IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 25, 2017, accepted August 14, 2017, date of publication August 21, 2017, date of current version September 19, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2742531

Design, Simulation, and Implementation of a CMOS Analog Decoder for (480,240) Low-Density Parity-Check Code

ZHE ZHAO, KAI YANG, (Member, IEEE), HAO ZHENG, FEI GAO, AND XIANGYUAN BU, (Member, IEEE)

School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Kai Yang (yangkai@ieee.org)

This work was supported by the National Natural Science Foundation of China under Grant 61501028 and Grant 61771054.

ABSTRACT The analog low-density parity-check (LDPC) decoder, which is a specific application of the probabilistic computing, is considered to be a promising solution for power-constrained applications. However, due to the lack of efficient electronic design automation tools and reliable circuit model, the analog LDPC decoders suffer from costly hand-craft design cycle, and are unable to provide enough coding gains for practical applications. In this paper, we present an implementation of a (480,240) CMOS analog LDPC decoder, which is the longest implemented code to date using the analog approach. We first propose an analog LDPC decoder architecture, which is constructed by the reusable modules and can significantly reduce the hardware complexity. And then, we present a mixed behavioral and structural model for the analog LDPC decoding circuits, which can reliably and efficiently predict the error-correcting performance. Finally, the experimental results show that the decoder prototype, which is fabricated in a 0.35-μm CMOS technology, can achieve a throughput higher than 50 Mbps with the power consumption of 86.3 mW for the decoder core, and can offer a superior 6.3-dB coding gain at the bit error rate of 10^{-6} when the tested throughput is 5 Mbps. The proposed analog LDPC decoder is suitable for the power-limited applications with moderate throughput and certain coding gains.

INDEX TERMS Analog LDPC decoder, architecture-aware LDPC code, reusable modules, circuit-level model, mismatch effects, probabilistic computing.

I. INTRODUCTION

Low-density parity-check (LDPC) codes [1], [2], which belong to an important class of capacity-approaching error-correcting codes, have been adopted by many commercialized communication standards, such as Wi-Fi, WiMAX, DVB-S2, and CCSDS. The key driver behind the success of LDPC is the use of iterative message-passing algorithms [3], which are powerful decoding algorithms with a manageable complexity. The iterative decoding algorithms are scalable, and hence can be efficiently implemented using digital [4], [5] and analog [6], [7] approaches. The motivation to use analog circuits for decoding is based on lower power dissipation and faster processing speed than the digital design [8], [9]. Therefore, the analog approach has the potential to meet the requirement of energy efficiency in the current and emerging application scenarios. For example, the space telecommand applications require low power consumption with high communication reliability [10], and the ultra-portable devices need long battery-driven life-time with small physical dimension [11]. Over the past decade, a number of low power analog LDPC decoding chips have been designed and fabricated [10]–[14], where the (120,75) Min-Sum (MS) based analog LDPC decoder [12] is the longest implemented code using analog techniques. It is noticed that the above mentioned analogue circuits were merely limited to proof-of-concept decoders with very short block lengths, which are unable to provide enough coding gains for practical applications [15].

There exist two factors that limit the applicability of analog decoders. On the one hand, costly hand-craft method plays a large part in the design flow of the analog LDPC decoders due to the lack of efficient electronic design automation tools [16]. The computation of the analog decoders is performed in a full-parallel fashion, and hence the complexity in

the traditional design grows proportionally with respect to the code length. To design an analog LDPC decoder for practical applications, the hand-craft process is tedious, which in turn prolongs the design cycle and raises the error rate. On the other hand, it is very difficult to obtain fast and accurate estimation of the decoding circuit performance for large codes due to the high-complexity of the analogue circuit [15]. Developing a model of the analog decoder to evaluate the bit error rates (BER) is necessary before performing the physical implementation. However, the transistor-level simulations are too time-consuming to obtain the BER for complex decoders. The circuit behavioral models do not consider the non-ideal effects, such as the device matching and the time delay [14], [17]. When considering complex decoders, there is a relatively large gap between the simulation results from the behavioral model and the actual measurements from the fabricated circuits [15], [18].

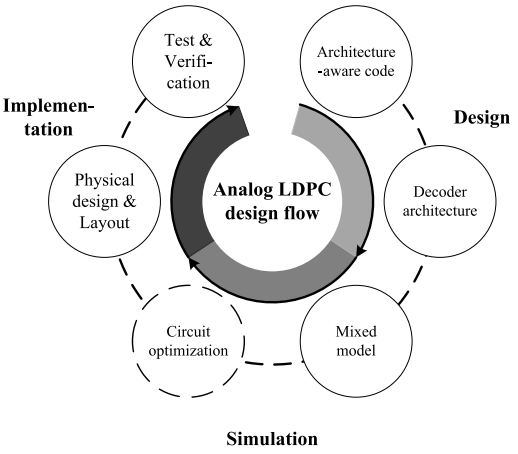


FIGURE 1. Proposed analog LDPC decoder design flow.

Motivated by this, we attempt to overcome the above limitations by performing operations at three levels of abstraction in this paper, namely, the architecture design level, the circuit simulation level, and the chip implementation level, as shown in Fig. 1. Firstly, a low-complexity decoder architecture with reusable modules is proposed to solve the hand-craft design complexity problem. Secondly, a mixed behavioral and structural model, which relates the device size to the error-correcting performance, is presented to permit the pre-layout simulation in an acceptable time. Finally, a (480,240) CMOS analog LDPC decoder, which is the longest implemented code to date using the analog approach, is realized in a 0.35-μm standard CMOS process. The experimental results show that the prototype chip can achieve a throughput higher than 50Mbps with a power consumption of 86.3mW for decoder core, which corresponds to the energy per decoded bit of 1.726nJ. When the test throughput is 5Mbps, the coding gain at BER = 10^{-4} is about 5.6dB, which is the highest among all reported analog decoders, and the coding gain at BER = 10^{-6} is up to 6.3dB. The experiments also indicate that the probability stopping criterion employed in the proposed analog decoder can reduce the decoding delay by at most 93%, compared with the experiment-based decoding delay adopted in the existing works.

The rest of this paper is organized as follows. The decoding algorithms and basic soft-logic gates for analog LDPC decoders are introduced in Section II. Details on the decoder architecture design are described in Section III. Section IV presents the mixed model for analog decoding circuits and analyzes the simulation results. In Section V, the hardware architecture of the fabricated chip is presented. In Section VI, the experimental results and comparisons with existing analog LDPC decoders are discussed. Conclusions can be found in Section VII.

II. ANALOG LDPC DECODER BASED ON SUM-PRODUCT ALGORITHMS

LDPC is defined by the check matrix H, whose structure can be graphically represented by a factor graph [3]. For an (N, K) LDPC code, the factor graph consists of variable nodes (VNs) v_n, $n = 1, \cdots, N$, which connect with check nodes (CNs) c_k, $k = 1, \cdots, N-K$ using edges. The degree of a node is equal to the number of nodes that connect to it. The dimension and structure of the check matrix H not only define the error-correcting performance, but also determine the complexity of the decoding process.

The iterative message-passing algorithm [19], which offers near-optimum decoding performance at a manageable complexity, is the most widely used method for LDPC codes. In practice, the standard decoding methods are either based on the sum-product (SP) algorithm [3] or its approximation, commonly referred to as the MS algorithm [20]. The SP algorithm outperforms the MS algorithm in terms of error correction performance at the cost of computational complexity. Both the SP and MS decoding algorithms can be performed by analog circuits.

