Area-efficient analog decoder design for low density parity check codes in deep-space applications

Zhao Zhe (✉), Gao Fei, Zheng Hao, Yin Xue

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

Abstract

Area-efficient design methodology is proposed for the analog decoding implementations of the rate-1/2 accumulate repeat-4 jagged-accumulate (AR4JA) low density parity check (LDPC) code. The proposed approach is designed using optimized decoding architecture and regularized routing network, in such a way that the overall wiring overhead is minimized and the silicon area utilization is significantly improved. The prototyping chip used to verify the approach is fully integrated in a four-metal double-poly 0.35 µm complementary metal oxide semiconductor (CMOS) technology, and includes an input-output interface that maximizes the decoder throughput. The decoding core area is 2.02 mm$^2$ with a post-layout area utilization of 80%. The decoder was successfully tested at the maximum data rate of 10 Mbit/s, with a core power consumption of 6.78 mW at 3.3 V, which corresponds to an energy per decoded bit of 0.677 nJ. The proposed analog LDPC decoder with low processing power and high-reliability is suitable for space- and power-constrained spacecraft system.

Keywords  low density parity check (LDPC) code, analog decoding, iterative message-passing algorithms, hardware efficient, area utilization

1 Introduction

Forward error correction (FEC) codes, which can be employed to correct transmission errors, have been an important component of space communication [1–2]. In deep-space missions, the use of FEC codes perhaps should be the single most cost effective means to improve the system performance in contrast to the larger power amplifiers and the bigger antennas. LDPC codes have recently gained acceptance in the aerospace community [3–4] because of their capacity-approaching performance and the ease of parallel implementation in hardware. A family of AR4JA LDPC codes [5] has been incorporated into the Consultative Committee for Space Data Systems (CCSDS) standards, and has been in use on several current missions [6]. When designing the LDPC decoders for deep-applications, power efficiency is one of the paramount concerns. The digital approach currently follows a field programmable gate array (FPGA)-based architecture with a precision analog-to-digital converter (ADC) [7–8]. However, the increasing data storage capacity and processing speed with the rapid pace of integration make the digital decoders more and more expensive in terms of hardware complexity and power consumption [9–10].

The major motivation to use analog decoders is based on the promises of low power dissipation and fast processing speed [11–12]. Compared with their digital counterpart, analog-based microchip can perform the complex operations in iterative decoding algorithms more efficiently with less hardware and power consumptions. Without the quantization process, the analog implementations can provide a finer estimation of the logic state of a single information bit. By the means of high modularity design, it is more immune to the non-ideal effects, such as the transistor mismatch effects and noise. Owing to the full-parallel, asynchronous, and continuous time processing, it also improves the total system efficiency. Because of the probabilistic computing paradigm, it offers more efficient error resilient capabilities.
against single event transient (SET). As a consequence, the analog message-passing methodology is suitable for space- and power-constrained spacecraft system.

Over the past few years, several analog decoding chips are already available in Refs. [13–17], claiming an outstanding improvement in the power efficiency with respect to their digital counterparts. However, these implementations are merely limited to proof-of-concept decoders with very short block lengths, which are unable to provide enough coding gains for practical application. Pursuing the analog decoding approaches based on the full-parallel architecture into the practicality introduces various challenges. The major challenge is that due to the customized hand-craft design, the straightforward implementations suffer from the extremely poor area utilization and a considerable speed penalty. The work in Ref. [15] devotes roughly 40% of the chip area to routing. This implementation, from the other perspective, revealed the routing congestion rather than gate count as the bottleneck in implementation of analog decoding circuits.

In this work, a designing approach is proposed for the analog decoder of the rate-1/2 AR4JA LDPC code, which is suitable for the deep-space applications. In this approach, the combination of a scalable decoding architecture and a well-thought-out routing strategy can minimize the hardware complexity and maximize the area efficiency. A prototype chip used to verify the approach is fabricated in a 0.35 µm standard CMOS process, and the measurement results show that the decoder can achieve a throughput of 10 Mbit/s and a power consumption of 6.768 mW.

