m-1$. All field elements in $\mathrm{GF}(2^m)$ can be represented in both polynomial and vector forms. If $a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1}$ is the polynomial representation of a field element, then the corresponding vector form representation is $(a_0, a_1, \cdots, a_{m-1})$, where $a_i \in \mathrm{GF}(2)$, $i = 0, 1, \cdots, m-1$.

For $\mathrm{GF}(2^m)$, the addition and the multiplication are defined as follows. The addition of two elements in $\mathrm{GF}(2^m)$ with polynomial representation: $a(\alpha) = \sum_{i=0}^{m-1} a_i\alpha^i$ and $b(\alpha) = \sum_{i=0}^{m-1} b_i\alpha^i$ is defined by $a(\alpha) + b(\alpha) = \sum_{i=0}^{m-1}(a_i + b_i)\alpha^i$ where addition $a_i + b_i$ is carried out in the ground field GF(2) and can be implemented using XOR operations. The multiplication of $a(\alpha)$ and $b(\alpha)$ is performed according to $a(\alpha) \cdot b(\alpha) \bmod p(\alpha) \equiv c(\alpha) \equiv \sum_{i=0}^{m-1} c_i\alpha^i$. It can be found that the coefficients c_i of the resultant $c(\alpha)$ can be obtained using only AND and XOR operations.

We can use $\mathrm{GF}(2^m)$ to construct a length-$2^m - 1$ BCH code that can correct t errors with $k \geq n - mt$ information bits, where $n = 2^m - 1$. In the IEEE 802.15.6 standard, the (63, 51, 2) BCH code is constructed using $\mathrm{GF}(2^6)$ and the other BCH code, i.e., the (31, 19, 2) code, is constructed by shortening 32 information bits of the (63, 51, 2) code.

B. Overview of BCH Decoding/Decoders

Conventionally, the Berlekamp-Massey (BM) [1] algorithm, which is an HDD algorithm, is commonly used for BCH codes with higher error-correction capability. Soft-decision decoding/decoders (SDD) for BCH codes have been discussed in [15]. The SDD presented in [15] is applied to the BCH code for the DVB-S2 standard in [5]. Evaluating test patterns using the HDD kernel was proposed by Chase in order to improve the error-rate performance [8]. For the type-II Chase-based SDD, there are a total of 2^t test patterns that should be considered. In order to reduce the complexity of the Chase-based BCH decoder with a moderate-to-large value of t, a one-pass scheme based on the BM algorithm to derive all the error locations was proposed in [16]. The associated implementation architecture was proposed in [17]. In [18], substantial modifications and simplifications were devised using the binary property of the BCH codes. In [18], it was demonstrated that the interpolation-based Chase BCH decoder achieves a better error-rate performance with a lower complexity compared to the schemes presented in [15] and [17] for the (4200, 4096, 8) BCH code. The decoder architecture presented in [18] is well suited for BCH codes with a moderate error-correction capability.

TABLE I
Relationship Between the Number of Errors, Syndrome Conditions, and the Classification of Test Patterns for BCH Codes ($t = 2$)

Number of Errors	Syndrome Condition	Number of Test Patterns with the same Codeword
0	$s_1 = s_3 = 0$	$\binom{p}{0} + \binom{p}{1} + \binom{p}{2}$
> 2	$s_1 = 0, s_3 \neq 0$	invalid
1	$s_1^3 = s_3$	$\binom{p}{0} + \binom{p}{1}$
2	else	1

Since the error-correction capability of the two BCH codes adopted in the IEEE 802.15.6 standard is only 2, i.e., $t = 2$, the Chase algorithm [8] suffices and is adopted in this work. Regarding the HDD kernel for double-error correcting codes, the lookup table (LUT)-based [4] and Peterson Rule-based [2], [3] schemes are more suitable than the BM algorithm. In the LUT-based scheme, the error locations can be determined directly from the syndromes after the syndrome vector is computed. In contrast, the Person Rule-based schemes needs extra steps to locate error locations. It will be demonstrated in Section III-C that the LUT-based scheme has a shorter latency at the cost of increased area overhead compared to the Peterson Rule-based scheme. Since area, and hence energy, is more concerned than latency in the target application, the Peterson Rule-based HDD is adopted as the HDD kernel.

C. Peterson Rule-Based HDD

In HDD, there are three major steps which are described as follows.

1) Computing Syndrome: Considering the 2-error-correcting BCH codes used in the IEEE 802.15.6 standard, the parity check matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha^1 & \alpha^2 & \alpha^3 & \cdots & \alpha^{2^m-2} \\ 1 & (\alpha^3)^1 & (\alpha^3)^2 & (\alpha^3)^3 & \cdots & (\alpha^3)^{2^m-2} \end{bmatrix}. \quad (1)$$

Using the vector-form representation for field elements in (1), the syndrome vector $\mathbf{s}$ of received codeword $\mathbf{r}$ can be calculated according to $\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T = [s_1, s_3]$, where $\mathbf{y}$ is the hard decision of $\mathbf{r}$. The syndrome vector is also used for error-condition check, as listed in Table I, where received codewords with more than two errors are viewed as invalid codewords since they are uncorrectable.

2) Finding the Error Location Polynomial (ELP): A general form of the ELP is $\delta(x) = \sum_{i=0}^{t} \delta_i x^i$. For BCH codes with $t = 2$, the ELP is represented by

$$\delta(x) = 1 + \delta_1 x + \delta_2 x^2. \quad (2)$$

By applying the Peterson rule [19], the coefficients of the polynomial are calculated according to $\delta_1 = s_1$ and $\delta_2 = (s_1^3 + s_3)/s_1$, where s_1 and s_3 are derived from the previous step.

3) Correcting Errors: The error locations can be deduced from the roots of the ELP, in which one root corresponds to one error position. After computing δ_1 and δ_2, the Chien search, which is a trial-and-error method, is usually used to identify the roots of the ELP by substituting all the possible values of α^i, where $i = 0, 1, \cdots, n-1$, into variable x in (2) and then checking whether the equation holds. If $\delta(\alpha^i) = 0$, we can deduce that the $n - i$th bit of the error location vector is 1.

The received codeword can then be corrected according to the error location vector. Since n possible roots of the ELP should be checked, the Chien search dominates the complexity of the HDD.

D. Chase Algorithm-Based SDD

The Chase-II algorithm [8] is a sub-optimum soft-decision algorithm that uses an error-correction-only HDD as the kernel. The procedure for the Chase-II algorithm includes:

1) Determine the positions of the p least reliable bits (LRBs), where $p = \lfloor d_{min}/2 \rfloor$, and d_{min} is the minimum Hamming distance of the code.
2) Generate 2^p test patterns by considering all combinations of the p LRBs.
3) Decode each test pattern by using the HDD kernel. If this test pattern is decoded successfully using the HDD kernel, the decoded word is regarded as a candidate codeword.
4) Evaluate the Euclidean distance for each candidate codeword in the list and select the one with the smallest Euclidean distance as the best decision codeword.