In the iterative decoding process, the basic computations of the SP algorithm can be expressed as [21]

$$p_Z(z) = \gamma \sum_{x \in X} \sum_{y \in Y} p_X(x)\, p_Y(y) f(x, y, z), \quad \forall z \in Z, \quad (1)$$

where p_X and p_Y are the input probability density functions (PDFs) defined on the finite sets X and Y respectively, p_Z is the output PDF defined on a finite set Z, $f(x, y, z)$ is an indicator function from $X \times Y \times Z$ into $\{0, 1\}$, and γ is a scale factor to guarantee that $\sum_m p_Z(z_m) = 1$, $\forall z_m \in Z$. Based on the factor graph of LDPC codes, the updating message, which is the PDF of each received bit, can be transmitted between VNs and CNs in an iterative manner. The function $f(x, y, z)$ for the updating rule of variable-to-check message is equal to 1 if and only if $x = y = z$ and 0 otherwise. As a result, the updating rule can be given by

$$\begin{bmatrix} p_Z(0) \\ p_Z(1) \end{bmatrix} = \gamma \begin{bmatrix} p_X(0)\, p_Y(0) \\ p_X(1)\, p_Y(1) \end{bmatrix}. \quad (2)$$

Eq. (2) is the formulation of the basic soft-equal gate. Similarly, the function $f(x, y, z)$ for the updating rule of

check-to-variable message is equal to 1 if and only if $z = x \oplus y$ and 0 otherwise, where $\oplus$ denotes the binary addition. The corresponding updating rule can be given by

$$\begin{bmatrix} p_Z(0) \\ p_Z(1) \end{bmatrix} = \begin{bmatrix} p_X(0)p_Y(0) + p_X(1)p_Y(1) \\ p_X(0)p_Y(1) + p_X(1)p_Y(0) \end{bmatrix}. \quad (3)$$

Eq. (3) is the formulation of the basic soft-XOR gate. From (2) and (3), it is shown that the basic soft-logic gate takes two input PDFs and calculates one output PDF. The updating computation in degree-3 nodes can be realized by three basic gates, and the nodes with a degree of more than three can be implemented by the degree-3 nodes in a cascade structure [21]. In the following, we will use the two basic gates to design and construct an analog LDPC decoder.

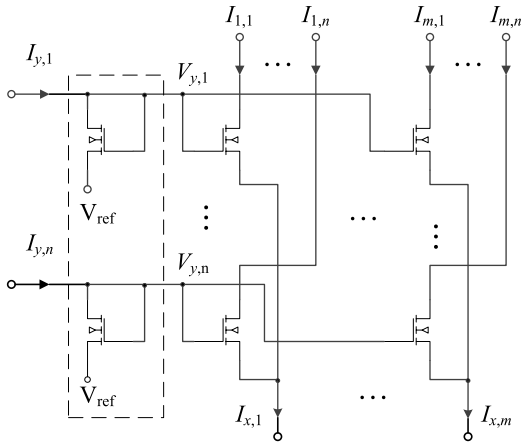


FIGURE 2. Fundamental circuit: pairwise multiplication.

The implementation of the basic soft-logic gate could be viewed as a generalization of the well-known Gilbert multiplier. In this approach, the currents represent the probability values, and are normalized to the "unity" current I_U. There are three operations in the basic gate: multiplication, summation, and normalization. The function of the pairwise multiplication circuit, as shown in Fig. 2, is then given by

$$I_{i,j} = I_Z \cdot \frac{I_{x,i}}{I_X} \cdot \frac{I_{y,j}}{I_Y}, \quad (4)$$

where $I_X = \sum_{i=1}^{m} I_{x,i}$, $I_Y = \sum_{j=1}^{n} I_{y,j}$, $I_Z = \sum_{i=1}^{m}\sum_{j=1}^{n} I_{i,j} = I_X$. According to the Kirchhoff current law, the summation computation is easily accomplished by connecting wires together. The normalization circuit is actually a degenerate version of the Gilbert multiplier with $m = 1$ and $I_{x,1}$ fixed to I_U, and its function is given by

$$I_{z,k} = I_U \cdot \frac{I'_{z,k}}{\sum_{l=1}^{n} I'_{z,l}}, \quad (5)$$

where $I'_{z,k}$ is the output current from the summation circuit.

III. LOW-COMPLEXITY DECODER ARCHITECTURE DESIGN

An important issue to be addressed in fabricating an analog LDPC decoder for practical application is the costly handcraft design caused by the lack of the efficient automation tools. To mitigate this problem, here we propose a design approach, whose core idea is to take the full advantage of the structured nature in the architecture-aware LDPC codes, to construct an efficient LDPC decoder architecture in this section.

A. ARCHITECTURE-AWARE LDPC CODES

Unstructured randomly-designed LDPC codes potentially achieve the best error correction performance at the cost of complex routing [22]. Therefore, all the standardized LDPC codes adopt architecture-aware design [23], and have shown comparable error performance as randomly structured codes. As such, we design an architecture-aware LDPC code with easy description.

The parity-check matrix of the proposed (480,240) LDPC code incorporates a common set of features favorable for efficient decoder implementation, and is expressed as

$$H = \left[H_d \middle| H_p \right] = \begin{array}{c} \\ C_A \\ C_B \\ C_C \\ C_D \end{array} \begin{array}{c} \begin{array}{cccc} V_A & V_B & V_C & V_D \end{array} \\ \left[\begin{array}{cccc} H_A & H_B & H_C & H_D \\ H_B & H_C & H_D & H_A \\ H_C & H_D & H_A & H_B \\ H_D & H_A & H_B & H_C \end{array}\right. \end{array} \begin{array}{c} V_p \\ \left.\begin{array}{c} 1\ 1\ \cdots\ 0 \\ 0\ 1\ \ddots\ 0 \\ \vdots\quad \ddots\ 1 \\ 0\ 0\ \cdots\ 1 \end{array}\right] \end{array}, \quad (6)$$

where the *parity matrix* H_p is a 240×240 dual-diagonal matrix, and the *information matrix* H_d is composed of four 60×60 sub-matrices H_A, H_B, H_C and H_D. The constructions of H_A, H_B, H_C and H_D are based on the rotated versions of the permutation matrices π_A, π_B, π_C, and π_D respectively, where π_A is defined by triplet (15, 6, 4) [24], and π_B, π_C, and π_D are obtained by rotating π_A counterclockwise, respectively, as shown in Fig. 3(a).

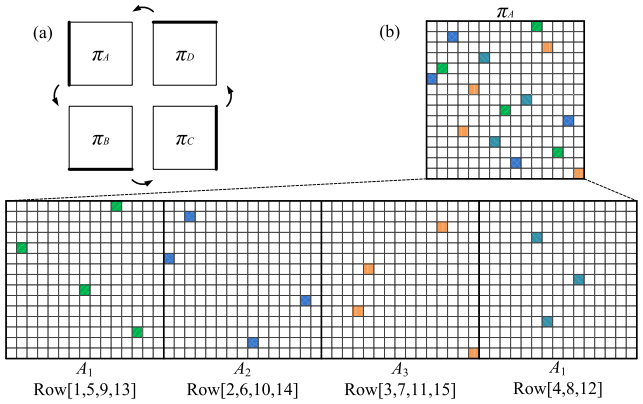


FIGURE 3. Proposed architecture-aware LDPC codes: (a) π-rotation construction, and (b) four component vector $[A_1, A_2, A_3, A_4]$ extracted from permuted matrix π_A.