The paper is organized as follows. Sect. 2 describes some preliminaries, including the AR4JA LDPC codes, the basics of the iterative message-passing algorithm and the basics of the analog sum-product module. Details on the proposed decoding architecture design and routing strategy are discussed in Sect. 3. The experimental setup and measurements obtained from fabricated decoder prototypes are shown in Sect. 4. Finally, Sect. 5 concludes this paper.

2 Preliminaries

2.1 AR4JA LDPC code

In this paper, we consider the analog implementation of the rate-1/2 LDPC code defined by the CCSDS standard. The selected code belongs to a family of AR4JA LDPC codes that exhibit very low error floors. Moreover, the AR4JA design also ensures that the code’s minimum distance grows linearly with the block size.

The AR4JA codes are structured LDPC codes built by making copies of a protograph and permuting the connecting edges. A protograph is a Tanner graph with a relatively small number of nodes. The protograph of the rate-1/2 AR4JA LDPC code is shown in Fig. 1, where the filled circles represent the variable nodes, the squares with a cross represent the check nodes, and the open circle represents variable nodes corresponding to symbols that are punctured, i.e., are not transmitted over the channel.

Similarly, from the view of parity-check matrix, the AR4JA codes are designed by lifting up a protograph parity-check matrix into a larger parity-check matrix consisting of circulants. The parity-check matrix \( H \) for the rate-1/2 codes are constructed from \( \frac{M}{2} \times \frac{M}{2} \) sub-matrices and specified as follows,

\[
H = \begin{bmatrix}
0_M & 0_M & I_M & 0_M & I_M & I_M \oplus I_1 \\
I_M & I_M & 0_M & I_M & I_M \oplus I_2 \oplus I_8 & I_M \\
I_M & I_2 \oplus I_6 & 0_M & I_M & I_M \oplus I_8 & I_M \\
I_M & I_3 \oplus I_6 & 0_M & I_M & I_M \oplus I_8 & I_M \\
I_M & I_4 \oplus I_6 & 0_M & I_M & I_M \oplus I_8 & I_M \\
I_M & I_5 \oplus I_6 & 0_M & I_M & I_M \oplus I_8 & I_M
\end{bmatrix}
\]

(1)

where \( I_M \) and \( 0_M \) are the \( M \times M \) identity and zero matrices, respectively, and \( I_I \) through \( I_8 \) are permutation matrices.

2.2 Message-passing schedules for decoding

The iterative message-passing algorithm, which offers near-optimum decoding performance at a manageable complexity, is the most widely used method for large linear LDPC codes. The LDPC decoding procedure operates on a graphical representation of the code dependencies, on which the sum-product algorithm is executed. A simplified illustration of decoding procedure is shown in Fig. 2, and the soft messages in terms of log-likelihood ratios (LLR) are passed between check-nodes and variable-nodes. In the first step, variable node \( v_i \) is initialized with the prior message \( L(v_i) \) using the noisy channel information of the transmitted bit \( y_i \):
\[ L(v_i) = \ln \frac{\Pr(x_i = 0 | y_i)}{\Pr(x_i = 0 | y_i)} = \frac{2}{\sigma^2} y_i \]  

where \( \sigma^2 \) is the variance of a Gaussian distributed random noise variable.

The check node \( c_j \) receives messages \( L(v_i \rightarrow c_j) \) from its neighboring variable node \( v_i \), and then computes messages \( L(c_j \rightarrow v_i) \) to be sent to the variable nodes \( v_i \).

\[ L(c_j \rightarrow v_i) = 2 \text{artanh} \left( \prod_{v_k \in C_j \setminus v_i} L(v_k \rightarrow c_j) \right) \]  

where \( C_j \) is the neighboring variable node set of the node \( c_j \) excluding the node \( v_i \).

Similarly, the messages from the node \( v_i \) to the node \( c_j \) is

\[ L(v_i \rightarrow c_j) = L(v_i) + \sum_{v_k \in V_j \setminus c_j} L(c_j \rightarrow v_k) \]  

where \( V_j \) is the neighboring variable node set of the node \( v_i \) excluding the node \( c_j \).