In fact, it is unnecessary for the HDD kernel in the Chase algorithm to process all test patterns since redundant test patterns can be eliminated to save energy. In [9], the authors proposed a two-stage method to reduce the number of redundant test patterns. The test patterns that generate the same codeword after being processed by the HDD kernel are categorized as a single class and they are processed by the HDD kernel only once. In the first stage, the syndromes for all the 2^p test patterns are generated in parallel using 2^p test pattern generation units and 2^p syndrome computation units. The 2^p syndromes are used for the classification performed in the second stage. The syndrome conditions for classification are listed in Table I. Each of the 2^p syndromes is checked to determine whether the number of errors in the specific test pattern is zero. If one of the test patterns is error free, the remaining test patterns do not need to be processed by the HDD kernel. If none of the 2^p test patterns are error-free, the syndrome conditions are sequentially checked according to the sequence listed in Table I. This reduces the number of test patterns to be processed by the HDD kernel.

III. Energy- and Area-Efficient Architecture

Several design techniques are proposed for energy- and area-efficient SDD. An effective early-termination technique is proposed to reduce the number of redundant test patterns and, hence, the energy consumption. An area-efficient *probabilistic* sorting method is used to determine the test patterns that need to be decoded. The hardware complexity of the HDD kernel is reduced by leveraging the Peterson rule and logic sharing in the Chien search unit.

A. Proposed SDD Architecture With Early Termination

Fig. 1 shows the architecture of the proposed Chase-type SDD, including an HDD kernel and peripherals for performing soft-decision decoding. Since the d_{min} value of both the BCH codes adopted in the IEEE 802.15.6 standard is 5, four test patterns, i.e., $p = 2$, should be considered in the Chase-II algorithm. Therefore, the first two LRBs must be determined from the received codeword $\mathbf{r} = (r_0, r_1, \cdots, r_{n-1})$. That is, the first two minimum values denoted as min_1 and min_2 of the reliability set $\{|r_0|, |r_1|, \cdots, |r_{n-1}|\}$, and the corresponding indices denoted as idx_1 and idx_2, respectively, must be determined. If we let $\mathbf{I}$ denote the index set $\{0, 1, 2, \cdots, n-1\}$, then min_1 and min_2 can be written as $min_1 = \min_{i \in \mathbf{I}} |r_i|$, $min_2 = \min_{i \in \mathbf{I} - \{idx_1\}} |r_i|$.

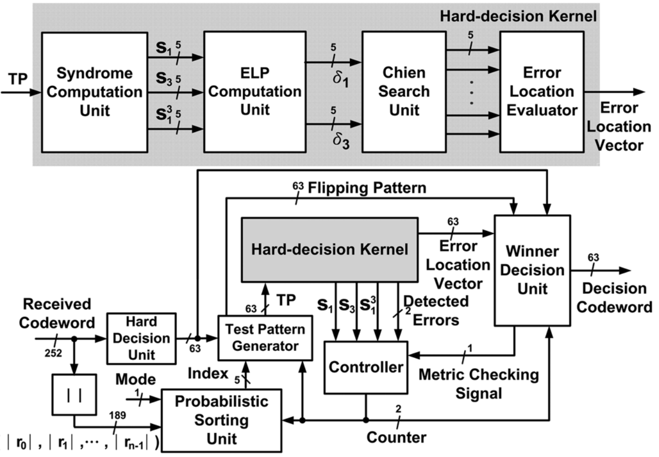


Fig. 1. Architecture of the proposed SDD, where the HDD kernel is highlighted in gray. The remainder of the units are required for performing soft-decision decoding. Four test patterns are processed sequentially.

In order to reduce hardware complexity, these four test patterns are processed in serial by iteratively using the HDD kernel. The hard-decision of the received codeword $\mathbf{r}$ (equivalently $\mathbf{y}$), is regarded as the first test pattern and processed in the first cycle. Through Gray coding [20], the same multiplexer can be shared in the test pattern generator to reduce hardware complexity since only one bit varies in each clock cycle. As a result, the remaining three test patterns are determined as follows. The second test pattern, which is the same as the first test pattern except that the LRB of $\mathbf{y}$, i.e., the idx_1-th bit of $\mathbf{y}$, is flipped. The third test pattern, which is the same as the second test pattern except that the idx_2-th bit is flipped. The fourth test pattern, which is the same as the third test pattern except that the idx_1-th bit is flipped.

In the first cycle, the HDD kernel processes the first test pattern. Simultaneously, the sorting unit determines the p minimum values of the set $\{|r_0|, |r_1|, \cdots, |r_{n-1}|\}$ and the p LRBs of $\mathbf{y}$. The Winner Decision unit calculates the soft-decision metric after the HDD kernel generates the error location vector for the first test pattern. In the next cycle, the HDD kernel processes the second test pattern. The required LRB of $\mathbf{y}$ is determined by the Sorting unit during the previous cycle. The metric of the second test pattern is computed and compared with that of the previous test pattern, then the pattern with the shortest Euclidean distance is stored as the provisional decision codeword. The third and fourth test patterns are processed in a similar manner. After all four test patterns have been processed, the codeword left in the Winner Decision unit is viewed as the decision codeword.

In this work, we propose an early termination (ET) scheme by checking two criteria to eliminate unnecessary test patterns and, hence, reduce energy dissipation. The first criterion dictates that if the number of errors detected in the *current* test pattern is less than t, the remaining test patterns are regarded as redundant patterns and can be discarded. The second criterion dictates that if the *third* test pattern yields the best soft-decision metric among the first three test patterns, the decoding can be terminated. Note that the the second criterion is only checked during the third cycle.

Although the second, instead of the third, test pattern can be used as a decision maker in the second criterion, non-negligible error-rate degradation is observed in the high SNR region. This

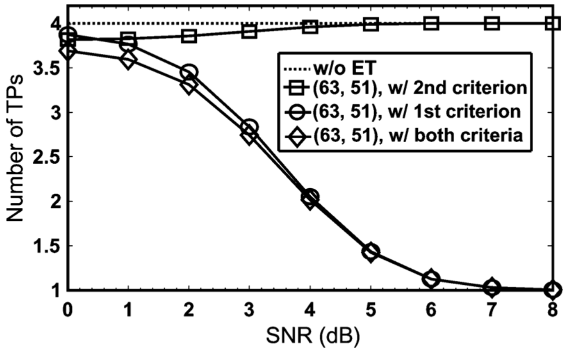


Fig. 2. Average number of test patterns required when applying different ET criteria to the (63, 51) code. The number of test patterns approaches one when $\mathrm{SNR} > 8$ dB.