The permutation matrix π_A is extended to a four component vector $[A_1, A_2, A_3, A_4]$ by extracting the row vector, as shown in Fig. 3(b). Using the cyclic shifted vector $[A_1, A_2, A_3, A_4]^T$,

the sub-matrix H_A is expressed as

$$H_A = \begin{bmatrix} A_1 & A_4 & A_3 & A_2 \\ A_2 & A_1 & A_4 & A_3 \\ A_3 & A_2 & A_1 & A_4 \\ A_4 & A_3 & A_2 & A_1 \end{bmatrix}. \quad (7)$$

Similarly, we can derive the sub-matrices H_B, H_C, and H_D.

According to the grouping structure of the parity-check matrix H, there are three parts of vertex set as shown in (6), namely, the check node part containing four subsets (C_A, C_B, C_C, and C_D), the variable node part for the information bits containing four subsets (V_A, V_B, V_C, and V_D), and the variable node part for the parity bits containing one subset V_p.

B. REUSABLE MODULES

From (6), we can observe that the parity-check matrix H, which is constructed by the circular row vectors and the dual-diagonal pattern, is particularly convenient for building the reusable modules.

By performing the row transformation of the parity matrix H and rearranging the order of the check nodes within each subset, the sub-matrices H_A, H_B, H_C, and H_D are converted to a similar structure. For example, the sub-matrix H_A is converted to

$$H'_A = \begin{bmatrix} \pi_A & 0_{15\times15} & 0_{15\times15} & 0_{15\times15} \\ 0_{15\times15} & \pi_A & 0_{15\times15} & 0_{15\times15} \\ 0_{15\times15} & 0_{15\times15} & \pi_A & 0_{15\times15} \\ 0_{15\times15} & 0_{15\times15} & 0_{15\times15} & \pi_A \end{bmatrix}. \quad (8)$$

After transforming, the parity-check matrix of the (480,240) LDPC code becomes

$$H' = \begin{array}{c} \\ C'_A \\ C'_B \\ C'_C \\ C'_D \end{array} \left[\begin{array}{cccc|c} V_A & V_B & V_C & V_D & V'_p \\ \hline H'_A & H'_B & H'_C & H'_D & \\ H'_B & H'_C & H'_D & H'_A & H'_p \\ H'_C & H'_D & H'_A & H'_B & \\ H'_D & H'_A & H'_B & H'_C & \end{array}\right], \quad (9)$$

where H'_p is also a 240 × 240 square matrix with a dual-diagonal pattern, but its corresponding variable node part V'_p has been reordered.

By fully exploiting the structural nature of the parity-check matrix H' in (9), the modular matrix with similar structure can be found and expressed as

$$H_m = \begin{array}{c} \\ C_\alpha \\ C_\beta \end{array} \begin{array}{c} \begin{array}{ccc} V_\alpha & V_\beta & V_\theta \end{array} \\ \begin{bmatrix} \pi_A & \pi_B & P_{15} \\ \pi_C & \pi_D & 0_{15\times15} \end{bmatrix} \end{array}, \quad (10)$$

where P_{15} is isolated from H'_p and is a 15 × 15 dual-diagonal matrix. Based on the cascade structure, each check node with a degree 6 in (6) can be constructed by connecting a degree-5 node and a degree-3 node, and each variable node with a degree 5 for the information bits in (6) can be constructed by connecting a degree-4 node and a degree-3 node. As such, there are also three parts of vertex set in (10), namely, the check node part containing a degree-5 check node subset C_α and a degree-3 check node subset C_β, the variable node part for the information bits containing a degree-4 variable node subset V_α and a degree-3 variable node subset V_β, and the variable node part for the parity bits containing one subset V_θ. The parity-check matrix H' can then be constructed by using 16 modular matrices H_m.

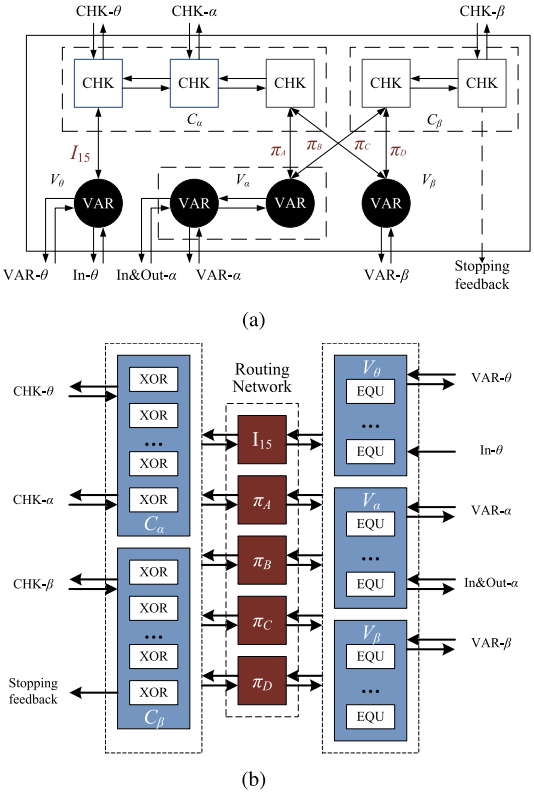


FIGURE 4. Reusable module: (a) isomorphic sub-graph according to the modular matrix and (b) the related building block.

Based on (10), the corresponding isomorphic sub-graph is shown in Fig. 4(a), and it is constructed from a (3,2) proto-graph by making 15 copies of each variable node and each check node. Due to the fact that the message representation is described by a probability vector with two elements, i.e., $p(0)$ and $p(1)$, each edge in the isomorphic sub-graph represents 2 × 15 lines. According to the probability stopping criterion [25], an edge named "Stopping feedback" is added to the node C_β,[1] whose degree becomes 4. The 5-degree check node C_α is realized by three 3-degree nodes in serial connection, and the 4-degree nodes V_α and C_β are implemented with two 3-degree variable nodes and two 3-degree check nodes in series, respectively. The connection "In&Out-α" represents the input probability of transmitting bits and the output probability of decoding bits for node V_α, and the connection "In-θ" represents the input probability of transmitting bits for node V_θ. The external interfaces "CHK-α", "CHK-β" and "CHK-θ" of the check nodes are connected to the corresponding nodes in other modular

[1]For convenience, the node subset C_β is simplified to the node C_β in Fig. 4(a), and other node subsets in the following are similar.

matrices, and so do the external interfaces of the variable nodes. The internal routing within the isomorphic sub-graph is connected according to the four permutation matrices (π_A, π_B, π_C, and π_D) and the identity matrix I_{15}.