This process repeats for another iteration by passing the previous iteration’s messages \( L(v_i \rightarrow c_j) \) to the check nodes. Finally, the algorithm terminates when it reaches a maximum number of decoding iterations or a valid code word is detected.

2.3 Module design in analog decoders

The basic cells, usually called the sum-product module, should be viewed as generalizations of the well-known Gilbert multiplier in sub-threshold CMOS technologies and implement sum and product operations required by the decoding algorithms. In this approach, currents represent probabilities and voltages represent LLR values. These currents are normalized to the ‘unity’ current \( I_U \). The value of this normalizing current is a key parameter in the design phase.

The transistor model of fundamental circuit, two-input Gilbert multiplier, is shown in Fig. 3 and the function of the circuit is then given by

\[ I_{a,b} = \sum_{a=1}^{\infty} \sum_{b=1}^{\infty} \left( I_{a,a} I_{b,b} \right) \frac{I_a I_b}{I_a I_b} \]  

where \( I_a = \sum_{a=0}^{\infty} I_{a,a} \) and \( I_b = \sum_{b=0}^{\infty} I_{b,b} \).

![Fundamental circuit: pairwise multiplication of probabilities](image)

We can use the Gilbert multiplier as the core circuit for both types of nodes in factor graph. In order to clarify the relationship between decoder nodes and the Gilbert multiplier, here we illustrate a two input variable node and derive its output in terms of currents in the circuit.

In Eq. (6) we present the construction of a two-input variable node

\[ L_v = \ln \frac{\Pr(a = 0)}{\Pr(a = 1)} + \ln \frac{\Pr(b = 0)}{\Pr(b = 1)} = \ln \frac{\Pr(a = 0) \Pr(b = 0)}{\Pr(a = 1) \Pr(b = 1)} = \ln \frac{I_{a,a} I_{b,b}}{I_{a,b} I_{b,a}} \]  

where \( L_v \) is the output LLR of the variable node.

The output of a two input check node, \( L_c \), is given by

\[ L_c = \ln \frac{\Pr(a = 0) \Pr(b = 0) + \Pr(a = 1) \Pr(b = 1)}{\Pr(a = 0) \Pr(b = 1) + \Pr(a = 1) \Pr(b = 0)} = \ln \frac{I_{a,a} I_{b,b} + I_{a,b} I_{b,a}}{I_{a,a} I_{b,b} + I_{a,b} I_{b,a}} \]  

Note that low level of currents is not only necessary for proper weak inversion operation, but also maintains low power consumption.

3 Optimized decoder architecture for AR4JA

The implementation of the analog decoder presents many unique challenges because the architectural characteristics of the decoder are so dissimilar to those of a typical application specific integrated circuit (ASIC).

In an analog decoder, the algorithmic computations are performed in continuous time, so there is no need for
temporary storage of intermediate data, which is one of their advantages over digital implementations. In other words, analog decoders need to be implemented in full-parallel architecture, which relies on the flooding schedule. The full-parallel LDPC decoder can be implemented by constructing the isomorphic architecture, which is a direct mapping of the code-representing factor graph to two types of hardware components: variable processing nodes and check processing nodes to compute the update equations, and an interconnect network to represent the edges of the graph. Due to the intrinsic parallelism offered by the codes, the full-parallel decoders can potentially have the high throughput and energy efficiency, which are matching the space applications. Unfortunately, the complex routing network that connects check nodes and variable nodes turns out to be a major implementation bottleneck in the design phase. The straightforward implementations of the full-parallel decoder exhibit extremely poor area utilization and suffer from a considerable speed penalty, and this may even become worse when proceeding to more advanced process technologies.

The AR4JA LDPC codes adopt group-and-permute design approach, in which the rows and columns of the corresponding parity-check matrices are divided into groups of the same size. In other words, these codes are built from protographs and circulants, and are particularly amenable to decoders implemented in hardware. As sketched in Fig. 4, the decoding architecture is constructed from $Z$ copies of a small $(n, r)$ protograph. More specifically, each group of variable nodes represented by circles is derived from a single variable node in the protograph. Similarly, the check nodes are shown as stacked sets of squares. Each edge in the small protograph is expanded into a set of edges in the full factor graph and represents a type of permutation, so the interconnections between the computation units are well organized. The scalable decoding architecture for the AR4JA LDPC codes takes full advantage of protograph and circulant construction.