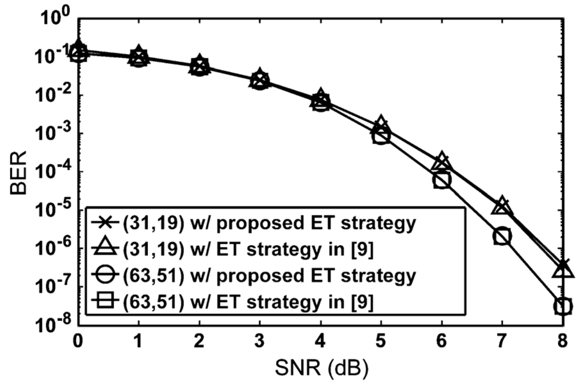


Fig. 3. BER comparison for different early termination techniques. The proposed ET technique results in negligible performance loss.

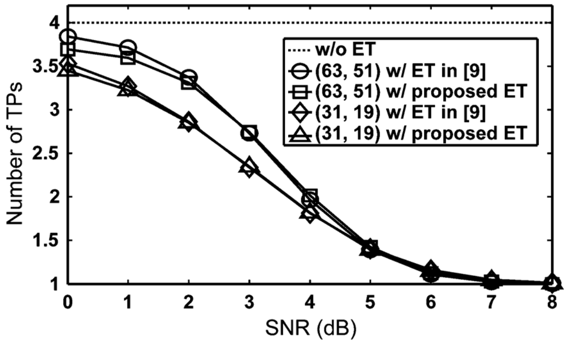


Fig. 4. Average number of test patterns required for different ET strategies. The proposed early termination scheme requires fewer test patterns in the low-SNR regime.

is because the probability of the second test pattern that yields the best soft-decision metric among the first two test patterns is not low enough in the high SNR region of interest. In this case, the decoding process will be falsely terminated and the third and fourth test patterns will not be considered. In contrast, the probability that the third test pattern yields the best soft-decision metric among the first three test patterns is much lower in the high SNR region and, hence, the fourth test pattern is still considered in the remaining Chase decoding.

Fig. 2 shows the simulation results for conditions with different ET criteria. It can be seen that implementing the first criterion can effectively reduce the number of test patterns required in the high SNR region. The number of test patterns approaches 1 when $\mathrm{SNR} > 8$ dB. In the low SNR region, the fourth test pattern is very likely to be redundant if the the third test pattern yields the best soft-decision metric among the first three test patterns. The decoding process can therefore be terminated prior to the fourth test pattern.

Figs. 3 and 4 compare the performance of the proposed ET scheme with the ET scheme proposed in [9] for (63, 51) and (31, 19) codes. Additive white Gaussian noise (AWGN) channel and BPSK are considered. In [9], the authors demonstrated that their ET scheme does not induce any degradation in BER performance compared to the original Chase-II algorithm presented in [8]. Hence, the BER curves shown in Fig. 4 designated (63, 51) and (31, 19) using the ET strategy described in [9] are the same as those when using the original Chase-II algorithm. As shown in Fig. 3, the proposed ET scheme introduces negligible performance degradation compared to the scheme proposed in [9]. Fig. 4 shows the average number of test patterns required for the two BCH codes. To achieve a BER of 10^{-5}, 1.02 and 1.04 test patterns are needed on average for 63- and 31-bit codewords, respectively. Compared with the ET scheme proposed in [9], the proposed method requires fewer test patterns in the low-SNR regime. In addition, the method in [9] requires many test pattern generation and syndrome computation units. Such complex checking circuits are not suitable for efficient hardware implementation.

B. Probabilistic Sorting

In this subsection, we describe sorting architectures to determine min_1, min_2 and the corresponding indices. Determining min_1 via a comparator tree is not complicated through $n-1$ comparisons. Determining the min_2, however, requires substantial hardware resources. Therefore, an approach that optimizes the number of logic gates should be considered to minimize hardware complexity.

In [10], a tree-structure-based algorithm (TS approach) is demonstrated to be more area- and delay-efficient than the sorting-based option for selecting the two minimum values. In the TS approach, a comparator block MVG1 that contains a single q-bit 2-to-1 multiplexer is used as the building block, as shown in Fig. 5. The min_1 and comparison signal (cp, as an index indicator) are generated after the comparison of the input signals. The min_2 is decided in the MVG2 block, which contains an additional multiplexer. The comparison signal cp is 0 when $x_0 < x_1$. Finding min_1 and min_2 from four inputs can be realized using two 2-input MVG2 blocks and a connection unit (CU). The inputs of the CU are labeled as Amin_1, Amin_2, Bmin_1 and Bmin_2. The min_1 can be determined by comparing Amin_1 and Bmin_1, and the min_2 can be determined by either comparing Amin_2 and Bmin_1 or Amin_1 and Bmin_2, depending upon the result of the comparison of Amin_1 or Bmin_1. The idx_1 is (q_1, q_0), where q_0 is determined by using a 1-bit 2-to-1 multiplexer. With these blocks, we can construct a 2^w-input block by recursively using two 2^{w-1}-input blocks and one CU to establish min_1, min_2 and idx_1 in the TS approach.

To reduce the complexity of the hardware, we propose a *probabilistic* sorting scheme that is used to determine the two minimum values from 2^w inputs based on a w-stage comparison tree structure. The second minimum value is generated from the last (bottom) s stages of the tree. Fig. 6(a) shows an example for $s = 2$. The resultant second minimum value may not be the correct min_2 value, but it is the correct min_2 value with a high probability. This value is called the *probabilistic* min_2 value. The probability that the probabilistic min_2 value is the correct min_2 value varies, depending on the s value. For $s = 1$, the last competitor of the first minimum is taken as the probabilistic

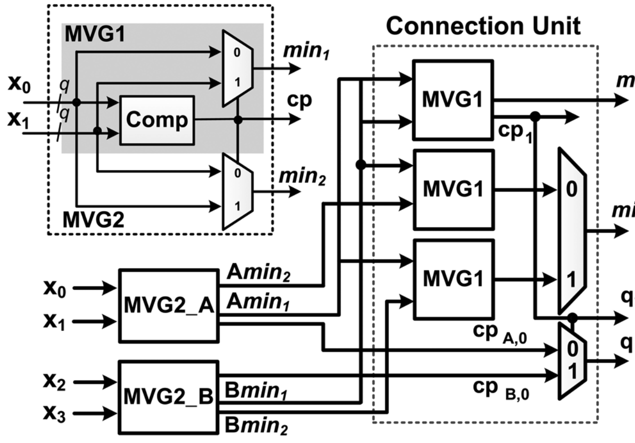


Fig. 5. The 4-input min_1 and min_2 finder used in the TS approach.