The implementation of the reusable module is shown in Fig. 4(b), which is a direct mapping of the isomorphic sub-graph to two types of hardware components: the soft-logic gates to compute the update messages, and the routing network to represent the edges of the graph. The reusable module is built by 225 basic soft-XOR gates, which is equal to the sum of the basic gates used by 15 degree-5 node C_α's and 15 degree-4 node C_β's, and 180 basic soft-equal gates, which is equal to the sum of the basic gates used by 15 degree-4 node V_α's, 15 degree-3 node V_β's, and 15 degree-3 node V_θ's. The connection "In&Out-α" consists of 2×15 inputs and 2×15 outputs for node V_α, and the connection "In-θ" consists of 2×15 inputs for node V_θ. The connection "Stopping feedback", which composes of 2×15 output signals, represents the corresponding probabilities that satisfy the parity-check equations. Each one of the other bi-directional connections out of the module includes $2 \times 2 \times 15$ wires. It is noticed that the routing network within the module is composed of five types of router predetermined respectively by the parity matrices π_A, π_B, π_C, π_D, and I_{15}. The hand-craft designing in the reusable module can be reduced as the connections defined by five routers, and this operation is performed only once in the whole analog design flow.

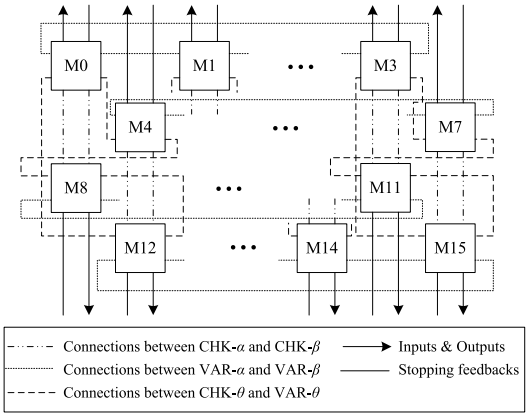


FIGURE 5. Block diagram of (480,240) analog LDPC decoding network.

C. COMPLEXITY ANALYSIS

The (480,240) analog LDPC decoding network is constructed by 16 reusable modules, as shown in Fig. 5. The design complexity of the analog decoding circuit is measured by the number of the placed wires here. There are 1439 bi-directional edges in the factor graph of (480,240) LDPC code, and therefore the mapped $1439 \times 4 = 5756$ wires are implemented in the original architecture without reusable modules. In the proposed decoding architecture, the placed 3180 wires are divided into two parts: the local connection including 300[2] wires within the reusable module, and the global connection including 2880[3] wires between different reusable modules. After adopting the reusable module to construct the decoding network, the number of wires is reduced by 44.8%. Furthermore, the routers within the reusable module are used to encapsulate irregular local wiring, and hence the global wires between different reusable modules are regular and structured. The combination of the scalable decoding architecture and the routing strategy significantly reduces the wiring overhead and minimizes the routing congestion. Hence, the hand-craft design complexity is greatly reduced to an acceptable level by constructing the analog LDPC decoding circuit with reusable modules.

IV. MIXED BEHAVIORAL AND STRUCTURAL MODEL FOR ANALOG DECODERS

Due to the large size of the analog circuit, the classic transistor-level simulation is too time-consuming to predict the system-level specifications for the proposed analog decoder. In order to facilitate the pre-layout simulation, a mixed behavioral and structural model for the analog LDPC decoders, which relates the transistor-level parameters to the system specifications, is presented here.

A. TRANSISTOR-LEVEL DESIGN IN THE SOFT-LOGIC GATE

Before modeling the analog decoding system, a design rule for the transistor size in the soft-logic gate is introduced here. For the gate-level design in the analog LDPC decoder, the Gilbert multiplier circuit, where the MOS transistors should operate in the weak inversion, is key consideration. The MOS transistors operating in the weak inversion should satisfy the following condition [26]

$$I_D \le I_U \le \frac{1}{10} \cdot \frac{2\mu C_{ox} U_T^2}{\kappa} \cdot \exp\left(\frac{\kappa V_{T0}}{U_T}\right) \cdot \frac{W}{L}, \quad (11)$$

where I_D is the drain current, I_U is the reference unit current, W/L is the transistor size, μ is the mobility of charge carriers in the device, C_{ox} is the gate capacitance per unit area, $U_T \approx 0.0258$ V is the well-known thermal voltage, and κ and V_{T0} are the constants of the fabrication process. It is noticed that the design parameter I_U not only keeps the transistors in weak inversion region, but also controls the total power consumption of the decoding core.

From a qualitative point of view, higher decoding accuracy together with better matching properties requires larger transistor size W/L and lower normalizing current I_U at the cost of slowing the circuit response and increasing the node processing delays. Hence, the choice of the device size in the basic gate is a trade-off between the mismatch considerations and the dynamic requirements.

[2]There are five routers within the reusable module, and each router includes 4×15 wires.

[3]There are 3×16 lines between different reusable modules, and each line includes 4×15 wires.

B. MIXED STRUCTURAL AND BEHAVIORAL MODEL

Circuit-level simulation of the proposed decoder is a very resource intensive task. As such, we propose a modeling approach, which is based on the structural description of the decoding network and the behavioral model of the basic gates, to fully characterize the mismatch effects and the dynamic features of the transistors. And the modeling approach of the analog LDPC decoding network is schematized in Fig. 6, where the "Soft XOR Gates" and "Soft EQU Gates" modules denote a column of soft-XOR gates and a column of soft-equal gates, respectively.

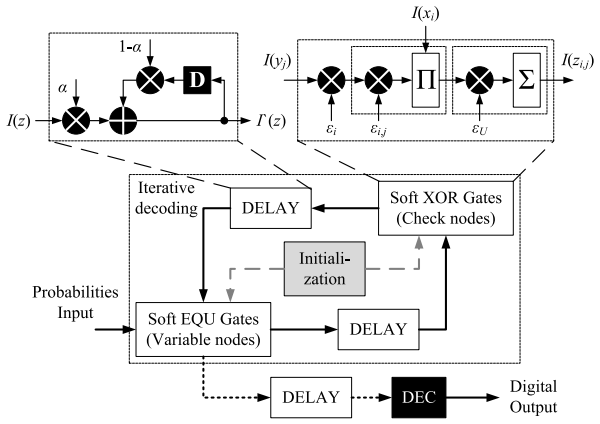


FIGURE 6. Mixed model for analog LDPC decoding circuit.

Based on the physical background of the parameters, we model the mismatch effect of the transistors in the soft-logic gate. As shown in Fig. 2, the fundamental multiplication circuit in the basic gate is divided into two parts: the diode-connected transistors biased in current mode within dashed box and the kernel multiplication transistor matrix biased in voltage mode. The consequence of parameter mismatch on the transistor behavior is calculated differently for different biasing part.

In the current biasing part, the gate-source voltage V_{GS} is dependent on the imposed current $I_{y,j}$, and the variance of the gate-source voltage difference ΔV_{GS} is [27]

$$\sigma^2(\Delta V_{GS}) = \frac{A_{VT0}^2}{WL}, \tag{12}$$

where A_{VT0} is a process-dependent constant. For simplicity, the absolute difference ΔV_{GS} is converted into the relative current error ε_j, whose variance is [27]

$$\sigma^2(\varepsilon_j) = \frac{1}{(nU_T)^2} \cdot \sigma^2(\Delta V_{GS}) = \frac{A_{VT0}^2}{WL(nU_T)^2}. \tag{13}$$

Each transistor in the current biasing part is affected by the mismatch, which changes the input current from $I_{y,j}$ to $I_{y,j}(1+\varepsilon_j)$.