To solve the problem of routing congestion in the physical design, careful use of hierarchy was employed. Because such structured codes allow the grouping of check and variable nodes, the wires between two types of nodes in the same group can be bundled and routed together. As shown in the Eq. (1), there are three kinds of sub-matrices, $I_{M}$, $0_{M}$ and $\Pi_{l}$ ($l = 1, 2, \ldots, 8$), which respectively correspond to three types of interconnections between nodes within the group. The connection corresponding to the sub-matrix $0_{M}$ is none, and the connection corresponding to the identity matrix $I_{M}$ is the simplest type where the node in row is connected to the one in column in order. The permutation matrix $\Pi_{l}$ specifies a connection $\Pi_{l}(i) = j$, then the $i$th variable node is connected to the $j$th check node. Note that the identity matrix can be regarded as a special permutation matrix $\Pi_{0}$.

The proposed routing strategy for structured LDPC decoder is localizing irregular wires and regularizing global wires. In the local routing, the router predetermined by the type of permutation is used to encapsulate irregular local wiring. Furthermore, the global wires outside of the permute router are regular and structured, thereby significantly reducing the wiring overhead. The block diagram of the decoder for rate-1/2 AR4JA code is shown in the Fig. 5. There are three groups of check nodes, five groups of variable nodes and nine types of permute router. With the combination of an optimized decoding architecture and the permutation routing technique to reduce routing congestion, the area efficiency can be significantly improved.
4 Chip implementation and measure results

To evaluate the performance of the proposed analog decoding scheme, a prototyping chip based on the rate-1/2 AR4JA LDPC code with the block length $n=40$ is designed and fabricated in a 0.35 µm standard CMOS process. Fig. 6 shows a system level schematic of the corresponding decoder. Besides the decoding network, the prototype integrates the following modules to facilitate output buffering and testing through an external digital interface.

1) Analog input buffer is used to store the serially-input 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.

2) Differential pair biased in weak inversion converts the differential voltages into a pair of complementary currents, and this converts the channel output symbols into the channel transition probabilities.

3) Output comparators generate digital decision by comparing the output currents representing the probabilities of ‘0’ and ‘1’, respectively. And the decisions are simultaneously latched.

4) Digital output buffer is a shift register chain which samples the parallel decoded bits (from the comparator modules) and convert them into a bit-serial format.

The microphotograph of a prototype analog based LDPC decoder is shown in Fig. 7. The main specifications of the fabricated chip are summarized in Table 1. Transistor sizes are reported for the core decoder circuit, in which each transistor has a ratio of 2 for transistors used in Gilbert multipliers and 0.5 for transistors used in current mirrors. The reported decoder power consumption refers to the power consumed in the core decoder, excluding the interfaces. The chip’s behavior was verified at data rate from 100 kbit/s to 10 Mbit/s which is of interest for the scope of deep-space telemetry applications. Typical power consumption of the processing core is 6.768 mW corresponding to the speeds 100 kbit/s.

To generate the required test data, a communication system with the binary phase shift keying (BPSK) modulation and the additive white Gaussian noise (AWGN) channel is considered. The measurement setup used to evaluate the fabricated chip is shown in Fig. 8. The FPGA which is responsible for clock generation and data synchronization selectively programs the digital-to-analog converter (DAC) to emulate data received over an AWGN channel after which the FPGA enables the analog decoding core. The decoded bits are serially retrieved from the LDPC chip and stored on the FPGA. At the end of each Monte Carlo run, the overall bit error rate (BER) will be computed. Test files were generated in Matlab for signal noise ratio (SNR) from 2 dB to 6 dB in steps of 1 dB.
The BER measurement results corresponding to different decoding throughput obtained from the fabricated prototype are shown in Fig. 9.