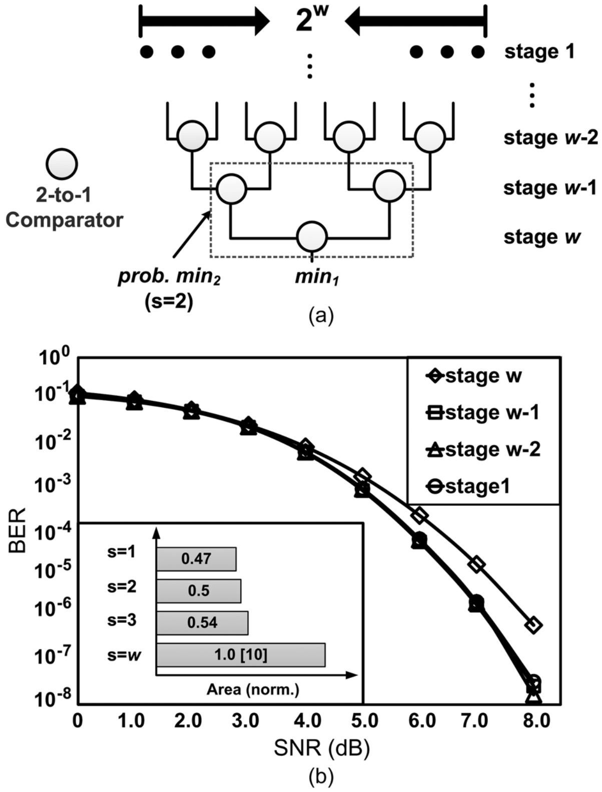


Fig. 6. (a) Conceptual diagram of a *probabilistic* sorting scheme based on a tree structure; (b) Hardware complexity and BER performance when using different sources to yield the *probabilistic* min_2 value.

min_2 value in the comparison tree. If the occurrence of the correct second minimum is uniformly distributed over the inputs of the comparison tree, the probability that the probabilistic min_2 value is the correct min_2 value is $(2^{(w-1)}/(2^w-1)) \approx 50\%$. For $s=2$, the probability that the probabilistic min_2 is the correct min_2 is $((3 \cdot 2^{(w-2)})/(2^w-1)) \approx 75\%$. For $s=w$, the probability that the probabilistic min_2 is the correct min_2 reaches 100%. It is worth noting that the resultant first minimum value is always the correct min_1 value.

Fig. 6(b) shows the hardware complexity and the BER performance when generating the probabilistic min_2 value from different stages. It can be seen that generating the probabilistic min_2 value from the last two stages (equivalently $s=2$) achieves a comparable error-rate performance with smaller area compared to using $s=3$ and $s=w$. The probabilistic min_2 value is therefore generated from the last two stages in our implementation. It is also worth noting that the probabilistic min_2 will be the correct second minimum if $s=w$. The BER curve labeled as $s=w$ is the same as the BER curve using the sorting method presented in [10].

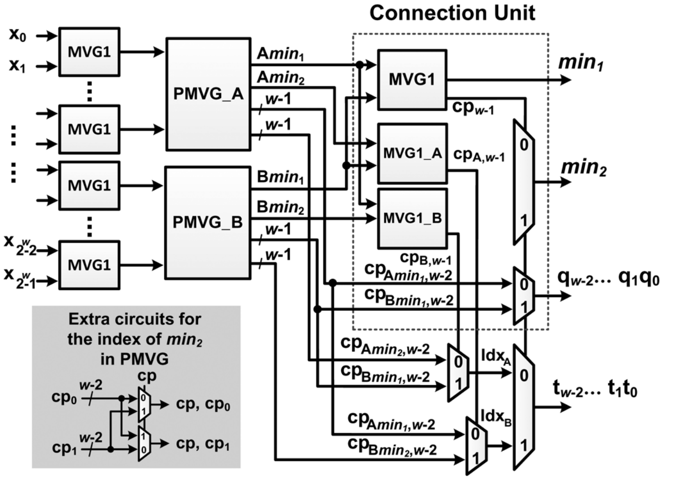


Fig. 7. The *probabilistic* sorting architecture for $s=2$. The MVG2 blocks in [10] are replaced with MVG1 blocks. Only one CU is needed for any value of w.

Fig. 7 shows the *probabilistic* min_2 sorting architecture for $s=2$. Compared to the TS approach, the MVG2 blocks are replaced with MVG1 blocks in the first $w-s$ stages. In contrast, the TS approach requires additional multiplexers in the MVG2 blocks, and the CUs are necessary to form each stage so as to record the correct min_2. By using $s=2$, only one CU is needed no matter how large w is. The method proposed in [10] was originally designed for low-density parity-check (LDPC) [21] decoders. In that case, only min_1, min_2 and idx_1 are required, but idx_2 is not generated using the method proposed in [10]. In the Chase-II algorithm, the idx_2 needs to be recorded. Hence, additional circuits are needed in the PMVG comparison block for recording the idx_2. The idx_2 recorded using $(t_{w-2}, t_{w-1}, \cdots, t_0)$ is decided based on the comparison signals in the final stage.

To evaluate the hardware complexity of the proposed *probabilistic* sorting architecture, as shown in Fig. 7, a serial architecture that generates min_1, min_2, and its associated indices in two clock cycles is used as the reference design. Direct implementation of the serial architecture needs a min_1-only sorting circuit to locate min_1 and its index from the inputs. The same circuit is reused to generate min_2 and its index in the next cycle. This, however, needs additional multiplexers to exclude min_1 from the inputs of the sorting list. The serial architecture introduces a 6% increase in area compared to the proposed sorting architecture.