In the voltage biasing part, the current I_D is dependent on the imposed voltage $V_{y,j}$, and the variance of the relative current error $\varepsilon_{i,j}$ for the transistor in row j and column i is [27]

$$\sigma^2(\varepsilon_{i,j}) = \left(\frac{\sigma(\Delta I_{DS})}{I_{DS}}\right)^2 = \frac{A_{VT0}^2}{WL(nU_T)^2}. \tag{14}$$

According to (13) and (14), the mismatch-infected output current $\tilde{I}_{i,j}$ from the multiplication circuits is

$$\tilde{I}_{i,j} = I_{x,i} \cdot \frac{I_{y,j}(1+\varepsilon_j)(1+\varepsilon_{i,j})}{\sum_{j=1}^{n} I_{y,j}(1+\varepsilon_j)}. \tag{15}$$

Similarly, the mismatch-infected output current $\tilde{I}_{z,k}$ from the normalization circuits is

$$\tilde{I}_{z,k} = I_U(1+\varepsilon_U) \cdot \frac{I'_{z,k}(1+\varepsilon_k)(1+\varepsilon_{1,k})}{\sum_{k=1}^{n} I'_{z,k}(1+\varepsilon_k)}, \tag{16}$$

where ε_U, ε_k, and $\varepsilon_{1,k}$ are the current errors in the normalization circuit. According to (15) and (16), the mismatch effects are introduced by all basic gates into the decoding network.

It is shown that the error-correct performance of analog decoding is relatively independent of the distribution of interconnection delays among the nodes [28]. To simplify the analysis, the presented model ignores the propagation delays and only considers the processing delay, which depends strongly on the transistor size.

In Fig. 6, the "Delay" module, which indicates the processing delay, is embedded after the output of each soft-logic gate in the decoding network. The processing delay can be modeled by a first-order RC time-delay approximation, and is characterized by the following recursive equation

$$\begin{aligned} I'_z[n+1] &= \left(1 - e^{(-\Delta t/\tau)}\right) I_z[n] + e^{(-\Delta t/\tau)} I'_z[n] \\ &= \alpha I_z[n] + (1-\alpha) I'_z[n], \end{aligned} \tag{17}$$

where n is the discrete-time index, $\alpha = 1 - \exp(-\Delta t/\tau)$, and Δt is the sampling period. According to (17), the delay model is completely characterized by the time constant τ, which is characterizing the response to a step input of the soft-logic gate. The "Initialization" module, which performs the reset function, restores the network to a uniform starting condition. The "DEC" module performs the hard decisions on the *a posteriori probabilities* (APPs) at each step.

C. SIMULATION RESULTS

The proposed analog decoder circuit is designed and fabricated in a 0.35-μm CMOS technology. Due to the power constraints, the reference current I_U is set to 1 μA. According to (11), the transistor size W/L satisfies

$$\frac{W}{L} \geq \frac{10 I_U}{2\mu C_{ox} U_T^2} \cdot \exp\left(\frac{\kappa V_{T0}}{U_T}\right) \approx 15.2. \tag{18}$$

Based on the above constraint, an MOS transistor with size $W/L = 16\mu\text{m}/1\mu\text{m}$ is adopted in the gate-level design. Using the post-layout simulations, the time constants τ_X for the soft-XOR gate and τ_E for the soft-equal gate are 42.78ns and 57.04ns, respectively.

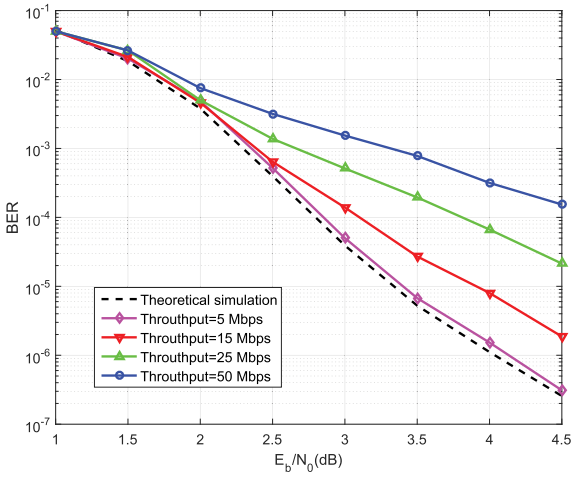


FIGURE 7. BER performance of the proposed analog decoder obtained from simulation.

The simulated BER performance with varying E_b/N_0 is shown in Fig. 7. In the low E_b/N_0 region, the BER performance of different throughputs is close. However, with the increase of E_b/N_0, the higher the throughput, the larger the BER gap between the theoretical curve and the simulation result. This is due to the fact that the settling behaviors of the decoding performance converge to their steady-state values over long decoding time.

V. HARDWARE IMPLEMENTATION

In order to verify the proposed decoding architecture and the modeling approach, an implementation of the (480,240) LDPC analog decoder is carried out using 0.35-μm CMOS technology. The details of this implementation are presented as follows.

TABLE 1. Main features of the analog decoding circuit design.

Reference current (μA)	1
Gilbert multiplier nMOS W/L (μm/μm)	16/1
Sink nMOS W/L (μm/μm)	2/8
Scaling pMOS W/L (μm/μm)	16/1.2
S/H capacitance C1/C2/C3 (nF)	1.2/12/4.8
S/H buffer bias current (μA)	5
DP pMOS W/L (μm/μm)	6/2.4
DP reset nMOS W/L (μm/μm)	4/1
Number of basic soft-XOR gates	$225 \times 16 = 3600$
Number of basic soft-equal gates	$180 \times 16 = 2880$

A. ARCHITECTURE OF THE ANALOG DECODING CIRCUIT

The system level schematic of the proposed (480,240) LDPC analog decoding circuit is shown in Fig. 8. The main features of the analog decoding circuit are summarized in Table 1. Besides the decoding core circuit, the prototype integrates the following modules to facilitate output buffering and testing.

- *Analog input buffer* is used to store the serially-inputting channel output symbols in the analog memory, and to output the symbols in parallel. The storage elements are realized using a two-stage pseudo-differential sample-and-hold (S/H) circuit.
- *Differential pair* biased in weak inversion converts the differential voltages into a pair of complementary currents, and converts the channel output symbols into the channel transition probabilities.
- *Current comparators* generate the digital decision by comparing the output currents, which represent the probabilities of '0' and '1', and then latch the decision.
- *Digital output buffer* is a shift register chain, which samples the parallel decoded bits from the comparator modules and converts them into a bit-serial format.

The timing diagram for the proposed analog LDPC decoder is presented in Fig. 9. When a string of the received serial symbols V_{in} arrives, it is firstly converted into parallel current pairs $[I_{0_{in}}(0:479), I_{1_{in}}(0:479)]$ via S/H controlled by the 'S_{in}' signal and differential pair circuits controlled by the 'Reset' signal. After a settling delay D_{Settle} of the analog decoding circuit, the 'Early stopping' enable signal that indicates the stage of decoding process becomes stable, and the valid parallel current signals $[I_{0_{out}}(0:239), I_{1_{out}}(0:239)]$ are outputted. Finally, the 'Latch' and 'Shift' signals triggered by the stable 'Early stopping' signal allow the comparators and shift register circuits to convert the parallel current signals $[I_{0_{out}}(0:239), I_{1_{out}}(0:239)]$ into a serial digital signal D_{out} with a fixed frame format. In addition, the 'Early stopping' signal is obtained by feeding all ''Stopping feedback'' signals, as shown in Fig. 5, into an 8-level adder-tree.