![Fig. 9 BER performance of the proposed decoder](image)

At low SNR, the hardware implementation outperforms its simulated counterpart which could be attributed to the continuous-time dynamics of the analog decoder. However, at high SNR the simulation result outperforms its hardware counterpart which could be attributed to the limited dynamic range of the programming DACs and due to the offset and gain errors introduced by the Gilbert multiplexer.

Table 2 summarizes the main features of the fabricated analog decoder along with those of the previously reported CMOS analog and digital iterative decoders. Due to different codes with different rates, and different CMOS technologies, it is difficult to make a fair comparison between these decoders. Because of the compact decoding architecture and the ingenious routing method, the proposed decoder has the highest silicon area utilization rate among the reported CMOS analog decoder. More importantly, this post-layout feature will maintain high efficiency as implementing the decoder architecture toward longer code words.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>CMOS Technology/µm</th>
<th>Core area/mm²</th>
<th>Area utilization(%)</th>
<th>Power/mW</th>
<th>Throughput (Mbit·s⁻¹)</th>
<th>Energy efficiency (nJ·bit⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital</td>
<td>LDPC [1 024,512] in Ref. [7]</td>
<td>0.160</td>
<td>52.50</td>
<td>50.0</td>
<td>690</td>
<td>500</td>
<td>1.400</td>
</tr>
<tr>
<td>Digital</td>
<td>LDPC [1 024,860] in Ref. [8]</td>
<td>0.065</td>
<td>5.35</td>
<td>84.5</td>
<td>144</td>
<td>6 670</td>
<td>0.022</td>
</tr>
<tr>
<td>Analog</td>
<td>Turbo (48,16) in Ref. [13]</td>
<td>0.35</td>
<td>1.42</td>
<td>77.0</td>
<td>185.000</td>
<td>13.3</td>
<td>13.900</td>
</tr>
<tr>
<td>Analog</td>
<td>Turbo (132,40) in Ref. [14]</td>
<td>0.35</td>
<td>4.10</td>
<td>46.0</td>
<td>6.800</td>
<td>2.0</td>
<td>3.400</td>
</tr>
<tr>
<td>Analog</td>
<td>LDPC (32,8) in Ref. [15]</td>
<td>0.18</td>
<td>0.57</td>
<td>60.0</td>
<td>5.000</td>
<td>6.0</td>
<td>0.830</td>
</tr>
<tr>
<td>Analog</td>
<td>LDPC (32,8) in Ref. [16]</td>
<td>0.50</td>
<td>5.40</td>
<td>60.0</td>
<td>1.254</td>
<td>12.8</td>
<td>0.098</td>
</tr>
<tr>
<td>Analog</td>
<td>TS-LDPC (120,75) in Ref. [17]</td>
<td>0.09</td>
<td>1.38</td>
<td>54.0</td>
<td>13.000</td>
<td>750.0</td>
<td>0.017</td>
</tr>
<tr>
<td>Analog</td>
<td>LPDC (40,16)* in the proposed method</td>
<td>0.35</td>
<td>2.02</td>
<td>80.0</td>
<td>6.768</td>
<td>10.0</td>
<td>0.677</td>
</tr>
</tbody>
</table>

5 Conclusions

In this paper, an analog decoder for the AR4JA LDPC code, which can be applied to the deep-space missions, is designed based the proposed area-efficient approach. A scalable decoding architecture exploiting the structured nature of the code is used to minimize the hardware complexity. When aided by a well-thought-out routing strategy, the full-parallel decoding architecture achieves 80% area utilization, which is one of the highest area efficiency to date using analog techniques. A proof-of-concept analog decoder chip for a (40, 16) LDPC code was fabricated using 0.35 µm standard technology. The decoder chip was successfully tested at the maximum data rate applied to the deep-space telemetry missions (10 Mbit/s), with a core power consumption of 6.768 mW at 3.3 V and a low energy per decoded bit (0.677 nJ). The measurement results validate the feasibility of the area-efficient design for the analog LDPC decoders.

References

4. Zhao W H, Long J P. Implementing the NASA deep space LDPC codes for

(Editor: Wang Xuying)