Fig. 8 shows the area and delay comparison between the TS approach and the proposed *probabilistic* min_2 sorting architecture with $s=2$ for a variety of input lengths. Since the sorting circuit is constructed only using multiplexers and comparators, the area and delay for the multiplexers and comparators are used when evaluating the hardware resources. Taking input $\mathrm{length} = 64$ as an example, 59% of the area is occupied by the multiplexers and the remaining 41% is occupied by the comparators. Considering the critical path, 43% of the delay comes from the multiplexers and the remaining 57% comes from the comparators. With the area and delay reductions shown in Fig. 8, the area

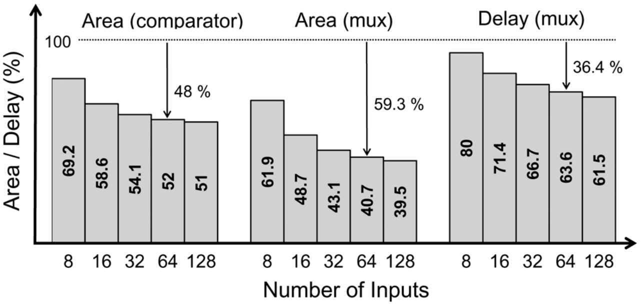


Fig. 8. Area and delay reductions for the proposed *probabilistic* sorting method at various input lengths. The area and delay of the TS approach are normalized to unity. The delay for the comparator used in the *probabilistic* sorting approach remains the same.

TABLE II
COMPARISON BETWEEN THE HDEC, PETERSON RULE-BASED AND LUT-BASED REALIZATIONS FOR THE HDD KERNEL

	[22]	[4]	HDD in this work
Code	HDEC	eBCH	BCH
Technology	45 nm	90 nm	90 nm
Algorithm	HDEC	LUT	Peterson
Data Width	79	64	63
Supply Voltage (V)	NA	1	1
Latency (ns)	7	2.2	3.15
Area (μm^2)	3567.59	37279	20794
Power (μW)	1108.6	NA	3753.9
NATP†	2528.92	1281.45	1039.70
Norm. Energy‡ (pJ/bit)	196.46	NA	187.68

† NATP = Norm. Area × $\frac{\text{Norm. Latency}}{\text{Data Width}}$ scaled to the 90-nm process.

‡ Energy scaled to the 90-nm process.

and delay of the proposed *probabilistic* sorting architecture are reduced by 54.7% and 15.7%, respectively, when compared to the TS approach in [10].

C. Low-Complexity HDD Kernel

The hardware complexity of the HDD kernel can be reduced by leveraging efficient algorithms and architectural transformations. For the (63, 51) and (31, 19) BCH codes, the LUT-based HDD and the Peterson Rule-based HDD are more suitable than the BM algorithm, as described in Section II-B. In the LUT-based scheme, the error positions can be determined directly from the syndromes after the syndrome vector is computed. The LUT-based scheme has a shorter latency at the cost of increased area overhead. When the number of correctable errors increases, the increased complexity of the LUT leads to an inefficient realization in both area and delay. Table II shows a comparison between the LUT-based [4] and the Peterson Rule-based schemes, where the results reported in [2], [3] are not included since the code lengths are not comparable. The area of the HDD kernel with Peterson Rule-based method is only 55.8% of the area required for the LUT-based method. Perterson Rule-based HDD kernel is, hence, adopted in the proposed SDD.

Table II also includes the decoder implementation result for the (79, 64) hierarchical double-error-correcting (HDEC) code, which is modified from the BCH code, proposed in [22]. The error-locator polynomial for the HDEC code is always in quadratic form such that its roots can be derived using the formulas presented in [23]. At the first glance, the HDEC decoder has smaller area and lower power consumption than the proposed HDD. However, HDEC codes are still different from the BCH codes, making the HDEC kernel proposed in [22] infeasible for this work. Moreover, taking technology, data width, and latency into consideration, the proposed HDD still archives a lower normalized area-time product (NATP) and a better normalized energy consumption.

The Chien search unit is used to identify the roots of the ELP in (2) and dominates the complexity of the HDD kernel. For the (63, 51) code, 124 GF multiplications are required for parallel implementation. Generally, AND and XOR operations are needed, however variable x in (2) now becomes known through parallel substitutions. The remaining computations for c_0 to c_5 are XOR operations given the coefficients δ_1 and δ_2. The hardware complexity can be reduced by sharing the XOR operations. A total of 744 bits in 124 GF multiplications need to be computed using XOR operations. All combinations of 2-to-6-bit XOR operations are 114 $(2 \times (\binom{6}{2} + \cdots + \binom{6}{6}))$, so these 744 bits can be shared in 114 XOR combinations. The same XOR term appears among these 114 operations can also be shared. Compared with the direct-mapped implementation that requires 1548 logic gates, a saving of 92.6% is achieved for $\mathrm{GF}(2^6)$.

IV. ENERGY AND AREA MINIMIZATION

A. Minimum-Energy Operation

There exists a minimum energy point due to the crossover between the decreased switching energy and the increased leakage energy when the supply voltage is reduced [13], [14]. The BCH decoder is designed to operate at the optimal operating point in order to minimize energy dissipation.

An inverter chain is used to extract the relationship between the energy and the supply voltage. The total energy dissipation comes from the switching energy and the leakage energy $E_{total} = E_{switching} + E_{leakage}$. The switching energy is modeled as $E_{switching} = \alpha_s N C V_{DD}^2$, where α_s is the switching activity, N is the number of clock cycles, C is the load capacitance at the output node, and V_{DD} is the supply voltage. The switching energy is reduced when the supply voltage is decreased.

A simplified model of the leakage energy is given by

$$E_{leakage} = V_{DD} \cdot I_S \cdot e^{\frac{V_{GS}-V_{th}}{n_s V_T}} \cdot \left(1 - e^{\frac{-V_{DS}}{V_T}}\right) \cdot T, \quad (3)$$

where I_S is a fitting parameter for different processes, V_{GS} is the gate-to-source voltage, V_{DS} is the drain-to-source voltage, V_{th} is the threshold voltage, V_T is the thermal voltage, n_s is related to the slope factor, and T is the computation latency. The increased T, owing to an decreased V_{DD}, in the subthreshold region results in increased leakage energy.

Fig. 9 shows the energy-voltage curve for switching activity α_s, varying from 0 to 0.5. It is observed that the minimum energy point shifts for different switching activities. The minimum energy point occurs at a lower V_{DD} when the weight of the switching energy increases. By applying $\alpha_s = 0.24$, extracted from a commercial power analysis tool, the minimum energy point for the proposed decoder occurs at 0.3 V in 90 nm CMOS.