B. HARDWARE MAPPING OF THE ANALOG DECODER

The analog decoder including the interface circuits is fabricated in a 0.35-μm CMOS process with single power supply 3.3 V. To improve the matching precision, the mirror symmetry form is adopted in the layout of the Gilbert multiplier circuit, and one- and two-dimensional common centroid layout techniques have been used in the current mirrors. The corresponding chip photo is provided in Fig 10. The silicon area including pads is 108.95mm^2, of which the analog decoding circuitry occupies 45.5mm^2. The input buffer circuit takes up a large area on the chip because of the two-stage S/H circuits. The testing circuit on the top left corner is used for the transistor mismatch testing and the circuit behavior verifying of the soft-logic gates. In the analog decoding core, it can be observed that there are 16 reusable modules.

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

The fabricated decoding chip is evaluated by the following experimental setup, as shown in Fig. 11. The received channel output symbols are the composite signals from the noise generator and the signal generator. An FPGA that is working at a frequency of 100 MHz is used for the generation of all digital control signals and the data synchronization. When the

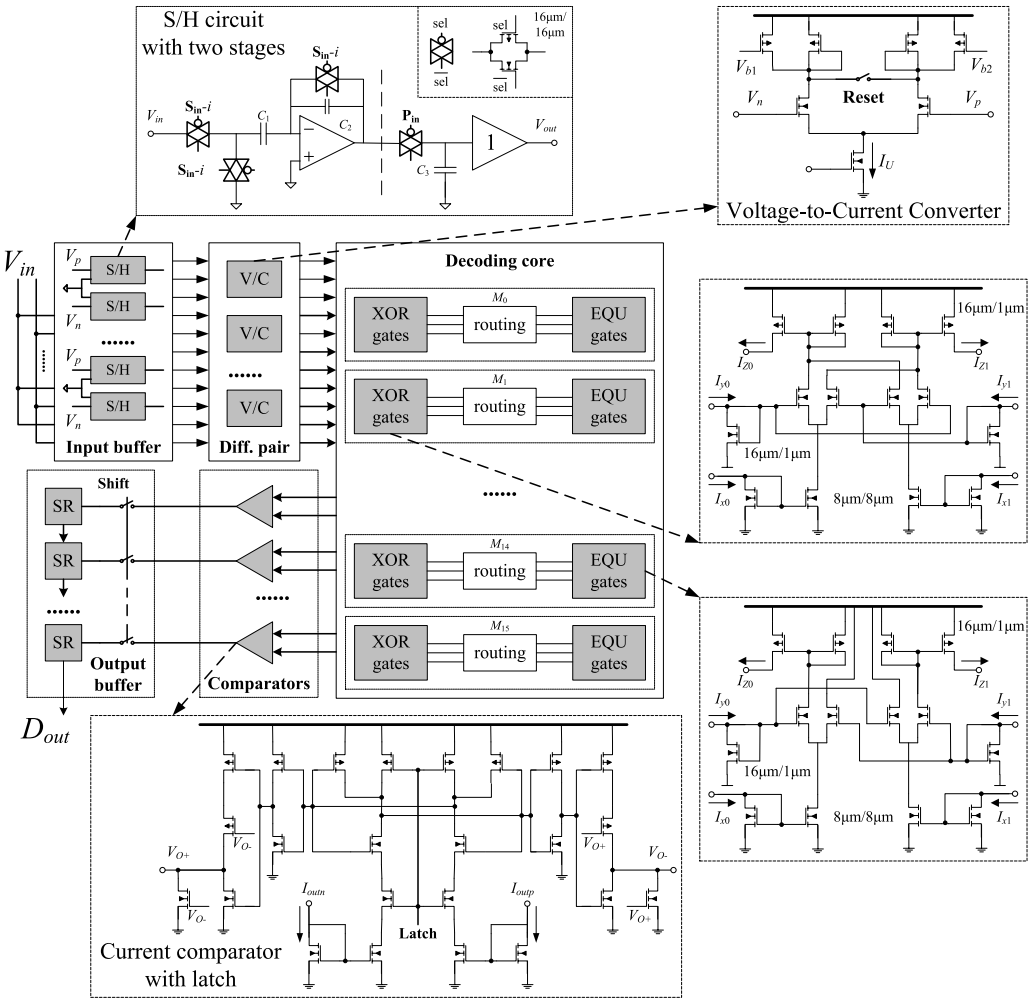


FIGURE 8. System level schematic of the proposed analog LDPC decoding circuit.

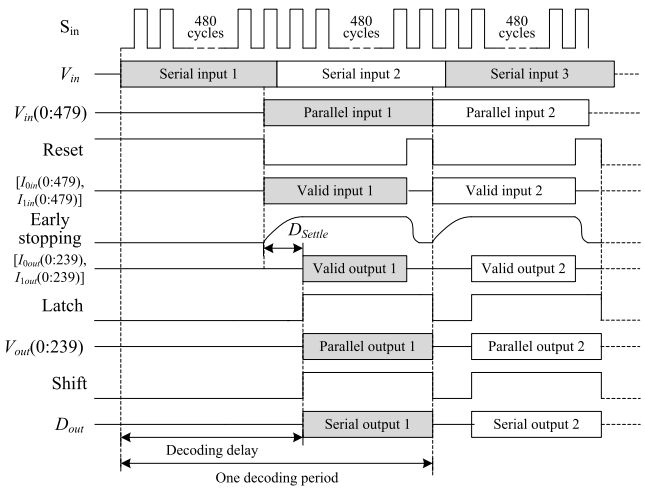


FIGURE 9. The timing diagram of the proposed analog LDPC decoder.

'Early stopping' signal is converging or the pre-determined maximum decoding time is reached, the FPGA retrieves the decoded bits from the decoder chip and records the number of decoding clocks. These data are firstly transferred to the computer, and then the BER and decoding delay are calculated.

Fig. 12 shows the BER performance of the fabricated decoding chip. It is observed that there is a good consistency between the model predictions from Section IV and the experimental results, which confirms the reliability of the proposed mixed behavioral and structural model. The experimental results show that the proposed analog decoder is more resilient to mismatch effect and other imperfection errors than expected [11]. This is because that the modularization design of the decoding architecture and the proper device size derived from the reliable model are adopted in design phase. Moreover, the probabilistic computing paradigm essentially belongs to the error-resilient system. With a throughput of 5Mbps, the experimental results show a loss of about 0.5dB with respect to the benchmark at BER = 10^{-6}, and the proposed decoder offers a superior 5.6dB coding gain at BER = 10^{-4} and 6.3dB coding gain at BER = 10^{-6} compared to an uncoded BPSK (binary phase-shift keying)

TABLE 2. Comparisons with other analog decoders.

Author	Code	CMOS Technology	Core Area (mm^2)	Power (mW)	Throughput (Mb/s)	Energy Efficiency (nJ/bit)	Coding Gain @BER 10^{-4} (dB)
Vogrig et al. [6]	Turbo N=132, R=1/3	0.35μm, 3.3V	4.1 (0.141)*	6.8	2	3.4	4.2 @2Mbps
Hemati et al. [14]	LDPC (32,8)	0.18μm, 1.8V	0.57 (0.074)*	5	6	0.83	2.1 @62.5kbps
Gu et al. [13]	LDPC (32,8)	0.5μm, 3.3V	5.4 (0.091)*	1.254	12.8	0.098	3.2 @320kbps
Abolfazli et al. [12]	TS-LDPC (120,75)	90nm, 1.1, 0.85V	1.38 (0.720)*	13	750	0.017	3.9 @10Mbps
Meraji et al. [11]	(7,5) tail-biting BCJR	65nm, 0.8V	0.038	0.084	2	0.067	<3.4 @125kbps
This work	LPDC (480,240)	0.35μm, 3.3V	45.5 (1.569)*	86.3	50	1.726	5.6 @5Mbps

* Normalized to 65nm

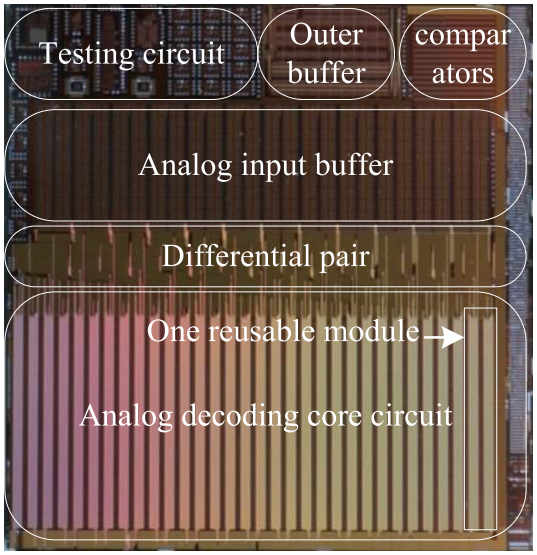


FIGURE 10. Die photograph of the (480,240) analog LDPC decoder chip.

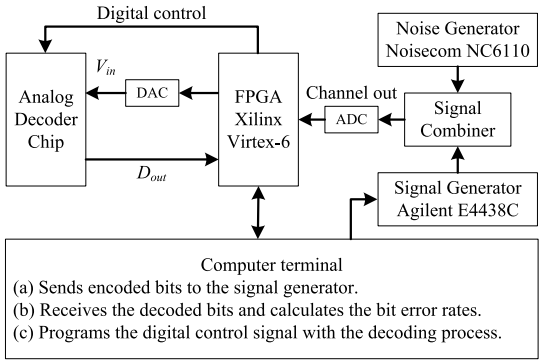


FIGURE 11. Measure setup for the analog decoder chip.

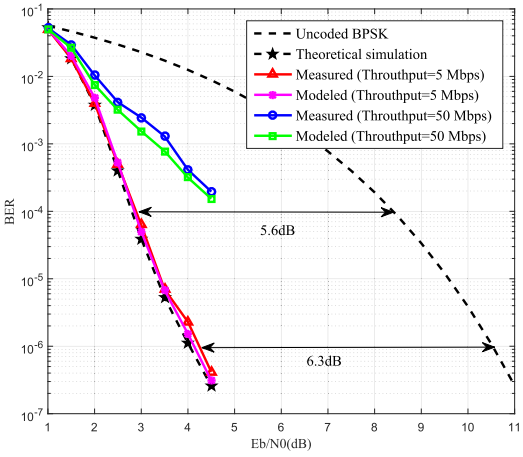


FIGURE 12. Comparison among BER performances of theoretical simulation, modeled analog decoder, and measurement results for the chip.

transmission. Since there is a conflict between the long settling delay of the decoding circuit and the decreasing maximum decoding delay, the performance of the decoder dramatically deteriorates with the increasing throughput. For example, with a throughput of 50Mbps, the degradation of the coding gain is about 2 dB at BER $= 10^{-4}$.

The average decoding delay of the proposed decoding chip is presented in Fig. 13. Here, the maximum decoding delays are set to $(480/throughput)$ s, i.e., 96 μs and 9.6 μs for the throughputs of 5Mbps and 50Mbps, respectively. Comparing with the maximum decoding delays, the average decoding delays are reduced by 93% and 50% at $E_b/N_0 = 4.5$ dB for the two scenarios, respectively. It is shown that the average decoding delay decreases gradually with the increase of E_b/N_0, especially for the scenario with a throughput of 5Mbps. From Fig. 13, we can conclude that the proposed decoder chip employing early termination can save the decoding time with high E_b/N_0. Hence, the probability stopping criterion can reduce the power consumption and improve the data throughput.

The performance comparisons of the proposed decoder with the previously reported analog decoders are summarized in Table 2. It is shown that the code length of our proposed decoder is the longest among all reported analog decoders. Due to the large scale of decoding network and fabricated CMOS technology, the core area and power of the proposed chip is relatively high. When normalized to the state-of-art 65nm CMOS technology [11], the core area will be scaled

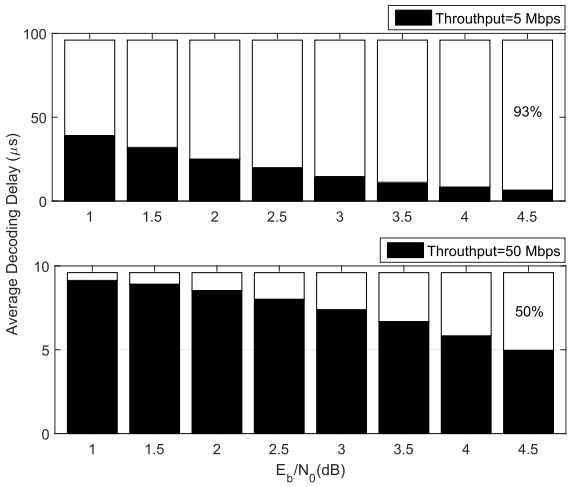


FIGURE 13. Average Decoding Delay at 5Mbps and 50Mbps.

down to a suitable size of about 1.569mm^2. Furthermore, the current level of the energy efficiency is about 10 pJ/bit in 65 nm CMOS, and the power of the analog implementation with advanced process technologies can be further lowered [29]. Our proposed decoder can achieve a coding gain of 5.6dB at BER $= 10^{-4}$ with a throughput of 5Mbps, which is the highest among all existing analog decoders. Hence, the proposed (480,240) CMOS analog LDPC decoder is much more suitable in some energy-limited applications, where moderate throughput and decoding gains are required.

VII. CONCLUSION

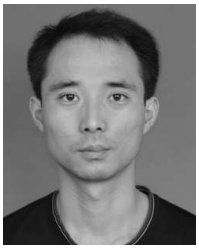
In this paper, a (480,240) CMOS analog LDPC decoder based on SP algorithm has been realized in a 0.35-μm CMOS technology. To the best of our knowledge, the proposed analog LDPC decoder is the longest code implemented using analog technology to date. The low-complexity decoder architecture design exploiting the structured nature of the target code minimizes the hardware overhead. More precisely, by using the reusable modules, only 3180 wires need to be routed in hand-craft process, which are reduced by 44.8%. A mixed behavioral and structural model is also presented for the analog LDPC decoding network, which relates transistor-level parameters to system-level specifications. The contrast result between the simulation value and the measured data validates the reliability of the proposed model. Furthermore, the model can provide circuit-optimization guidelines for the physical design. The chip area is 108.95mm^2 including I/O pads (45.5mm^2 for decoding core). The achievable throughput of the tested chip is higher than 50Mbps with a decoding power of 86.3mW at 3.3V, which corresponds to an energy per decoded bit of 1.726nJ. When the tested throughput is 5Mbps, the decoding chip can offer an outstanding coding gain of 5.6dB at BER $= 10^{-4}$.