B. PTL-Based Chien Search

The Chien search unit, implemented based on XOR operations, is the core of the HDD kernel and its hardware complexity needs to be minimized. The number of XOR gates has been reduced through hardware sharing. Instead of standard transmission-gate based XOR realization, the area and energy of the XOR gates can be further reduced by leveraging the pass-transistor logic (PTL) family [11]. Due to the decreased I_{on}/I_{off} ratio in the subthreshold region, the output of the XOR may not

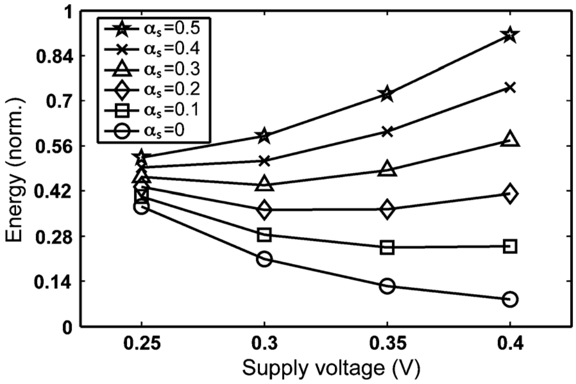


Fig. 9. Minimum energy point analysis for a variety of switching activities in 90 nm CMOS. The minimum energy point of the proposed decoder occurs at 0.3 V.

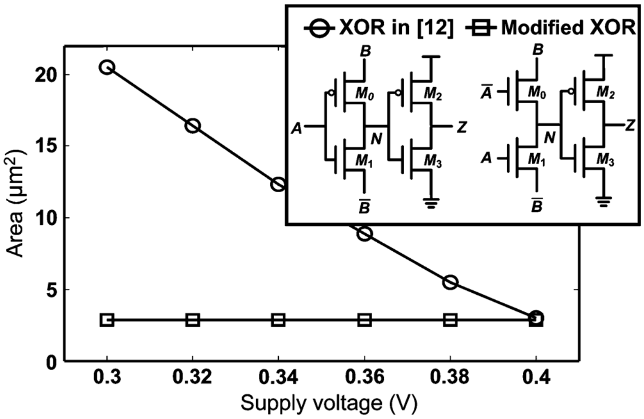


Fig. 10. A comparison of the area required for the XOR ($Z = A \oplus B$) implementation between [12] and the modified XOR structure in order to retain correct functionality in the subthreshold region.

reach full swing. In this work, the circuit is defined as having "correct functionality" if its output voltage can reach the unity-gain points (with $\mathrm{slope} = -1$ on the curve for the inverter Voltage Transfer Characteristic (VTC)). An area-time-efficient XOR implementation [12], as shown in Fig. 10, is used as the reference design. This circuit, however, does not function correctly in the FS corner in the subthreshold region. When a signal level of '0' is applied to the input node A, and '1' is applied to the input node $\bar{B}$, output node N should be '0'. The area of the output inverter is increased to raise the output node N to '1'. In the worst-case corner, node N cannot be lowered enough to be regarded as level '0' due to the weak driving capability of the upper PMOS M_0. This leads to a skewed output inverter with a larger PMOS to retain correct functionality.

To address this issue, M_0 is replaced by an NMOS with an added inverter to generate input $\bar{A}$, as shown in Fig. 10. With a minimum-sized PMOS M_2, the XOR gate functions correctly at the target minimum energy point. In contrast, the circuit proposed in [12] requires a larger M_2 to ensure correct functionality. Fig. 10 shows that the modified XOR gate is more area efficient, even taking the added inverter into consideration, when the supply voltage is lower than 0.4 V in the worst-case corner. Operating at 0.3 V, the skewed inverter in the XOR in [12] introduces an extra 2 ns delay compared to the modified XOR. In the TT corner, the XOR in [12] has a 3 times longer delay than the modified design. Compared to a standard-cell-based implementation, a 22.7% reduction in area and a 33.3% reduction in energy are achieved.

TABLE III
ENERGY AND AREA TRADEOFF FOR THE HDD KERNEL

Level of parallelism	63	32	16	8	4
Norm. Energy	1	1.5	3.82	12	45.54
Norm. Power	1	0.99	0.8	0.7	0.66

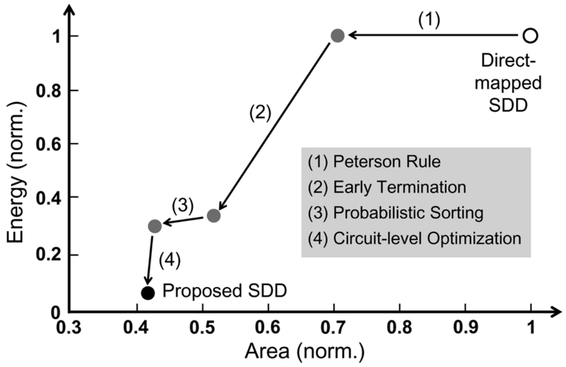


Fig. 11. Summary of the area- and energy-reduction procedure. Compared to the direct-mapped SDD, an overall reduction in area of 58% and an overall reduction in energy of 94% ($\mathrm{SNR} = 5$ dB) are achieved.

C. Energy-Efficient Fully-Parallel Implementation

The direct-mapped serial architecture of the HDD kernel has lower hardware complexity, but the energy dissipation becomes higher owing to the increased operating frequency and supply voltage, given the same throughput requirement. To further minimize the energy dissipation, parallel architecture is applied to the HDD kernel. Table III shows the tradeoff in area and energy by applying the levels of parallelism $(\mathrm{P}) = 63$, 32, 16, 8, and 4. These architectural transformations result in 1, 2, 4, 8, and 16 processing cycles, respectively. The allowable minimum supply voltages are 0.3 V, 0.4 V, 0.46 V, 0.58 V, and 0.8 V, respectively, to achieve the same throughput. The fully-parallel $(\mathrm{P} = 63)$ HDD kernel is the most energy efficient since it operates at the minimum energy point. It also has the lowest energy-area product compared to other feasible architectures.

The area and energy reductions contributed by the techniques discussed above are summarized in Fig. 11 for operation at $\mathrm{SNR} = 5$ dB. The direct-mapped SDD consists of a LUT-based HDD kernel [4], a low-complexity early termination scheme [9], and a tree-structure-based sorting circuit [10]. Starting from a direct-mapped SDD, an overall 58% area reduction and an overall 94% energy reduction are achieved through the proposed techniques.