REFERENCES

[1] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 1, no. 8, pp. 21–28, Jan. 1962.
[2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *IEEE Power Electron Lett.*, vol. 33, no. 6, pp. 457–458, Mar. 1997.
[3] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[4] P. Hailes, L. Xu, R. G. Maunder, and L. Hanzo, "A flexible FPGA-based quasi-cyclic LDPC decoder," *IEEE Access*, to be published.
[5] A. Li, P. Hailes, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "1.5 Gbit/s FPGA implementation of a fully-parallel turbo decoder designed for mission-critical machine-type communication applications," *IEEE Access*, vol. 4, pp. 5452–5473, Aug. 2016.
[6] D. Vogrig, A. Gerosa, and A. Neviani, "A 0.35-μm CMOS analog turbo decoder for a 40 bit, rate 1/3, UMTS channel code," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 753–762, Mar. 2005.
[7] C. Winstead, J. Dai, S. Yu, C. Myers, R. R. Harrison, and C. Schlegel, "CMOS analog MAP decoder for (8,4) hamming code," *IEEE J. Solid-State Circuits*, vol. 39, no. 1, pp. 122–131, Jan. 2004.
[8] J. Hagenauer and M. Winklhofer, "Probability propagation and decoding in analog vlsi," in *Proc. IEEE ISIT*, Cambridge, MA, USA, Aug. 1998, p. 145.
[9] H. A. Loeliger, M. Helfenstein, and F. Lustenberger, "The analog decoder," in *Proc. IEEE ISIT*, Cambridge, MA, USA, Aug. 1998, p. 146.
[10] M. Bacci *et al.*, "Low-power analogue receiver ASIC for space telecommand applications," in *Proc. TTC Workshop*, Noordwijk, The Netherlands, Sep. 2016, pp. 1–5.
[11] R. Meraji, S. M. Y. Sherazi, H. Sjöland, V. Öwall, and J. B. Anderson, "Low power analog and digital (7,5) convolutional decoders in 65 nm CMOS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 7, pp. 1863–1872, Jul. 2015.
[12] A. R. Abolfazli, Y. R. Shayan, and G. E. R. Cowan, "750 Mb/s 17 pj/b 90 nm CMOS (120,75) TS-LDPC min-sum based analog decoder," in *Proc. IEEE A-SSCC*, Singapore, Nov. 2013, pp. 181–184.
[13] M. Gu and S. Chakrabartty, "A 100 pj/bit, (32,8) CMOS analog low-density parity-check decoder based on margin propagation," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1433–1442, Jun. 2011.
[14] S. Hemati, A. H. Banihashemi, and C. Plett, "A 0.18-μm CMOS analog min-sum iterative decoder for a (32,8) low-density parity-check (LDPC) code," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2531–2540, Nov. 2006.
[15] A. G. I. Amat, S. Benedetto, D. Vogrig, A. Neviani, A. Gerosa, and G. Montorsi, "Design, simulation, and testing of a CMOS analog decoder for the block length-40 UMTS turbo code," *IEEE Trans. Commun.*, vol. 54, no. 11, pp. 1973–1982, Nov. 2006.
[16] D. Miyashita, R. Yamaki, and K. Hashiyoshi, "An LDPC decoder with time-domain analog and digital mixed-signal processing," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 73–83, Jan. 2014.
[17] F. Lustenberger and H.-A. Loeliger, "On mismatch errors in analog-vlsi error correcting decoders," in *Proc. IEEE ISCAS*, Sydney, NSW, Australia, May 2001, pp. 198–201.
[18] N. Duchaux, C. Lahuec, M. Arzel, and F. Seguin, "Analog decoder performance degradation due to BJTs' parasitic elements," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 11, pp. 2402–2410, Nov. 2009.
[19] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
[20] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005.
[21] F. Lustenberger, "On the design of analog VLSI iterative decoders," Ph.D. dissertation, Signal Inf. Process. Lab. (ISI), Swiss Federal Inst. Technol., Zurich, Switzerland, 2000.
[22] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
[23] T. Brack *et al.*, "Low complexity LDPC code decoders for next generation standards," in *Proc. Design Autom. Test Eur. Conf.*, Nice, France, May 2007, pp. 1–6.
[24] R. Echard and S. C. Chang, "Design considerations leading to the development of good π-rotation LDPC codes," *IEEE Commun. Lett.*, vol. 9, no. 5, pp. 447–449, May 2005.
[25] Y. Gao, Z. Zhao, and H. Zheng, "An efficient early stopping analog decoder for LDPC codes," in *Proc. ICCSNT*, Harbin, China, Dec. 2015, pp. 1094–1098.

[26] D. M. Binkley, M. Bucher, and D. Foty, "Design-oriented characterization of CMOS over the continuum of inversion level and channel length," in *Proc. ICECS*, Jounieh, Lebanon, Dec. 2000, pp. 161–164.
[27] P. Kinget and M. Steyaert, *Analog VLSI Integration of Massive Parallel Processing Systems*. Norwell, MA, USA: Kluwer, 1997.
[28] S. Hemati and A. H. Banihashemi, "Dynamics and performance analysis of analog iterative decoding for low-density parity-check (LDPC) codes," *IEEE Trans. Commun.*, vol. 54, no. 1, pp. 61–70, Jan. 2006.
[29] C. Schlegel and V. Gaudet, "Hardware implementation challenges of modern error control decoders," in *Proc. ISCAS*, Rio de Janeiro, Brazil, May 2011, pp. 1788–1791.

ZHE ZHAO received the B.S. and M.S. degrees from the Harbin Institute of Technology in 2007 and 2009, respectively. He is currently pursuing the Ph.D. degree with the Beijing Institute of Technology, Beijing, China. His research interests include channel coding and analog signal processing. In particular, he contributed to the design and testing of two prototypes of analog LDPC decoders in CMOS technology.



KAI YANG received the B.S. degree in communications engineering from the National University of Defense Technology, Changsha, China, in 2005, and the Ph.D. degree in communications engineering from the Beijing Institute of Technology, Beijing, China, in 2010. He is currently with the School of Information and Electronics, Beijing Institute of Technology. His current research interests include convex optimization, cooperative communications, multiple-input-multiple-output systems, resource allocation, and interference mitigation.

HAO ZHENG received the B.S. degree from the Beijing Institute of Technology in 2011 and the M.S. degree from Jinan University in 2013. He is currently pursuing the Ph.D. degree with the Beijing Institute of Technology, Beijing, China. His research interests include channel coding, iterative decoding, and massive access networks.

FEI GAO received the B.S. and M.S. degrees from the Changchun University of Science and Technology, Changchun, China, in 1982 and 1989, respectively, and the Ph.D. degree from the Institute of Electrics, Chinese Academy of Sciences, Beijing, China, in 1999. She is currently a Professor with the School of Information and Electronics, Beijing Institute of Technology. Her research interests include digital watermarking and chaos-based security communication.

XIANGYUAN BU received the B.S. and Ph.D. degrees from the Beijing Institute of Technology, Beijing, China, in 1987 and 2007, respectively. He is currently a Professor with the School of Information and Electronics, Beijing Institute of Technology. His research interests include software-defined radio and digital communications.