A LUT-based HDD (with 9618 logic gates) is used as a reference design to demonstrate the improvement in performance of the proposed SDD. The HDD operates at a nominal 1 V supply voltage and only one clock cycle is needed for the hard-decision. Fig. 12 shows the reduction in energy when $\mathrm{SNR} = 0$ to 8 dB based on the synthesis estimate. The energy consumption of the proposed SDD is only 6.2–24.8% that of the LUT-based HDD. Table IV summarizes the comparison between the LUT-based HDD, the direct-mapped SDD, and the proposed SDD. Operating at the nominal voltage (1 V), the proposed SDD achieves a higher throughput with less power/energy consumption and smaller area compared to the direct-mapped (with a LUT-based

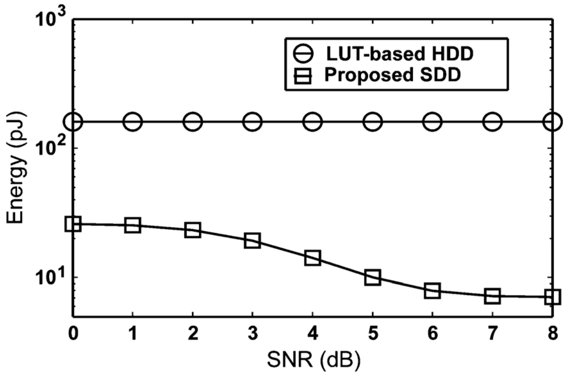


Fig. 12. Reduction in energy of the proposed SDD compared to a LUT-based HDD. The energy dissipation of the proposed SDD is 6.2–24.8% that of the LUT-based HDD for $\mathrm{SNR} = 0$ to 8 dB.

TABLE IV
PERFORMANCE COMPARISON

	LUT-based HDD	Direct-mapped SDD	Proposed SDD	
SDD Algorithm	NA	Chase-II	Chase-II	
Early Termination	NA	[9]	Proposed	
HDD Kernel	LUT	LUT	Peterson	
Technology	90 nm	90 nm	90 nm	
Data Width	63	63	63	
SNR @ BER $= 10^{-6}$	8.0 dB	7.25 dB	7.25 dB	
Voltage (V)	1	1	1	0.29
Max. Clock (MHz)	142.8	30.3	66	0.5
Latency (ns)	7	33	15	2000
Area (μm^2)	35140	82780	34768	
Power (μW)	23000	5455.18	4616	5.4
Thoughtput (Mbps)	7282.8	386.3	842.16	6.38
Thoughtput with ET (Mbps)	NA	1089.53	2365.44	17.92
Energy (pJ) @ SNR = 5dB	161	256.17	98.53	15.37

HDD core) SDD. This clearly shows the advantage of the proposed SDD.

V. Chip Verification

The chip was fabricated in a standard 90 nm CMOS process. The chip micrograph and summary are shown in Fig. 13 and Table V, respectively. This BCH decoder integrates 16 698 transistors in an area of $0.247 \times 0.247\ \mathrm{mm}^2$. In addition to the SDD, additional I/O buffers are added to perform data buffering due to the limited number of pads. The custom Chien search unit is characterized as a macro and is placed at the center to facilitate wire routing. PMOS cross-coupled level shifters are properly inserted across the voltage domains. Fig. 14 shows the Shmoo plot. The throughput is 6.38 and 2.37 Mbps for (63, 51) and (31, 19) codes, respectively, when operating at 500 kHz at the minimum energy point $V_{DD} = 0.29$ V. Although the achieved throughputs are higher than the specifications for WBAN (485.7 and 91.9 kbps, respectively), the decoder can power down after decoding completion through power gating so as to minimize energy [13]. The power dissipation is 5.4 μW at 0.29 V. A 94% energy saving is achieved compared to the estimate of the direct-mapped SDD at $\mathrm{SNR} = 5$ dB, as shown for comparison in Table IV.

Fig. 15 shows the BER performance of the Peterson rule-based HDD and the proposed SDD. In the proposed SDD, the Chase-II algorithm with the proposed early termination and probabilistic sorting methods with $s = 2$ is used. In addition, the HDD kernel used in the proposed SDD is based on the Peterson rule. Coding gains of 0.75 and 1 dB are achieved by SDD for (63, 51) and (31, 19) BCH codes, respectively. Since none of the studies presented in the published literature discuss the soft-decision BCH decoder for WBAN, only the energy and area comparisons with the direct-mapped SDD is included in the previous discussion.

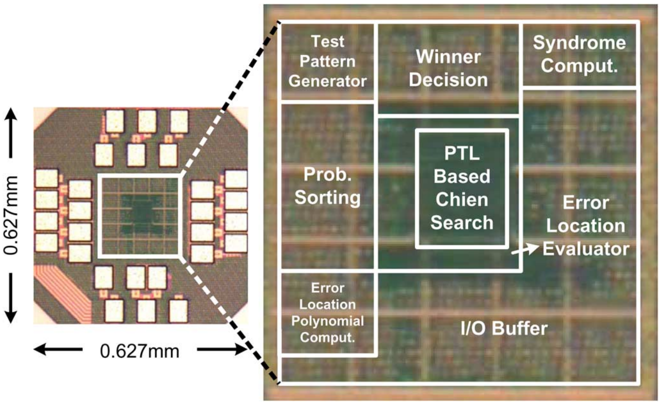


Fig. 13. Chip micrograph for the proposed soft-decision BCH decoder. The custom Chien search unit is placed at the center to facilitate wire routing.

TABLE V
CHIP SUMMARY

Technology	90 nm CMOS	
Die Size (mm^2)	0.627 × 0.627	
Core Size (mm^2)	0.247 × 0.247	
Gate Count	I/O buffer	7,182
	SDD	9,516
Supply Voltage (V)	0.29	
Power Dissipation (μW)	5.4	
Clock Frequency (kHz)	500	
Throughput (Mbps)	(63, 51)	(31, 19)
	6.38	2.37

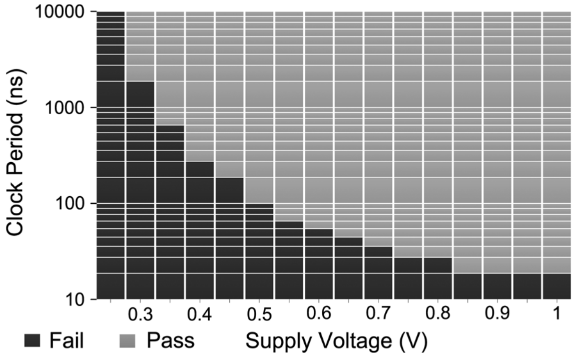


Fig. 14. Shmoo plot for the BCH decoder chip.

VI. Conclusion

This paper presents a low-energy soft-decision (63, 51) and (31, 19) BCH decoder for a WBAN, compatible with the IEEE 802.15.6 standard. Hardware-efficient early termination and area-efficient *probabilistic* sorting are proposed in order to reduce the overhead for performing soft-decision decoding. At circuit level, an area-efficient Chien search unit based on pass-transistor logic for low-voltage operation is proposed. The entire decoder is designed and operated at the minimum energy point of 0.29 V in order to minimize the energy dissipation. Compared to the direct-mapped SDD, a 94% energy saving at $\mathrm{SNR} = 5$ dB is achieved. This chip dissipates only 5.4 μW at

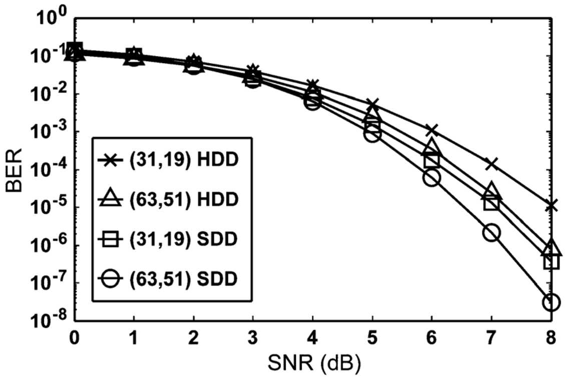


Fig. 15. BER performance for (63, 51) and (31, 19) BCH codes. The soft-decision decoding provides coding gains of 0.75 and 1 dB, respectively.

500 kHz from a 0.29 V supply voltage. For energy-constrained biomedical applications, this work provides a promising channel decoder solution for reliable data transmission in WBAN environments.

Acknowledgment

The authors thank Prof. H.-C. Chang for his valuable comments regarding architecture design.

References

[1] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.

[2] C. C. Chu, Y. M. Lin, C. H. Yang, and H. C. Chang, "A fully parallel BCH codec with double error correcting capability for NOR flash applications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2012)*, Mar. 2012, pp. 1605–1608.

[3] W. Xueqiang, P. Liyang, W. Dong, H. Chaohong, and Z. Runde, "A high-speed two-cell BCH decoder for error correcting in MLC NOR flash memories," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 11, pp. 865–869, Nov. 2009.

[4] R. Naseer and J. Draper, "Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in *Proc. Eur. Solid-State Circuits Conf. (ESSCIRC 2008)*, Sep. 2008, pp. 222–225.

[5] Y. M. Lin, C. L. Chen, H. C. Chang, and C. Y. Lee, "A 26.9 K 314.5 Mb/s soft (32400, 32208) BCH decoder chip for DVB-S2 system," *IEEE J. Solid-State Circuits*, vol. 45, no. 11, pp. 2330–2340, Nov. 2010.

[6] IEEE WPAN Task Group 6 [Online]. Available: http://www.ieee802.org/15/pub/TG6.html

[7] W. Zhang, X. Zhang, and H. Wang, "Increasing the energy efficiency of WSNs using algebraic soft-decision Reed-Solomon decoders," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS 2012)*, Dec. 2012, pp. 49–52.

[8] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 170–182, Jan. 1972.

[9] G. T. Chen, L. Cao, L. Yu, and C. W. Chen, "Test-pattern-reduced decoding for turbo product codes with multi-error-correcting eBCH codes," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 307–310, 2009.

[10] C. L. Wey, M. D. Shieh, and S. Y. Lin, "Algorithms of finding the first two minimum values and their hardware implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 11, pp. 3430–3437, Dec. 2008.

[11] J. Rabaey, *Low Power Design Essentials*. New York: Springer, 2009.

[12] J. C. Lo, "A novel area-time efficient static CMOS totally self-checking comparator," *IEEE J. Solid-State Circuits*, vol. 28, no. 2, pp. 165–168, Feb. 1993.

[13] A. Wang and A. Chandrakasan, "180-mV subthreshold FFT processor using a minimum energy design methodology," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Sep. 2005.

[14] B. H. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1778–1786, Sep. 2005.

[15] Y. M. Lin, H. C. Chang, and C. Y. Lee, "An improved soft BCH decoder with one extra error compensation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS 2010)*, May 2010, pp. 3941–3944.

[16] Y. Wu, "Fast Chase decoding algorithms and architectures for Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 109–129, Jan. 2012.

[17] X. Zhang, J. Zhu, and Y. Wu, "Efficient one-pass Chase soft-decision BCH decoder for multi-level cell NAND flash memory," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, Aug. 2011, pp. 1–4.

[18] X. Zhang, "An efficient interpolation-based Chase BCH decoder," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 4, pp. 212–216, Apr. 2013.

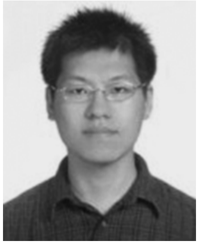
[19] W. W. Peterson and E. J. Weldon, Jr., *Error-Correction Codes*, 2nd ed. Cambridge, MA, USA: MIT Press, 1972.

[20] S. A. Hirst, B. Honary, and G. Markarian, "Fast Chase algorithm with an application in turbo decoding," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1693–1699, Oct. 2001.

[21] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. IT-8, no. 1, pp. 21–28, Jan. 1962.

[22] Z. Wang, "Hierarchical decoding of double error correcting codes for high speed reliable memories," in *Proc. 50th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, 2013, pp. 1–7.

[23] C.-L. Chen, "Formulas for the solutions of quadratic equations over $\mathrm{GF}(2^m)$," *IEEE Trans. Inf. Theory*, vol. 28, no. 5, pp. 792–794, Sep. 1982.



Chia-Hsiang Yang (S'07–M'10) received his B.S. and M.S. degrees from the National Taiwan University, Taiwan, in 2002 and 2004, respectively, all in electrical engineering. He received his Ph.D. degree from the Department of Electrical Engineering of the University of California, Los Angeles, CA, USA, in 2010. He then joined the faculty of the Electronics Engineering Department at the National Chiao Tung University, Taiwan, as an Assistant Professor. His research interests include energy-efficient integrated circuits and architectures for biomedical and communication signal processing.

Ting-Ying Huang received his B.S. degree in electrical engineering from National Chung Hsing University, Taiwan, in 2008 and M.S. degree in electrical engineering from National Tsing Hua University, Taiwan, in 2013. He is a Research and Development (R&D) Engineer in Phison Electronics Corporation, Taiwan, now. His research interests include channel coding, VLSI designs for channel coding, and mix-signal VLSI design.

Mao-Ruei Li received his B.S. and M.S. degrees in electrical engineering from National Tsing Hua University, Taiwan, in 2011 and 2013, respectively. He is currently a doctoral student in the Department of Electrical Engineering, National Tsing Hua University. His research interests include channel coding, VLSI designs for channel coding, and mix-signal VLSI design.

Yeong-Luh Ueng (M'05) received the Ph.D. degree in communication engineering from the National Taiwan University, Taipei, Taiwan, in 2001. From 2001 to 2005, he was with a private communication technology company, where he focused on the design and development of various wireless chips. Since December 2005, he has been a member of the faculty of the National Tsing Hua University, Hsinchu, Taiwan, where he is currently an Associate Professor with the Department of Electrical Engineering and the Institute of Communications Engineering. His research interests include coding theory, wireless communications, and communication ICs.